

**TERMOELEKTRİK MODÜLLERDEN OLUŞAN BİR SİSTEMİN YAPAY
SİNİR AĞLARI İLE MODELLENMESİ**

Gamze Hatice BİLEN

**YÜKSEK LİSANS TEZİ
ELEKTRONİK VE BİLGİSAYAR EĞİTİMİ**

**GAZİ ÜNİVERSİTESİ
BİLİŞİM ENSTİTÜSÜ**

**NİSAN 2011
ANKARA**

Gamze Hatice BİLEN tarafından hazırlanan TERMOELEKTRİK MODÜLLERDEN OLUŞAN BİR SİSTEMİN YAPAY SİNİR AĞLARI İLE MODELLENMESİ adlı bu tezin Yüksek Lisans olarak uygun olduğunu onaylarım.

Prof. Dr. İnan GÜLER

Tez Yöneticisi

Bu çalışma, jürimiz tarafından oy birliği ile Elektronik ve Bilgisayar Eğitimi Anabilim Dalında Yüksek lisans olarak kabul edilmiştir.

Başkan: : Prof. Dr. Ömer Faruk BAY

Üye : Prof. Dr. İnan GÜLER

Üye : Yrd. Doç. Dr. Derya YILMAZ

Tarih : 04/05/2011

Bu tez, Gazi Üniversitesi Bilişim Enstitüsü tez yazım kurallarına uygundur.

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada orijinal olmayan her türlü kaynağa eksiksiz atıf yapıldığını bildiririm.

Gamze Hatice BİLEN

**TERMOELEKTRİK MODÜLLERDEN OLUŞAN BİR SİSTEMİN YAPAY
SİNİR AĞLARI İLE MODELLENMESİ**

(Yüksek Lisans Tezi)

Gamze Hatice BİLEN

GAZİ ÜNİVERSİTESİ

BİLİŞİM ENSTİTÜSÜ

Nisan 2011

ÖZET

Termoelektrik modüllerden oluşan bir sistemin yapay sinir ağları ile eğitilmesinde termoelektrik başlıktan elde edilen deney verileri kullanılmıştır. Deney verileri Alyuda Neuro Intelligence programında Quick Propagation, Conjugate Gradient Descent, Quasi-Newton, Limited Memory Quasi-Newton, Levenberg Marquardt, Online Back Propagation ve Batch Back Propagation algoritmaları ile eğitilmiştir. Elde edilen en başarılı eğitim, C#' da kodlanan eğitim programı ile karşılaştırılmıştır. Eğitimin performans ölçütü olarak Korelasyon katsayısı, Regresyon analizi ve Mutlak hata değerleri kullanılmıştır. Böylece en başarılı eğitim ağ yapısı bulunarak benzer sistemlerde yaşanan sistemin çıkışını olumsuz etkileyen ortamın sıcaklığı, akım, gerilim gibi faktörler sonucu yanlış ölçümlerin yapılması ayrıca ölçüm sonuçlarının kontrol edilmesi için defalarca ölçüm yapılması gibi olumsuzlukların önüne yapay sinir ağların ileri yönelik tahmin yeteneği kullanılarak geçmek amaçlanmıştır. Yapılan eğitimler sonucunda C#'da kodlanan eğitim programının, Alyuda Neuro Intelligence programından elde edilen en başarılı eğitim ile yaklaşık sonuç verdiği görülmüştür.

Bilim Kodu : 704.3.013

Anahtar Kelimeler: termoelektrik, termoelektrik başlık, yapay sinir ağı,
regresyon, korelasyon katsayısı, çok katmanlı algılayıcı

Sayfa Adedi : 78

Tez Yöneticisi : Prof. Dr. İnan GÜLER

**MODELLING OF THE SYSTEM INCLUDING THERMOELEKTRIC
MODULES USING ARTIFICIAL NEURAL NETWORKS**

(M.Sc. Thesis)

Gamze Hatice BİLEN

GAZİ UNIVERSITY

INFORMATICS INSTITUTE

April 2011

ABSTRACT

The system including thermoelectric modules is trained by using the experimental data obtained from thermoelectric cap. Experimental data are trained by Alyuda Neuro Intelligence program using Quick Propagation, Conjugate Gradient Descent, Quasi-Newton, Limited Memory Quasi-Newton, Levenberg Marquardt, Online and Batch Back Propagation algorithms. Simulation program encoded by C # is compared with the most successful simulation trained by Alyuda Neuro Intelligence program, Correlation coefficient, regression analysis and the absolute error values were used for criterion of performance. Thus, by finding the most successful education systems of the network structure intend to prevent negative factors can be seen the other systems like affecting the measurement results as environment temperature that affecting the output of the system, current, voltage and take many times to check the measurement. It is seen that the result of training simulation program encoded by C # is almost same with results of the simulation like the Alyuda Neuro Intelligence's most successful results of the simulation.

Science Code : 704.3.013
Key Words : thermoelectric, thermoelectric teadgear, artificial neural network, regression, correlation coefficient, multi layer perceptron
Page Number : 78
Adviser : Prof. Dr. İnan GÜLER

TEŐEKKÜR

Çalıőmalarım boyunca deęerli yardım ve katkılarıyla beni yönlendiren Sayın Prof. Dr. İnan GÜLER'e, tez için gereken verileri saęlayan Dr. Hakan YAVUZ'a teőekkürü bir borç bilirim. Ayrıca tüm çalıőmalarım boyunca bana destek veren ailem ve öęretmen arkadaşlarıma da teőekkür ederim.

İÇİNDEKİLER

	Sayfa
ÖZET	iv
ABSTRACT.....	vi
TEŞEKKÜR.....	viii
İÇİNDEKİLER	ix
ÇİZELGELERİN LİSTESİ.....	xi
ŞEKİLLERİN LİSTESİ	xiii
RESİMLERİN LİSTESİ	xv
SİMGELER VE KISALTMALAR.....	xvi
1. GİRİŞ	1
2. MATERYAL VE METOT	3
2.1. Termoelektrik ve Bileşenleri.....	3
2.1.1. Termoelektrik başlık	4
2.1.2. Deney düzeneği.....	5
2.2. Yapay Sinir Ağları ve Bileşenleri	7
2.2.1. İnsanda sinir sistemi	8
2.2.2. Yapay sinir hücresi.....	9
2.2.3. Yapılarına göre yapay sinir ağları	13
2.2.4. Öğrenme algoritmalarına göre yapay sinir ağları.....	14
2.2.5. Öğrenme zamanına göre yapay sinir ağları.....	16
2.2.6. Yapay sinir ağlarının ağ yapısı.....	16
2.2.7. Çok katmanlı algılayıcı	17
2.2.8. Online geri yayılım algoritması.....	20

2.2.9. Kümesel geri yayılım algoritması	21
2.2.10. Hızlı yayılım algoritması.....	22
2.2.11. Levenberg Marquardt algoritması	22
2.2.12. Newton öğrenme algoritması	23
2.2.13. Normalizasyon	23
2.2.14. Korelasyon katsayısı ve regresyon.....	24
2.2.15. ÇKA ağının çalışma şekli.....	25
2.3. Yazılım.....	25
2.3.1. Materyal	26
2.3.2. Metot	26
3. BULGULAR VE TARTIŞMA	32
3.1. Termoelektrik Başlık Verileri İle Yapay Sinir Ağı Analizi	32
3.1.1. %80'i eğitim, %20'si test seti olan veri setinin YSA ile analizi.....	32
3.1.2. %50'si eğitim, %50'si test girişi olan veri setinin YSA ile analizi...41	
3.1.3. CPEP'de tüm deney sonuçlarının test edilmesi	50
3.2. Tartışma.....	51
4. SONUÇ VE ÖNERİLER	53
KAYNAKLAR	55
EKLER.....	59
EK-1. CPEP eğitim programı kodları	60
ÖZGEÇMİŞ	78

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 2.1. Ağ türleri ve başarılı oldukları alanlar	16
Çizelge 2.2. Eğitim ve test setinin normalizasyon sonucu.....	27
Çizelge 2.3. ANI ile QP algoritmasında bulunan mutlak hata değerleri.....	28
Çizelge 2.4. OBP katsayılarının tespit analizi.....	29
Çizelge 2.5. CPEP' de kullanılan OBP katsayılarının tespit sonuçları	30
Çizelge 2.6. BBP katsayılarının tespit analizi	30
Çizelge 3.1. Eğitimde kullanılan algoritmaların değişken değerleri.....	33
Çizelge 3.2. Algoritmalarından elde edilen performans değerleri	33
Çizelge 3.3. QP eğitiminin performans sonuçları	34
Çizelge 3.4. CGD eğitiminin performans sonuçları.....	35
Çizelge 3.5. QN eğitiminin performans sonuçları	36
Çizelge 3.6. LMQN eğitiminin performans sonuçları	37
Çizelge 3.7. LM eğitiminin performans sonuçları	38
Çizelge 3.8. OBP eğitiminin performans sonuçları	39
Çizelge 3.9. BBP eğitiminin performans sonuçları.....	40
Çizelge 3.10. CPEP eğitiminin performans sonuçları.....	41
Çizelge 3.11. Eğitimde kullanılan değişken değerleri	42
Çizelge 3.12. Algoritmalarından elde edilen performans değerleri	42
Çizelge 3.13. QP eğitim performans çizelgesi	43
Çizelge 3.14. CGD eğitim performans çizelgesi.....	44
Çizelge 3.15. QN eğitim performans çizelgesi	45

Çizelge	Sayfa
Çizelge 3.16. LMQN eğitim performans çizelgesi	46
Çizelge 3.17. LM eğitim performans çizelgesi	47
Çizelge 3.18. OBP eğitim performans çizelgesi	48
Çizelge 3.19. BBP eğitim performans çizelgesi.....	49
Çizelge 3.20. CPEP performans çizelgesi.....	50
Çizelge 3.21. Ağın genel performansı.....	51

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 2.1. Sistemin çalışma prensibi	3
Şekil 2.2. Termoelektrik sistem blok diyagramı	6
Şekil 2.3. Tek bir modülün yapısı ve deney düzeneği	7
Şekil 2.4. Biyolojik sinir sisteminin blok gösterimi.....	8
Şekil 2.5. Nöronun biyolojik modeli.....	9
Şekil 2.6. Yapay sinir hücresi	9
Şekil 2.7. Doğrusal aktivasyon fonksiyonu	11
Şekil 2.8. Adım aktivasyon fonksiyonu	11
Şekil 2.9. Sigmoid aktivasyon fonksiyonu.....	12
Şekil 2.10. Tanjant hiperbolik aktivasyon fonksiyonu.....	12
Şekil 2.11. İleri beslemeli ağ blok diyagramı	13
Şekil 2.12. Geri beslemeli ağ blok diyagramı	14
Şekil 2.13. Danışmalı öğrenme blok diyagramı.....	14
Şekil 2.14. Danışmansız öğrenme blok diyagramı.....	15
Şekil 2.15. Destekleyici öğrenme blok diyagramı	15
Şekil 2.16. Geri yayılım ÇKA yapısı	18
Şekil 2.17. Termoelektrik başlığın blok diyagramı.....	26
Şekil 3.1. QP' nin ANI ile eğitiminden elde edilen beklenen ve çıkış eğrileri	34
Şekil 3.2. CGD' nin ANI ile eğitiminden elde edilen beklenen ve çıkış eğrileri....	35
Şekil 3.3. QN' nin ANI ile eğitimimden elde edilen eğriler	36
Şekil 3.4. LMQN' nin ANI ile eğitimden elde edilen eğriler	37

Şekil	Sayfa
Şekil 3.5. LM' nin ANI ile eğitiminden elde edilen beklenen ve çıkış eğrileri	38
Şekil 3.6. OBP' nin ANI ile eğitiminden elde edilen beklenen ve çıkış eğrileri	39
Şekil 3.7. BBP' nin ANI ile eğitiminden elde edilen beklenen ve çıkış eğrileri.....	40
Şekil 3.8. CPEP sonuç grafiği.....	41
Şekil 3.9. QP' nin ANI ile eğitiminden elde edilen beklenen ve çıkış eğrileri.....	43
Şekil 3.10. CGD eğitimi sonucu elde edilen beklenen ve çıkış eğrileri.....	44
Şekil 3.11. QN' nin ANI ile eğitiminden elde edilen beklenen ve çıkış eğrileri	45
Şekil 3.12. LMQN' nin eğitiminden elde edilen eğriler	46
Şekil 3.13. LM' nin ANI ile eğitiminden elde edilen beklenen ve çıkış eğrileri	47
Şekil 3.14. OBP' nin ANI ile eğitiminden elde edilen beklenen ve çıkış eğrileri ..	48
Şekil 3.15. BBP' nin ANI ile eğitiminden elde edilen beklenen ve çıkış eğrileri...	49
Şekil 3.16. CPEP' in eğitiminden elde edilen beklenen ve çıkış eğrileri.....	50

RESİMLERİN LİSTESİ

Resim	Sayfa
Resim 2.1. Termoelektrik başlık	5

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış bazı simgeler ve kısaltmalar açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler	Açıklama
X_i	Toplam giriş
W_{ij}	Bağlantılar üzerindeki ağırlıkları
NET_j	Hesaplanmış bir ağırlıklı toplamıdır.
C_m	Ağın çıkışı
B_m	Beklenen çıkış
E_m	İterasyonun hatası
λ	Öğrenme katsayısını
α	Momentum
β	Eşik değer ağırlıkları
ΔA^i	Ağırlıkların değişimi
δ^a	Hata terimi
$S(n)$	Mevcut iterasyon boyunca hesaplanan eğim;
$S(n-1)$	Bir önceki iterasyonun eğimidir
μ	Marquardt parametresi
H	Hessian matrisini
J	Jakobiyen matris

Kısaltmalar	Açıklama
TEC	Termoelektrik Soğutucu
YSA	Yapay Sinir Ağları
ÇKA	Çok Katmanlı Algılayıcı
ANI	Alyuda NeuroIntelligence

TH	Sıcak Yüzey Sıcaklığı
I	Termoelektrik Modüllere Uygulanan Akım
Tc	Soğuk Yüzey Sıcaklığı
E	Termoelektrik Modül Uçlarında İndüklenen Emk
K	Korelasyon Katsayısı
R	Regresyon Analizi
CPEP	C# Dilinde Kodlanmış Bir Eğitim Programı
QP	Quick Propagation
CGD	Conjugate Gradient Descent
QN	Quasi-Newton
LMQN	Limited Memory Quasi-Newton
LM	Levenberg Marquardt
OBP	Online Back Propagation
BBP	Batch Back Propagation
W	Ağın İlk Ağırlık Değerleri
AE	Mutlak Hata
ÖK	Öğrenme Katsayısı
M	Momentum
EGT	Eğitim Setinden Bir Giriş Olduğu
TST	Test Setinden Bir Giriş

1.GİRİŞ

Termoelektrik soğutucular (TEC), bir nesnenin sıcaklığını çevre sıcaklığının altına düşürürken, çevredeki sıcaklık ne olursa olsun, nesne sıcaklığını dengede tutarlar. Termoelektrik soğutucular ısı transfer elemanlarının aktif bir soğutma sistemi olup, miliwatt'tan kilowatt'a kadar değişen bir yelpazedeki uygulamalar için kullanılabilir [1].

Termoelektrik modüllerden oluşan bir sistemin yapay sinir ağları ile eğitilmesinde termoelektrik başlıktan elde edilen deney verileri kullanılmıştır. Böylece en başarılı eğitim ağ yapısı bulunarak benzer sistemlerde yaşanan sistemin çıkışını olumsuz etkileyen ortamın sıcaklığı, akım, gerilim gibi faktörler sonucu yanlış ölçümler elde edilebilmesi. Ayrıca ölçüm seti için yapılan deneylerin zaman alması, bu verilere göre sistem davranışını yorumlamanın zor olması ve ölçüm sonuçlarının kontrol edilmesi için defalarca ölçüm yapılması gibi olumsuzlukların önüne geçmek istenmiştir [1].

Hipotermiya, kafa travması sonucunda veya açık kalp ameliyatı esnasında metabolik hızı ve oksijen ihtiyacını azaltarak organların korunmasını sağlayan bir tedavi yöntemidir. Beyin hipotermiyasındaki temel problem beynin hızla (ilk 1 saat) gereken sıcaklığa kadar soğutulması ve bu sıcaklıkta kontrollü bir şekilde uzun süre tutulmasıdır. Kabul edilebilir sıcaklık hatası $\pm 0,5$ °C'dir [2-4]. Beyin hipotermiyasını sağlamak amacıyla 120 adet esnek termoelektrik modülden oluşan bir başlık tasarımı yapılmıştır. Bu başlığın sıcaklık kontrolünü sağlamak üzere 4 adet modülden oluşan özel bir ölçüm seti geliştirilerek deneyler yapılmıştır [1]. Deney verileri kullanılarak en başarılı ağ yapısı bulunmaya çalışılmıştır. Böylece termoelektrik modüllerden oluşan herhangi bir sistemin davranışını da tahmin etmek amaçlanmıştır.

Yapay sinir ağlarının(YSA) geleceği tahmininde en az geleneksel metotlar kadar, hatta daha iyi, başarılı neticeler verdiği görülmüştür [5-7]. YSA'nın özellikle doğrusal olmayan zaman serilerinde gösterdiği başarı, bir tahmin aracı olarak tercih edilmesini sağlamıştır [8].

Bu tezde, deney setinden elde edilen sonuçlarla Alyuda Neuro Intelligence (ANI) ve C#'da kodlanan eğitim programı (CPEP) kullanılarak termoelektrik başlığın giriş çıkış parametrelerinin benzetimi yapılmıştır. Buna göre, termoelektrik modüllere uygulanan akım (I), sıcak yüzey sıcaklığı(TH), soğuk yüzey sıcaklığı (Tc) parametrelerine göre termoelektrik modül uçlarında indüklenen emk'ya (E) bağlı bir eğitim gerçekleştirilmiştir. Eğitim programı kodsız anlamda termoelektrik sistemin davranışını tanımlamıştır [1].

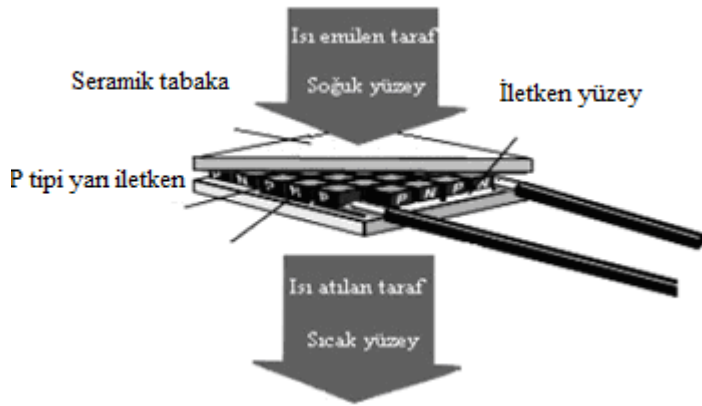
Bu tez 4 bölümden oluşmaktadır. Tezin ikinci bölümünde termoelektrik başlık incelenmekte ve YSA, regresyon, korelasyon ile ilgili genel bilgiler verilmektedir. Daha sonra termo elektrik başlığın giriş çıkış parametrelerinin benzetimi için kullanılan materyal ve yöntemlerden bahsedilmiştir. Üçüncü bölümde elde edilen verilerin analizi, YSA ile eğitim ve test sonuçları bulunmaktadır. Dördüncü bölümde tez bütünüyle sonuç bakımından irdelenmektedir.

2. MATERYAL VE METOT

Tezin bu kısmında TEC ile ilgili genel bilgilerden sonra termoelektrik başlık hakkında detaylı bilgi verilmiştir. Daha sonra TEC eğitimi için kullanılan YSA ile ilgili genel bilgiler ve çok katmanlı algılayıcı (ÇKA) metodundan bahsedilmiştir.

2.1. Termoelektrik ve Bileşenleri

İki metalin birleşim noktasından bir akım geçirildiğinde ısı yutulur, akımın yönü ters çevrildiğinde ise ısı açığa çıkar (Peltier etkisi). Peltier etkisinden faydalanarak termo elementler Şekil 2.1'de gösterildiği gibi n ve p tipi yarıiletkenlerin bir iletkenle elektriksel olarak seri bağlanmasıyla TEC modülü oluşturulur. Peltier soğutucusu, TEC diye de adlandırılabilir.



Şekil 2.1. Sistemin çalışma prensibi

Termoelektrik soğutucular, küçük bir ısı pompası gibi çalışan yarı iletkenlerdir. Bir doğru akım kaynağından sağlanan küçük bir voltaj sayesinde, ısı, modülün bir ucundan diğerine doğru hareket eder. Böylece modülün bir yüzü ısınırken, diğeri de eşzamanlı olarak soğumaya başlar. Bu olay, doğru akım kaynağının artı ve eksi kutuplarının yer değiştirmesiyle tersine çevrilebilir. Bir TEC, kullanım amacına göre ısıtıcı ya da soğutucu olarak kullanılabilir. Eğer soğuk kısma azar azar ısı eklenirse,

bu kısmın sıcaklığı, ısı musluğunun sıcaklığına eşitlenene kadar yükselir. Bu noktada termoelektrik soğutucular, en yüksek ısı pompalama kapasitelerine ulaşırlar [1].

Termoelektrik soğutucular, evde kullandığımız buzdolaplarıyla aynı termodinamik yasalara göre çalışır, ama bazı farklılıklar vardır. Buzdolabında kullanılan dondurucu sıvının yerine, bir yarı iletken; yoğuşturucusunun yerine de ısı transfer elemanı vardır. Ayrıca kompresörün yerini de doğru akım kaynağı alır. Termoelektrik modüle doğru akım kaynağının bağlanması, elektronların yarı iletken nesneden geçmesini sağlar. Maddenin soğuk tarafında, elektron hareketi sayesinde ısı soğurulur ve sıcak uca gönderilir. Sıcak olan uca ısı transfer elemanı bağlandığı için, ısı, ısı transfer elemanından çevreye verilir. Onlarca termo element, elektriksel olarak seri, termal olarak paralel bağlanarak farklı amaçlar için çeşitli büyüklüklerde termoelektrik modüller elde edilir. Termoelektrik soğutucu modülleri genellikle n ve p tipi malzeme çiftlerinden eşit sayıda konulmasıyla üretilmektedir. Termoelektrik modüller, hareketli parçası olmayan küçük birer ısı pompasıdır. Termoelektrik modüller genellikle alan sınırlamasının olduğu, güvenilirliğin önemli olduğu ve zararlı soğutucu gazların kullanılmasının istenmediği ortamlarda tercih edilirler. Termoelektrik soğutucu, DA gerilimle çalışmakla beraber akım yönünün değiştirilmesiyle soğutma veya ısıtma rejimine kolayca geçebilmektedir. Soğutma, termoelektrik modülün bir yüzeyindeki ısının diğer yüzeye taşınmasıyla gerçekleşmektedir [9-12].

2.1.1. Termoelektrik başlık

Termoelektrik cihazların yapımında ilk olarak esnek termobatarya tasarlanmıştır. Esnek birleşim tekniği sıcaklık farkının meydana getirdiği termo mekanik gerilimleri sıfıra indirerek termobataryanın sağlamlığını ve garantili çalışma süresinin artırılmasını sağlamaktadır. Esnek termobataryanın tıpta kullanılmasının önemli avantajı da bu bataryanın ayrı ayrı esnek modüllerden oluşması dolayısıyla soğutulacak alanın şekline kolayca getirilebilmesidir. Tasarlanan termoelektrik soğuk başlıkta 120 adet termoelektrik modül bulunmaktadır. Bu modüllerin çalışma

gerilimleri 0,1 V ve çektikleri akım 40 amperdir. Modüllerin seri bağlandığı tüm alt birimleri sürebilmek için 12 V DC gerilime ihtiyaç vardır.



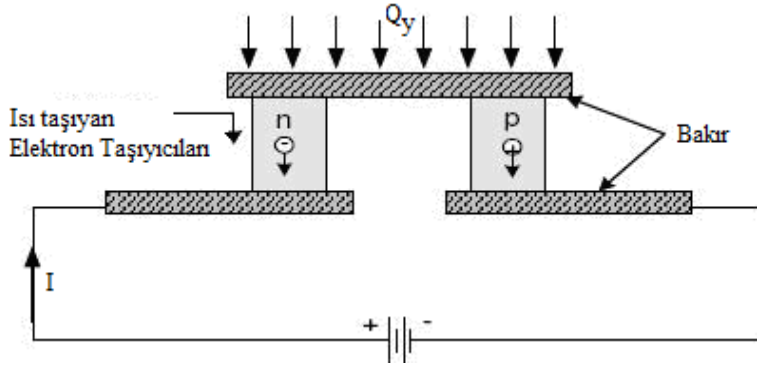
Resim 2.1. Termoelektrik başlık

Resim 2.1'deki termoelektrik başlığın sıcaklığını kontrol etmek için sıcak yüzey tarafındaki sıcaklığın bir ısı atılım sistemi tarafından belirli bir sıcaklıkta tutulması gerekir. Bu nedenle termoelektrik başlığın sıcak tarafı su kanalları ile donatılmıştır. Sistemden ayrı bir su çevrim sistemi ile su kanalları içerisinde iklimlendirilmiş su dolaştırılarak sıcak tarafındaki sıcaklık belirli bir sıcaklıkta sabitlenmektedir. Kapalı su çevrim sistemi belirli bir sıcaklık değerine set edilerek su kanalları içerisine bir su motoru tarafından pompalanmakta, ısınıp gelen su ise tekrar iklimlendirildikten sonra sisteme tekrar pompalanmaktadır [1].

2.1.2. Deney düzeneği

Cihazda kullanılan modüllerin özellikleri araştırılmıştır. Bunun için dört adet modülden oluşan ve Şekil 2.2'de gösterilen özel bir ölçüm seti geliştirilmiştir.

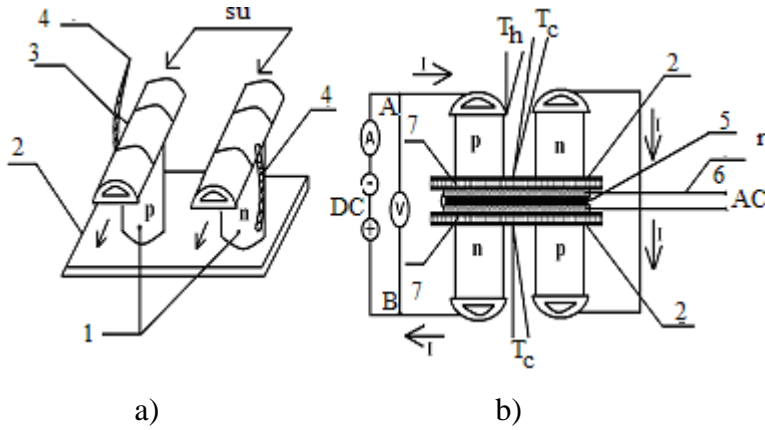
Deney setini oluşturan 4 adet modül birbiriyle elektriksel olarak seri, termal olarak da paralel bağlanmıştır. Modüllerin soğuk yüzeyleri yüz yüze olup aralarında ısı yüklerini oluşturan bir ısıtıcı bulunmaktadır. Modüllerin soğuk yüzeyleri ile ısıtıcının arasına elektriği geçirmeyen anodiroze edilmiş levhalar konulmuştur. Bu levhalar modülleri elektriksel olarak ısıtıcıdan yalıtılmaktadır. Bununla beraber gelen ısıyı homojen olarak dağıtmak için ince bakır levhalar kullanılmıştır. Soğuk ve sıcak



Şekil 2.2. Termoelektrik sistem blok diyagramı

yüzeydeki sıcaklıkları direk ölçmek için çift termo kuplu CIE 307 model dijital termometre kullanılmıştır. Termoelektrik modüllere uygulanan akım Fluke 380 model multimetre ile ölçülmüştür. Aynı modelde ikinci multimetre modüller üzerindeki voltajı ve bunların ürettiği termoemk'yı ölçmek için kullanılmıştır. Ayrıca modülleri çalıştırmak için DA 0 – 50A'lık bir güç kaynağı kullanılırken ısıtıcıyı çalıştırmak için AA 0 –50V'luk bir varyak kullanılmıştır [1].

Şekil 2,3'deki 1 – p ve n-tipi yarıiletkenler, 2 – bakır levalardan yapılmış modüllerin soğuk yüzeyleri, 3 - bakır borulardan yapılmış modüllerin sıcak yüzeyleri, 4 - modülün voltaj girişi için kullanılan çok telli bakır kabloyu, 5 – üzerine ısıtıcı telinin sarıldığı anodiroze edilmiş alüminyum levha, 6 -ısıl yükü oluşturmak için ısıtıcının çıkışlarını, 7 –ısıyı homojen dağıtan inçe bakır levhayı, T_c ve T_h - soğuk yüzeyin ve sıcak yüzeyin sıcaklığını ölçmek için K tipi termokuplları, A ve V ampermetre ve voltmetreyi göstermektedir. Modüllerin ısınan yüzeylerini soğutmak için paralel seri olarak birbirleriyle bağlanmış boru kanallardan geçen sudan faydalanılmıştır. Su üstteki modüllerin borularına paralel girip alttakilerden paralel olarak çıkmaktadır. Şekil 2.3'deki düzeneği kullanarak modüllerin dinamik özelliklerinin araştırması önce yüksüz daha sonra ısıtıcıyı çalıştırarak yüklü yapılmaktadır [13].



Şekil 2.3. a. Tek bir modülün yapısı
b. Deney düzeneği (4 adet modül)

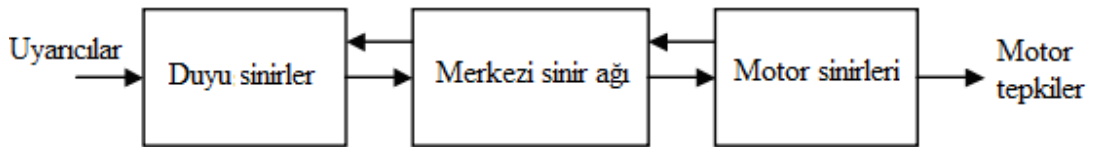
2.2. Yapay Sinir Ağları ve Bileşenleri

YSA, insan beyninin özelliklerinden olan öğrenme yolu ile yeni bilgiler türetebilme, yeni bilgiler oluşturabilme ve keşfedebilme gibi yetenekleri herhangi bir yardım almadan otomatik olarak gerçekleştirmek için geliştirilen bilgisayar sistemleridir. Bu yetenekleri geleneksel programlama yöntemleri ile gerçekleştirmek oldukça zordur veya mümkün değildir. O nedenle, YSA'nın, programlanması çok zor veya mümkün olmayan olaylar için geliştirilmiş adaptif bilgi işleme ile ilgilenen bilgisayar bilim dalıdır [14].

YSA, olayların örneklerine bakmakta, onlardan ilgili olay hakkında genellemeler yapmakta, bilgiler toplamakta ve daha sonra hiç görmediği örnekler ile karışılışınca öğrendiği bilgileri kullanarak o örnekler hakkında karar verebilmektedir. 1990'lı yıllardan beri bilgisayarların öğrenmesini sağlayan YSA teknolojisinde oldukça hızlı bir gelişme görüldü. YSA'nın, insan beyninin özelliklerinden olan öğrenme yolu ile yeni bilgiler türetebilme, yeni bilgiler oluşturabilme ve keşfedebilme gibi yetenekleri vardır. Bir yardım almadan otomatik olarak bu yetenekleri gerçekleştirmek amacıyla geliştirilen bilgisayar sistemleri olmaları nedeniyle hem yeni gelişmeler sağlıyor hem de nasıl çalıştığı bilinmeyen insan beyni hakkında yapılan araştırmalara da önemli katkı sağlamaktadır [15].

2.2.1. İnsanda sinir sistemi

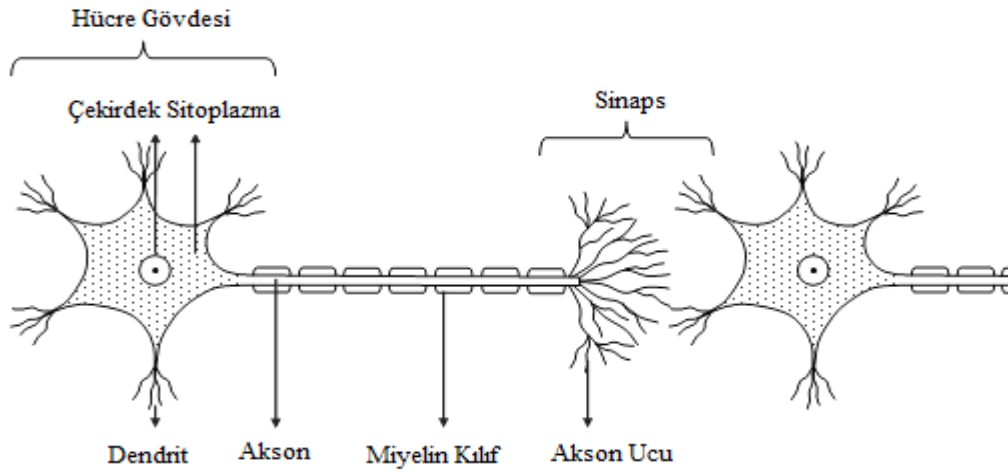
Biyolojik sinir sistemi, merkezinde sürekli olarak bilgiyi alan, yorumlayan ve uygun bir karar üreten beyin (merkezi sinir ağı) bulunduğu 3 katmanlı bir sistem olarak açıklanır. Alıcı sinirler organizma içerisinde ya da dış ortamlardan algıladıkları uyarıları, beyne bilgi ileten elektriksel sinyallere dönüştürür. Tepki sinirleri ise, beyinin ürettiği elektriksel darbeleri organizma çıktısı olarak uygun tepkilere dönüştürür. Şekil 2.4’de bir sinir sisteminin blok gösterimi verilmiştir.



Şekil 2.4. Biyolojik sinir sisteminin blok gösterimi

Merkezi sinir ağında bilgiler, alıcı ve tepki sinirleri arasında ileri ve geri besleme yönünde değerlendirilerek uygun tepkiler üretilir. Bu yönüyle biyolojik sinir sistemi, kapalı çevrim denetim sisteminin karakteristiklerini taşır [16].

YSA’deki işleme elemanları biyolojik olarak insan beynindeki nöronlara karşılık gelmektedir. Şekil 2.5’de bir sinir hücresi görülmektedir. Dendrit olarak adlandırılan yapı, diğer hücrelerden bilgiyi alan nöron girişleri olarak görev yapar. Diğer hücelere bilgiyi transfer eden eleman aksonlardır. Dolayısıyla aksonlar nöron çıkışları olarak görev yaparlar. Akson ile dendrit arasındaki bağlantı ise sinapslar tarafından gerçekleştirilir [17].



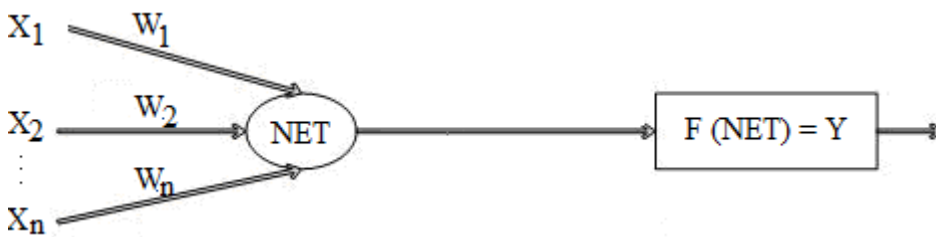
Şekil 2.5. Nöronun biyolojik modeli

2.2.2 Yapay sinir hücresi

YSA'nın işleyişi insan sinir hücresine benzemektedir. 1940 yılında McCulloch ve Pitts nöronun, mantık sistemlerinde basit eş değer yapısıyla modellenebileceğini ortaya atmışlardır. Bu amaçla yaptıkları çalışmalar sonunda Şekil 2.6'da görüldüğü gibi bir yapay sinir ağı modeli geliştirmişlerdir [17].

Bir yapay sinir hücresi beş bölümden oluşmaktadır;

1. Girdiler
2. Toplama fonksiyonu
3. Aktivasyon fonksiyonu
4. Çıktılar



Şekil 2.6. Yapay sinir hücresi

Girdiler, nöronlara gelen verilerdir. Girdiler yapay sinir hücresine bir diğer hücreden gelebileceği gibi direk olarak dış dünyadan da gelebilir. Bu girdilerden gelen veriler biyolojik sinir hücrelerinde olduğu gibi toplanmak üzere nöron çekirdeğine gönderilir [17-18].

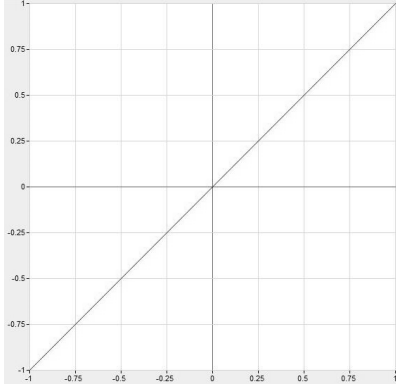
Ağırlıklar, yapay sinir hücresine gelen bilgiler girdiler üzerinden çekirdeğe ulaşmadan önce geldikleri bağlantıların ağırlığıyla çarpılarak çekirdeğe iletilir. Bu sayede girdilerin üretilen çıktı üzerindeki etkisi ayarlanabilmektedir. Bu ağırlıkların değerleri pozitif, negatif veya sıfır olabilir. Ağırlığı sıfır olan girdilerin çıkış üzerinde herhangi bir etkisi olmamaktadır [18].

Toplama fonksiyonu, Şekil 2.6’da görüldüğü gibi herhangi bir katmandaki i. birime gelen toplam giriş (X_i), ilk katman için girişlerin bağlantılar üzerindeki ağırlıkları (W_{ij}) ile hesaplanmış bir ağırlıklı toplamdır (NET_j).

$$NET_j = \sum_{i=1}^N X_{ij} * W_{ij} \quad (2.1)$$

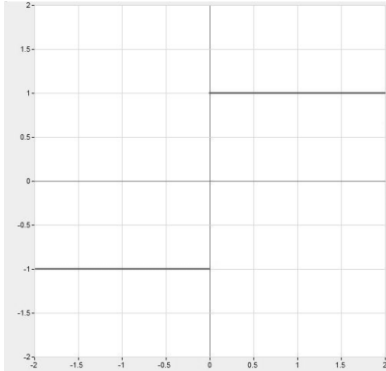
Aktivasyon fonksiyonu, birleştirme (toplama) fonksiyonundan çıkan NET toplam hücrenin çıktısını oluşturmak üzere aktivasyon fonksiyonuna iletilir. Aktivasyon fonksiyonu genellikle doğrusal olmayan bir fonksiyon seçilir. YSA’nın bir özelliği olan “doğrusal olmama” aktivasyon fonksiyonlarının doğrusal olmama özelliğinden gelmektedir. Aktivasyon fonksiyonu seçilirken dikkat edilmesi gereken bir diğer nokta ise fonksiyonun türevinin kolay hesaplanabilir olmasıdır. Bir problem için en uygun fonksiyon tasarımcının denemeleri sonucu belirlenebilir. Uygun fonksiyonu gösteren bir formül bulunmuş değildir [15,18].

Doğrusal aktivasyon fonksiyonu, doğrusal problemler çözmek amacıyla aktivasyon fonksiyonu doğrusal bir fonksiyon da seçilebilir. Doğrusal aktivasyon fonksiyonları matematiksel olarak $F(x) = A * x$ olarak genellenebilir. Bu formülde A sabit bir katsayıdır. A değerinin değişimi şekilde gösterilen doğrunun çıkış eksenine yaptığı açığı değiştirmektedir (Şekil 2.7).



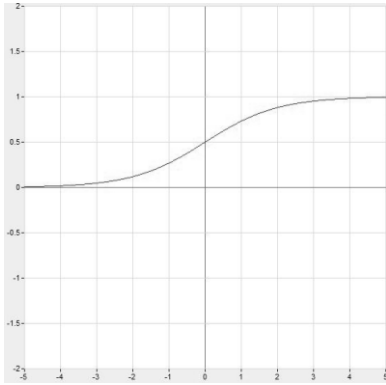
Şekil 2.7. Doğrusal aktivasyon fonksiyonu

Adım aktivasyon fonksiyonu, girdilerin sıfırdan büyük olup olmasına göre -1 veya 1 çıktısı veren fonksiyondur. Sadece iki çeşit çıktı vermektedir (Şekil 2.8).



Şekil 2.8. Adım aktivasyon fonksiyonu

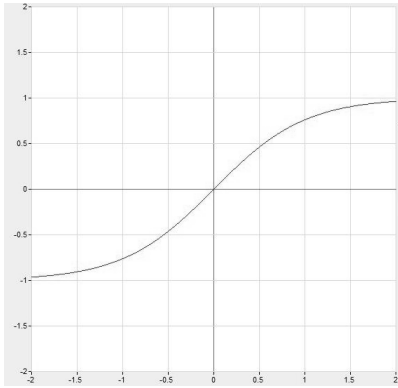
Sigmoid aktivasyon fonksiyonu, sürekli ve türevi alınabilir bir fonksiyondur. Şekil 2.9'da görüldüğü gibi doğrusal olmayışı nedeniyle yapay sinir ağı uygulamalarında en sık kullanılan fonksiyondur. Bu fonksiyon girdi değerlerinin her biri için 0 ile 1 arasında bir değer üretir. Günümüzde en yaygın olarak kullanılan Çok katmanlı Algılayıcı modelinde genel olarak sigmoid fonksiyonu kullanılmaktadır. Bu fonksiyon "Eş. 2.2" de gösterilmektedir [15,18].



$$F(NET) = \frac{1}{1 + e^{-NET}} \quad (2.2)$$

Şekil 2.9. Sigmoid aktivasyon fonksiyonu

Tanjant hiperbolik aktivasyon fonksiyonu, Tanjant hiperbolik fonksiyonu, Şekil 2.10'da görüldüğü gibi sigmoid fonksiyonuna benzer bir fonksiyondur. Sigmoid fonksiyonunda çıkış değerleri 0 ile 1 arasında değişirken hiperbolik tanjant fonksiyonunun çıkış değerleri -1 ile 1 arasında değişmektedir (Eşitlik 2.3).



$$F(NET) = \frac{(e^{NET} + e^{-NET})}{(e^{NET} - e^{-NET})} \quad (2.3)$$

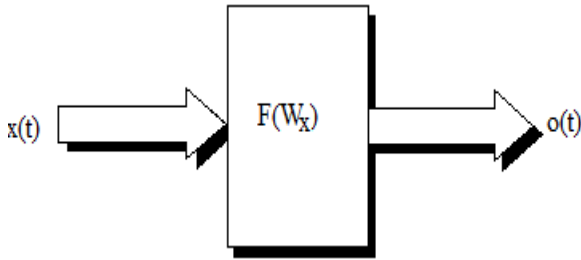
Şekil 2.10. Tanjant hiperbolik aktivasyon fonksiyonu

Çıktılar, aktivasyon fonksiyonundan çıkan değer nöronun çıktı değeri olmaktadır. Bu değer ister yapay sinir ağının çıktısı olarak dış dünyaya verilir ister tekrardan ağın içinde kullanılabilir. Nöronun bir çıktısı olmasına rağmen bu çıktı istenilen sayıda nörona bağlı olabilir [19].

2.2.3. Yapılarına göre yapay sinir ağları

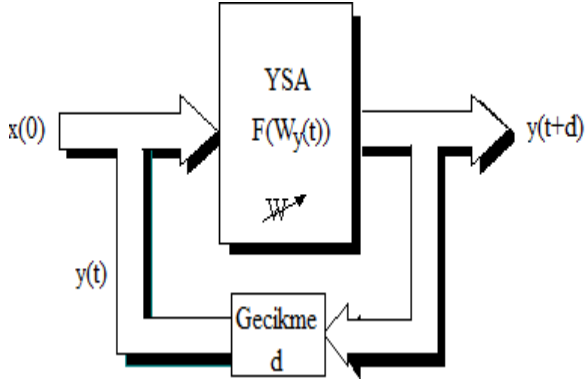
YSA içerdği nöronların birbirine bağlantı şekline göre ileri ve geri beslemeli olarak ikiye ayrılır.

İleri beslemeli ağlar, ileri beslemeli ağlarda nöronlar girişten çıkışa doğru düzenli katmanlar Şekil 2.11'deki gibidir. Bir katmandan sadece kendinden sonraki katmanlara bağ bulunmaktadır. Yapay sinir ağına gelen bilgiler giriş katmanına daha sonra sırasıyla ara katmanlardan ve çıkış katmanından işlenerek geçer ve daha sonra dış dünyaya çıkar [18,19].



Şekil 2.11. İleri beslemeli ağ blok diyagramı

Geri beslemeli ağlar, geri beslemeli YSA'da ileri beslemeli olanların aksine bir nöronun çıkışı sadece kendinden sonra gelen nöron katmanına girdi olarak verilmez. Kendinden önceki katmanda veya kendi katmanında bulunan herhangi bir nörona girdi olarak bağlanabilir. Bu yapısı ile geri beslemeli YSA doğrusal olmayan dinamik bir davranış göstermektedir. Geri besleme özelliğini kazandıran bağlantıların bağlantı şekline göre geri aynı yapay sinir ağıyla farklı davranışta ve yapıda geri beslemeli YSA elde edilebilir (Şekil 2.12) [18].

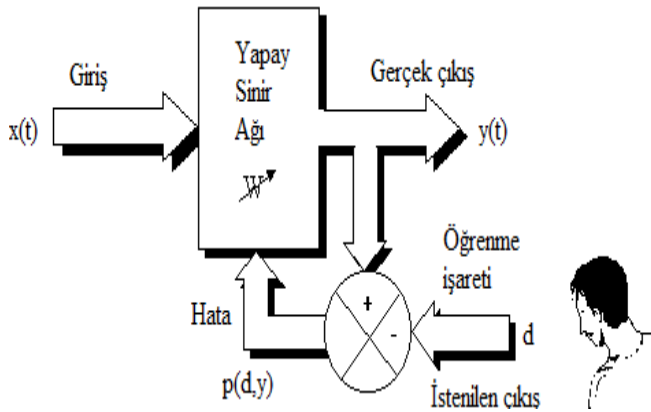


Şekil 2.12. Geri beslemeli ağ blok diyagramı

2.2.4. Öğrenme algoritmalarına göre yapay sinir ağları

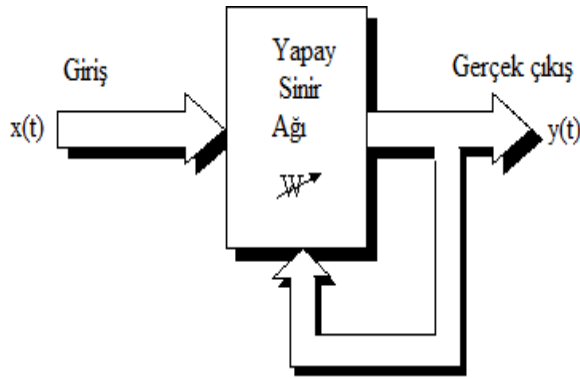
YSA'nın verilen girdilere göre çıktı üretebilmesinin yolu ağı öğrenbilmesidir. Bu öğrenme işleminin de birden fazla yöntemi vardır. YSA öğrenme algoritmalarına göre danışmanlı, danışmansız ve takviyeli öğrenme olarak üçe ayrılır.

Danışmanlı öğrenme, danışmanlı öğrenme sırasında Şekil 2.13'de görüldüğü gibi ağa verilen giriş değerleri için çıktı değerleri de verilir. Ağ verilen girdiler için istenen çıktıları oluşturabilmek için kendi ağırlıklarını günceller. Ağı çıktuları ile beklenen çıktılar arasındaki hata hesaplanarak ağı yeni ağırlıkları bu hata payına göre düzenlenir. Hata payı hesaplanırken ağı bütün çıktıları ile beklenen çıktıları arasındaki fark hesaplanır ve bu farka göre her nörona düşen hata payı bulunur. Daha sonra her nöron kendine gelen ağırlıkları günceller [15,18,19].



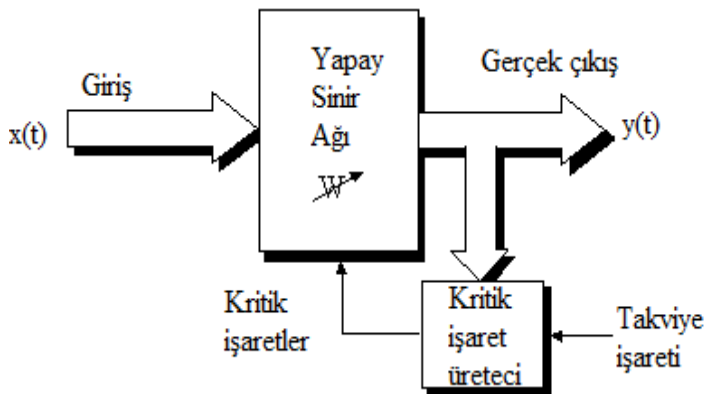
Şekil 2.13. Danışmanlı öğrenme blok diyagramı

Danışmansız öğrenme, danışmasız öğrenmede ağa öğrenme sırasında sadece örnek girdiler verilmektedir. Herhangi bir beklenen çıktı bilgisi verilmez. Girişte verilen bilgilere göre ağ her bir örneği kendi arasında sınıflandıracak şekilde kendi kurallarını oluşturur. Ağ bağlantı ağırlıklarını aynı özellikte olan dokuları ayırabilecek şekilde düzenleyerek öğrenme işlemini tamamlar [15,18,19].



Şekil 2.14. Danışmansız öğrenme blok diyagramı

Destekleyici öğrenme, ağın her iterasyon sonunda elde ettiği sonucun iyi veya kötü olup olmadığına dair bir bilgi verilir. Ağ bu bilgilere göre kendini yeniden düzenler. Bu sayede ağ herhangi bir girdi dizisiyle hem öğrenerek hem de sonuç çıkararak işlemeye devam eder



Şekil 2.15. Destekleyici öğrenme blok diyagramı

Örneğin satranç oynayan bir yapay sinir ağı yaptığı hamlenin iyi veya kötü olduğunu anlık olarak ayırt edememesine rağmen yine de hamleyi yapar.

2.2.5. Öğrenme zamanına göre yapay sinir ağları

YSA öğrenme zamanına göre de statik ve dinamik öğrenme olarak ikiye ayrılır.

Statik öğrenme, statik öğrenme kuralıyla çalışan YSA kullanılmadan önce eğitilmektedir. Eğitim tamamlandıktan sonra ağı istenilen şekilde kullanılabilir. Ancak bu kullanım sırasında ağın üzerindeki ağırlıklarda herhangi bir değişiklik olmaz [15].

Dinamik öğrenme, dinamik öğrenme kuralı ise YSA'nın çalıştığı süre boyunca öğrenmesini öngörerek tasarlanmıştır. Yapay sinir eğitim aşaması bittikten sonra da daha sonraki kullanımlarında çıkışların onaylanmasına göre ağırlıklarını değiştirerek çalışmaya devam eder [19].

2.2.6. Yapay sinir ağlarının ağ yapıları

YSA'nın tasarımı sürecinde ağ yapısının seçilmesi, uygulama problemine bağlı olarak seçilmelidir. Hangi problem için hangi ağın daha uygun olduğunun bilinmesi önemlidir. Kullanım amacı ve o alanda başarılı olan ağ türleri Çizelge 2.1'de verilmiştir [18].

Çizelge 2.1. Ağ türleri ve başarılı oldukları alanlar

Kullanım amacı	Ağ türü	Ağın kullanımı
Tahmin	ÇKA	Ağın girdilerinden bir çıktı değerinin tahmin edilmesi
Sınıflandırma	LVQ ART Counterpropagation Olasılıklı Sinir Ağları	Girdilerin hangi sınıfa ait olduklarının belirlenmesi
Veri ilişkilendirme	Hopfield Boltzman Machine Bidirectional Associative Memory	Girdilerin içindeki hatalı bilgilerin bulunması ve eksik bilgilerin tamamlanması

Uygun YSA yapısının seçimi, büyük ölçüde ağda kullanılması düşünülen öğrenme algoritmasına da bağlıdır. Ağda kullanılacak öğrenme algoritması seçildiğinde, bu algoritmanın gerektirdiği mimaride zorunlu olarak seçilmiş olacaktır. Örneğin geri yayılım algoritması ileri beslemeli ağ mimarisi gerektirir. Bir YSA'nın karmaşıklığının azaltılmasında en etkin araç, YSA ağ yapısını değiştirmektir. Gereğinden fazla sayıda işlemci eleman içeren ağ yapılarında, daha düşük genelleme kabiliyeti ile karşılaşılır [18].

2.2.7. Çok katmanlı algılayıcı

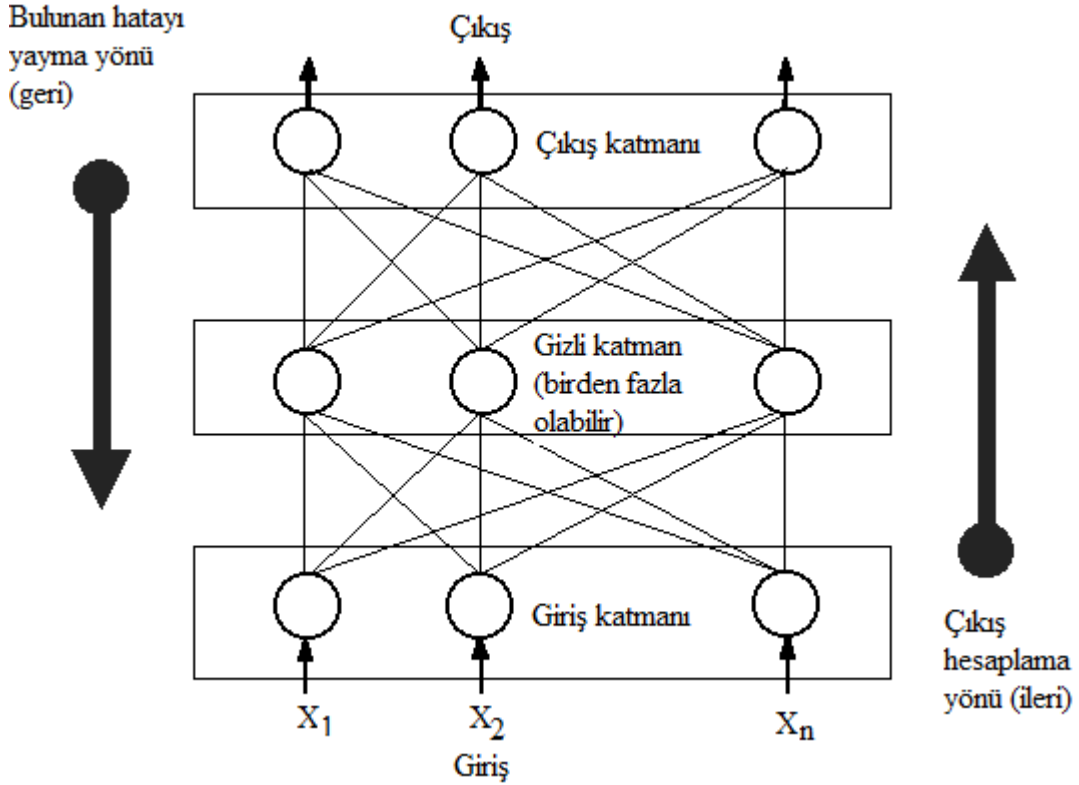
Bu modelin yapısı Şekil 2.16'da gösterildiği gibidir. Bu ağ modeli özellikle mühendislik uygulamalarında çok kullanılan sinir ağı modelidir. Birçok eğitim algoritmasının bu ağı eğitmede kullanılabilir olması, bu modelin yaygın kullanılmasının sebebidir. Bir ÇKA modeli, bir giriş, bir veya daha fazla ara ve bir de çıkış katmanından oluşur. Bir katmandaki bütün işlem elemanları bir üst katmandaki bütün işlem elemanlarına bağlıdır. Bilgi akışı ileri doğru olup geri besleme yoktur. Bunun için ileri beslemeli sinir ağı modeli olarak adlandırılır. Giriş katmanında herhangi bir bilgi işleme yapılmaz. Buradaki işlem elemanı sayısı tamamen uygulanan problemler giriş sayısına bağlıdır. Ara katman sayısı ve ara katmanlardaki işlem elemanı sayısı ise, deneme-yanılma yolu ile bulunur. Çıkış katmanındaki eleman sayısı ise yine uygulanan probleme dayanılarak belirlenir [18-20].

ÇKA ağlarında, ağa bir örnek gösterilir ve örnek neticesinde nasıl bir sonuç üreteceği de bildirilir (danışmanlı öğrenme). Örnekler giriş katmanına uygulanır, ara katmanlarda işlenir ve çıkış katmanından da çıkışlar elde edilir. Kullanılan eğitim algoritmasına göre, ağın çıkışı ile arzu edilen çıkış arasındaki hata tekrar geriye doğru yayılarak hata minimuma düşünceye kadar ağın ağırlıkları değiştirilir [18-20].

İleri beslemeli ağlar, en genel anlamıyla giriş uzayıyla çıkış uzayı arasında statik haritalama yapar. Bir andaki çıkış, sadece o andaki girişin bir fonksiyonudur [20].

ÇKA'nın öğrenme kuralı en küçük kareler yöntemine dayalı Delta öğrenme kuralının genelleştirilmiş halidir. O nedenle öğrenme kuralına Genelleştirilmiş Delta Kuralı da

denir. Ağın öğrenilmesi için örneklerden oluşan bir sete ihtiyaç vardır. Bu set içinde her örnek için ağın hem girdiler hem de o girdiler için ağın üretmesi gereken çıktılar belirlenmiştir. Bu öğrenme kuralı ileri doğru hesaplama ve geriye doğru hesaplama olmak üzere iki safhadan oluşur (Şekil 2.16) [15,20].



Şekil 2.16. Geri yayılım ÇKA yapısı

İleri doğru hesaplamada ağın çıktısı hesaplanır. İşlem Girdi katmanından başlar. Örnek setinden bir örnek ağa Girdi (X_1, X_2, \dots) katmanından gösterilir ve hiç bir değişiklik olmadan ara katmana gönderilir.

Ara katmandaki her nöron elemanı girdi katmanındaki bütün nöron elemanlarından gelen bilgileri bağlantı ağırlıkları (W_1, W_2, \dots) yolu ile alır. Önce ara katmandaki nöron elemanlarına gelen net girdi hesaplanır [18].

Net girdisi hesaplanan nöronun probleme en iyi sonuç vereceği bir aktivasyon fonksiyon seçilir. Genellikle ÇKA'da Sigmoid fonksiyonu iyi sonuç vermeye

birlikte kullanılması zorunlu değildir. Önemli olan türevi alınabilir bir fonksiyon kullanmaktır. Sigmoid Fonksiyonu kullanılması halinde çıktı (Eşitlik 2.4);

$$C_j^a = \frac{1}{1 + e^{-(NET_j^a + \beta_j^a)}} \quad (2.4)$$

“Eş. 2.4” deki β_j , j nolu ara katmana gelen eşik değer elemanın ağırlığını göstermektedir. Bu eşik değeri ünitesinin çıktısı sabit olup 1'e eşittir. Ağırlık değeri ise sigmoid fonksiyonunun oryantasyonunu belirlemek üzere konulmuştur. Eğitim esnasında ağ bu değeri kendisi belirlemektedir [15].

Ağın çıkışı(C_m) hesaplandıktan sonra beklenen çıkış(B_m) ile karşılaştırılır. Bulunan sonuç o iterasyonun hatasıdır (E_m) buna mutlak hata da denebilir(AE)(Eşitlik 2.5).

$$E_m = B_m - C_m \quad (2.5)$$

Çıktı katmanı için oluşan toplam hatayı(TH) bulmak için “Eş. 2.6” kullanılır [26,27].

$$TH = \frac{1}{2} \sum_m E_m^2 \quad (2.6)$$

Geriye doğru hesaplamada beklenen değer alınan çıktıya eşit değilse, geriye doğru hesaplama yapılarak ağırlık değerleri değiştirilir. ÇKA ağlarının eğitime felsefesi diğer ağlarından farklı değildir. Ağ kendisine gösterilen girdi örneği için beklenen çıktıyı üretmesini sağlayacak ağırlık değerleri bulunmaya çalışılır. Başlangıçta bu değerler rastgele olarak atanır ve ağ örnekleri gösterdikçe ağın ağırlıkları değiştirilerek zaman içerisinde istenen değerlere ulaşması sağlanır. İstenen ağırlık değerlerinin ne olduğu bilinmemektedir. Bu nedenle YSA'nın davranışlarını yorumlamak ve açıklamak mümkün olmamaktadır. Zaten diğer yapay zeka tekniklerinden mesela uzman sistemlerden en önemli farkı da davranışlarını açıklayamamasıdır. Bunun temel nedeni bilginin ağ üzerine dağıtılmış olması ve ağırlık değerlerinin tek başlarına herhangi bir anlam göstermemeleridir. Bilinen tek

şey problem uzayında en az hata verebilecek ağırlık değerlerinin bulunduğudır. Ağırlık değişimini hesaplamak için çeşitli algoritmalar geliştirilmiştir [15].

2.2.8. Online geri yayılım algoritması

Ağın ağırlıklarını değiştirmek için iki durum söz konusudur [15-20].

Ara katman ile çıktı katman arasındaki ağırlıkların değiştirilmesi, ara katmandaki j. nöron elemanını çıktı katmanındaki m. nöron elemanına bağlayan bağlantının ağırlığındaki değişim miktarına ΔA^a denirse, herhangi bir t. iterasyonda ağırlığın değişim miktarı ve değişim miktarı hesaplandıktan sonra ağırlıkların t. İterasyonda ki yeni değerleri “Eş. 2.7” ve “Eş. 2.8” gibi olacaktır.

$$\Delta A_{jm}^a(t) = \lambda \delta_m \zeta_j^a + \alpha \Delta A_{jm}^a(t-1) \quad (2.7)$$

$$A_{jm}^a(t) = A_{jm}^a(t-1) + \Delta A_{jm}^a(t) \quad (2.8)$$

Burada λ öğrenme katsayısını, α momentum katsayısını göstermektedir. Öğrenme katsayısı ağırlıkların değişim miktarını, momentum katsayısı ise ağın öğrenme esnasında yerel bir optimum noktaya takılıp kalmaması için ağırlık değişim değerinin belirli bir oranda bir sonraki değişime eklenmesini sağlar. δ_m ise m. çıktı elemanının hatasını göstermektedir. Aktivasyon fonksiyonu sigmoid fonksiyonu olduğundan “Eş. 2.9” deki gibi hesaplanır.

$$\delta_m = \zeta_m (1 - \zeta_m) E_m \quad (2.9)$$

Benzer şekilde eşik değer ünitesinin de ağırlıklarını değiştirmek gerekmektedir. Eğer çıktı katmanında bulunan nöron elemanlarının eşik değer ağırlıkları β ile gösterilirse, değişim miktarı ve eşik ünitesinin t. iterasyondaki ağırlığının yeni değeri “Eş. 2.10” ve “Eş. 2.11” gibi olacaktır:

$$\Delta \beta_{jm}^a(t) = \lambda \delta_m + \alpha \Delta \beta_{jm}^a(t-1) \quad (2.10)$$

$$\beta_{jm}^a(t) = \beta_{jm}^a(t-1) + \Delta\beta_{jm}^a(t) \quad (2.11)$$

Ara katmanlar arası veya ara katman girdi katmanı arasındaki ağırlıkların değiştirilmesi, ara katmanlar arası veya ara katmanı ile girdi katmanları arasındaki ağırlıkların değişimi ΔA^i ile gösterilirse, t. iterasyondaki değişim miktarı “Eş. 2.12” gibi hesaplanır [15]:

$$\Delta A_{kj}^i(t) = \lambda \delta_j^a C_k^i + \alpha \Delta A_{kj}^i(t-1) \quad (2.12)$$

Buradaki δ^a hata terimi, aktivasyon fonksiyonu sigmoid fonksiyonu olduğundan şu şekilde hesaplanacak ve buradan hareketle ağırlıkların yeni değerleri [15]:

$$\delta_j^a = C_j^a (1 - C_j^a) \sum_m \delta_m A_{jm}^a \quad (2.13)$$

$$A_{kj}^i(t) = A_{kj}^i(t-1) + \Delta A_{kj}^i(t) \quad (2.14)$$

“Eş. 2.13” ve “Eş. 2.14” gibi olacaktır. Benzer şekilde, eşik değer ünitesinin yeni ağırlıkları da yukarıdaki gibi hesaplanır. Ara katman eşik değer ağırlıkları β^a ile gösterilirse, değişim miktarı ve ağırlıkların yeni değerleri “Eş. 2.15” ve “Eş. 2.16” deki gibi olacaktır [15]:

$$\Delta\beta_j^a(t) = \lambda \delta_j^a + \alpha \Delta\beta_j^a(t-1) \quad (2.15)$$

$$\beta_j^a(t) = \beta_j^a(t-1) + \Delta\beta_j^a(t) \quad (2.16)$$

2.2.9. Kümesel geri yayılım algoritması

Online Back Propagation formülleri kullanılır tek farkı Online Back Propagation’da her iterasyondan sonra yapılan ağırlık güncellemesi Batch Back Propagation’da tüm eğitim seti (bir epoch) ağa sunulduktan sonra yapılır [21].

2.2.10. Hızlı yayılım algoritması

Bazı uygulamalarda Back Propagation'dan daha hızlı olabilir. Ancak bazen eğitim sırasında bir noktaya takılabilir. Bu nedenle genelde dikkate alınmayan bir tekniktir.

İlk epochda ağırlık güncellemesi Back Propagation'da olduğu gibidir. Diğer epochlarda ise aşağıdaki formül kullanılır(Eşitlik 2.17).

$$\Delta A(n) = \frac{S(n)}{S(n-1) - S(n)} \quad (2.17)$$

“Eş. 2.17” de kullanılan $S(n)$,mevcut iterasyon boyunca hesaplanan eğim; $S(n-1)$ ise bir önceki iterasyonun eğimidir. $S(n)$ değeri $S(n-1)$ değerine çok yaklaşırsa veya daha büyük olması durumunda ise aşağıdaki formül kullanılır(Eşitlik 2.18).

$$\Delta A = a\Delta A(n-1) \quad (2.18)$$

“Eş. 2.18” de kullanılan a , Quick Propagation katsayısıdır [22,23].

2.2.11. Levenberg Marquardt algoritması

Eşlenik gradyan ve Newton yöntemlerinin iyi yanlarını alarak oluşturulan hata fonksiyonunu minimize etme işlemini hızlı gerçekleştiren algoritmalarından biridir. Oldukça hızlı ve çok kullanılan bir yöntemdir. Quasi Newton yöntemi gibi Hessian matrisi yaklaşık değeri olan Jakobiyen matrisini kullanır(Eşitlik 2.19).

$$H(n) = J^T(n)J(n) + \mu I \quad (2.19)$$

Elde edilen yakınsak bu değerle ağırlık ve bias değerleri güncellenir(Eşitlik 2.20).

$$w(n+1) = w(n) - (H)^{-1} J^T(n)e(n) \quad (2.20)$$

“Eş. 2.20” de kullanılan μ Marquardt parametresi Hessian matrisi hesaplanırken kullanılır ve hata fonksiyonu azaldıkça azaltılır [28,29].

2.2.12. Newton öğrenme algoritması

Hızlı eğitim algoritmalarından biridir. Toplam hata fonksiyonunu minimize etmek için hata fonksiyonunun ağırlıklara göre ikinci dereceden türevlerinin alınmasıyla oluşan Hessian matrisini (H) kullanır (Eşitlik 2.21).

$$H = \Delta^2 \varepsilon(w) \quad (2.21)$$

“Eş. 2.22” deki gibi Hessian matrisinin tersi $(H)^{-1}$ ile ağırlıklar güncellenir.

$$w(n+1) = w(n) - (H)^{-1} J^T(n) g(n) \quad (2.22)$$

Çok karmaşık ileri beslemeli bir ağda ikinci dereceden türevlerle işlem yapmak oldukça zor olacaktır. Quasi Newton algoritması ile Hessian matrisinin yaklaşık değeri kullanılarak bu problem aşılr (Eşitlik 2.23).

$$H(n) = J^T(n)J(n) + \delta I \quad (2.23)$$

“Eş. 2.23” de kullanılan Jakobiyen matris (J) ağ hatalarının ağırlıklara göre birinci dereceden türevinin alınması ile belirlenir. Yeni ağırlıklar, Hessian matrisine göre daha kolay hesaplanabildiği için Jakobiyen matris ile yeniden düzenlenir (Eşitlik 2.24).

$$w(n+1) = w(n) - (H)^{-1} J^T(n) e(n) \quad (2.24)$$

Bu yöntem en dik gradyan azalış yöntemlerine göre daha hızlı yakınsama özelliğine sahiptir, Conjugate Gradient kadar hızlı bir öğretim algoritmasıdır [29,30-32].

2.2.13. Normalizasyon

Verileri [-1,1] aralığında normalize etmek için “Eş. 2.25”, “Eş. 2.26” ve “Eş. 2.27” kullanılır.

$$ortdizin = \frac{Y(enbüyükdeğer) + Y(enküçükdeğer)}{2} \quad (2.25)$$

$$diz_{in} = Y(\text{enbüyükdeğer}) - Y(\text{enküçükdeğer}) \quad (2.26)$$

$$Z_i = \frac{Y_i - \text{ortdiz}_{in}}{diz_{in}/2} \quad (2.27)$$

[0,1] aralığında normalize etmek için “Eş. 2.28” kullanılır.

$$Z_i = \frac{Y_i - Y(\text{enküçükdeğeri})}{Y(\text{enbüyükdeğeri}) - Y(\text{enküçükdeğeri})} \quad (2.28)$$

Eşitliklerde kullanılan Y_i , veri setinde normalize dilecek sütunun i . satırı. Z_i , ise Y_i 'nin normalize edilmiş halidir [33].

2.2.14. Korelasyon katsayısı ve regresyon

Korelasyon analizi, iki ya da daha çok değişken arasında ilişki olup olmadığını, ilişki varsa yönünü ve gücünü incelemektedir. Regresyon analizi, değişkenlerden birisi belirli bir birim değiştiğinde diğerinin nasıl bir değişim gösterdiğini incelemektedir. Yapay sinir ağlarının performansını ölçmek için bu yöntemler yaygın olarak kullanılmaktadır.

Korelasyon katsayısı (K) iki değişken arasındaki ilişkinin ölçüsüdür, [-1, 1] arasında değişim gösterir. Korelasyon katsayısı [0,5, 1] arasında ise değişkenler arasında ilişki kuvvetlidir. Negatif yönde elde edilen değerler değişkenler arasında ters yönde ilişki olduğunu ifade eder.

Regresyon analizi (R), bilinen bulgulardan gelecekteki olaylarla ilgili tahminler yapılmasına izin verir. Regresyon, bağımlı ve bağımsız değişkenler arasındaki ilişkiyi ve doğrusal eğri kavramını kullanarak, bir tahmin eşitliği geliştirir. Değişkenler arasındaki ilişki belirlendikten sonra bağımsız değişkenlerin katsayıları bulunursa bağımlı değişkenin değeri tahmin edilebilir.

2.2.15. ÇKA ağının çalışma şekli

a. Örneklerin toplanması ve ağın topolojik yapısının belirlenmesi : İlk olarak ağın çözmesi istenilen olay için daha önce gerçekleşmiş örnekler temin edilir. Örneklerle göre, kaç tane girdi ünitesi, ara katman, her ara katmanda kaç nöron elemanı ve kaç çıktı elemanı olacağı belirlenir. Ağın görevi her girdi için o girdiye karşılık gelen çıktıyı üretmektir.

b. Öğrenme parametrelerinin belirlenmesi : Ağın öğrenme katsayısı, nöron elemanlarının toplama ve aktivasyon fonksiyonları, momentum katsayısı gibi parametreler bu adımda belirlenir.

c. Öğrenme işlemi : Çok Katmanlı YSA'nın öğrenme kuralı en küçük kareler yöntemine dayalı Delta Öğrenme kuralının genelleştirilmiş halidir. Çok Katmanlı Ağa örnekler (girdi, çıktı değerleri) belirli bir düzene göre gösterilip ağın bulunduğu çıktı ile gerçek çıktı değerleri karşılaştırılarak hata değeri hesaplanır. Amaç toplam hatayı minimize etmektir. Hata değerini azaltmak için bağlantı ağırlıkları değiştirilir. Belirli bir iterasyondan sonra öğrenme tamamlanır [15,18].

Yukarıdaki adımlar ÇKA ağının öğrenmesi tamamlanıncaya yani gerçekleşen çıktılar ile beklenen çıktılar arasındaki hatalar kabul edilebilir düzeye ininceye kadar devam eder. Ağın öğrenmesi için bir durdurma kriterinin olması gerekmektedir. Bu ise genellikle üretilen hatanın belirli bir düzeyin altına düşmesi olarak alınmaktadır [15,18].

2.3. Yazılım

Bu uygulamada Ankara Gazi Üniversitesi Teknik Eğitim Fakültesi ARGE laboratuvarında termoelektrik başlık üzerinde yapılan 564 adet deney sonucundan bir veri seti oluşturulmuştur [1]. Termoelektrik başlığın giriş çıkış parametrelerinin benzetiminde YSA kullanılmıştır.

2.3.1. Materyal

Uygulamada kullanılan veri setinden dört özellik seçilmiştir [1]. Bu özelliklerden I , T_h ve E bilgileri giriş sütunu, T_c ise çıkış sütunu olarak belirlenmiştir (Şekil 2.17). Aynı olan veriler veri setinden çıkarılarak veri seti yeniden düzenlenmiştir. Bunun sonucunda 456 adet deney sonucu giriş çıkış parametrelerinin benzetiminde kullanılmıştır. Bu veri seti içinden diğer satırları örnekleyecek 96 adet veri satırı seçilmiştir [1]. Bu 96 deney sonucunu içeren örneklem veri setinin önce %80'i eğitim, kalan %20'si test seti; daha sonra %50'si eğitim, kalan %50'si test seti olarak ayrılmıştır. YSA ile Termoelektrik başlığın sıcaklık tahmininde C# dilinde kodlanmış bir eğitim programı (CPEP) ve Alyuda NeuroIntelligence (ANI) 2.2 (507) programı kullanılmıştır [35-39].



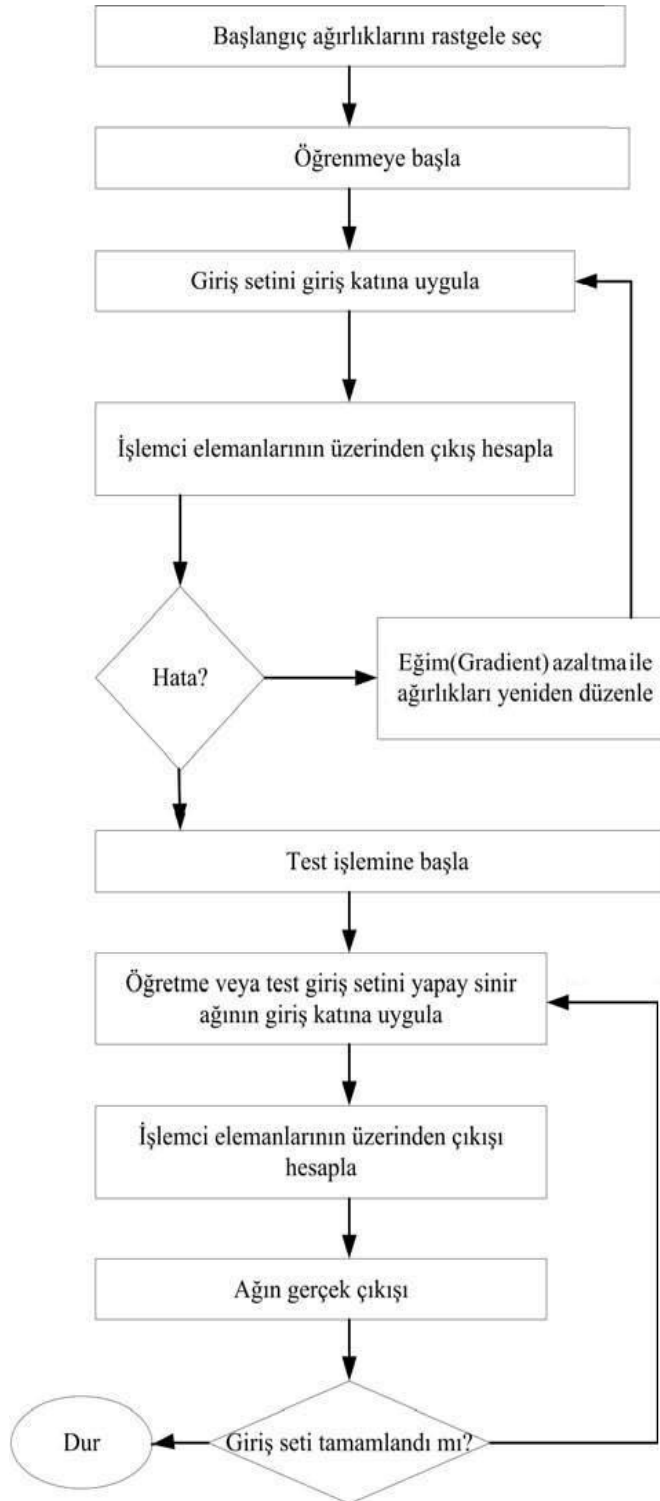
Şekil 2.17. Termoelektrik başlığın blok diyagramı

2.3.2. Metot

Uygulamamızdaki amaç ileri dönük sıcaklık değerlerini tahmin etmek olduğundan ÇKA yapısı kullanılmıştır [18]. Bu nedenle ileri beslemeli, geri yayımlı ağ modeli kullanılmıştır. Ağa eğitim sırasında hem girdiler hem de üretmesi gereken sonuçlar gösterilmiştir (danışmalı öğrenme). Yapılan eğitim denemelerinde hem sigmoid hem de tanjant hiperbolik aktivasyon fonksiyonları kullanılmış ve en iyi sonuç veren fonksiyon eğitimde kullanılmıştır.

Eğitim tamamlandıktan sonra test setinde olan veya olmayan tüm sorgulamalarda ağ üzerindeki ağırlıklarda herhangi bir değişiklik olmaz (statik öğrenme). Çıkış sütunu

olan T_c [0,1], E sütunu [-1,1] aralığında, T_h ve I sütunları ise 1 veya -1 olarak ANI programı ile normalize edilmiştir (Çizelge 2.2).



Şekil 2.18. Geri yayılım algoritması akış diyagramı

ANI programı kullanılarak ağ Quick Propagation (QP), Conjugate Gradient Descent (CGD), Quasi-Newton (QN), Limited Memory Quasi-Newton (LMQN), Levenberg Marquardt (LM), Online Back Propagation (OBP) ve Batch Back Propagation (BBP) algoritmaları kullanılarak en iyi sonuca ulaşılmaya çalışılmıştır. CPEP’de OBP algoritması kullanılmıştır (Şekil 2.18).

Çizelge 2.2. Eğitim ve test setinin normalizasyon sonucu

Kodlanmış sütunlar	Sütun türü	Kodlama türü / Kodlanan veriler	Kodlama sayısı aralığı	Kodlanma sonrası sütun sayısı	Sütun en küçük değeri	Sütun en büyük değeri
I	Giriş	Kategoriksel { 10A,20A,30A,40A }	-1 veya 1	4 sütun	-	-
Th	Giriş	Kategoriksel \ { 15 °C, 25 °C, 35 °C }	-1 veya 1	3 sütun	-	-
E	Giriş	Sayısal	[-1..1]	1 sütun	-8	78
Tc	Çıkış	Sayısal	[0..1]	1 sütun	-37	30,4

ANI programına sırasıyla eğitim, validasyon ve test setleri gösterilmiştir. CPEP’de ise eğitim ve test setleri sırasıyla gösterilmiştir.

Ağın ilk ağırlık değerleri (W) deneme yolu ile ± 1 aralığında rastgele verilmiştir. W eğitimin süresini etkileyen bir unsur olduğu ve istenilen sonucu üreten ağın ağırlıklarına ne kadar yakın bir ilk değer verilirse ağın o kadar kısa sürede beklenen hata değerine ulaştığı görülmüştür. Verilen daha büyük ve daha küçük ağırlık değerlerinde eğitimin daha uzun sürede beklenen hata değerine ulaştığı görülmüştür. Ayrıca ağırlığın rastgele verilmesi nedeniyle aynı özelliklere sahip YSA ağların farklı ama yakın hata sonuçları verebilmektedir.

Ağın topolojisi ANI programında Yapıyı Araştır (Search Architecture) butonu ile gizli katmanda 10-20 nöron arasında iyi sonuç verdiği gözlemlenmiştir. ÇKA’nın giriş katmanında 3 adet giriş sütunu olması nedeniyle 3 adet nöron bulunmaktadır. Gizli katman nöron sayısı 10-20 arasında seçilmiştir. Bir adet çıkış sütunu olması nedeniyle çıkış katmanında 1 nöron bulunmaktadır.

ANI programı kullanarak QP algoritmasının en iyi eğitim sonucunu verecek katsayılarını bulmak için yapılan 3/17/1 topoloji yapısı 1000 iterasyonluk eğitimler yapılmıştır. Her eğitimde aynı topoloji ve katsayıların kullanılmasına karşın her eğitim sonunda birbirine yakın ancak farklı hata değerleri gözlemlenmiştir. Yapılan eğitimlerden bir tanesinin AE sonuçları örnek olarak Çizelge 2.3'de verilmiştir. Yapılan eğitimlerde QPK=1,75 sütunu ile QPK=4 arasındaki hataların 0,50 ile 0,60 arasında değiştiği diğer sütunlarda ise 0,70 ve üzeri hata görülmüştür. Bu nedenle eğitimde QPK değeri 1,75 ile 4 arasında değiştirilerek, öğrenme katsayısı (ÖK) 0,1 ile 1 arasında değiştirilerek en iyi eğitim sonucuna ulaşılmaya çalışılmıştır.

Çizelge 2.3. ANI ile QP algoritması katsayılarında bulunan mutlak hata değerleri

ÖK	QPK'ya göre AE değerleri															
	0,25	0,50	0,75	1	1,25	1,50	1,75	2	2,25	2,50	2,75	3	3,25	3,50	3,75	4
0,1	10,96	8,33	4,36	1,66	0,65	0,57	0,62	0,57	0,60	0,57	0,52	0,55	0,46	0,57	0,54	0,55
0,2	6,00	3,10	2,15	1,36	0,64	0,62	0,54	0,56	0,62	0,57	0,56	0,55	0,55	0,60	0,64	0,52
0,3	3,06	2,25	2,17	1,29	0,65	0,55	0,59	0,56	0,55	0,60	0,56	0,53	0,59	0,54	0,57	0,60
0,4	2,43	2,16	2,13	1,24	0,68	0,59	0,57	0,55	0,59	0,55	0,58	0,54	0,57	0,57	0,61	0,55
0,5	2,26	2,19	2,09	1,12	0,63	0,57	0,64	0,54	0,55	0,53	0,57	0,55	0,56	0,54	0,55	0,59
0,6	2,22	2,14	2,10	1,14	0,63	0,57	0,66	0,57	0,54	0,56	0,54	0,54	0,57	0,59	0,65	0,60
0,7	2,15	2,50	2,13	1,22	0,61	0,55	0,55	0,56	0,57	0,54	0,60	0,54	0,57	0,55	0,60	0,55
0,8	2,11	2,17	2,08	1,03	0,71	0,70	0,62	0,60	0,61	0,64	0,58	0,61	0,62	0,53	0,52	0,60
0,9	2,16	2,15	2,06	1,05	0,74	0,58	0,57	0,58	0,66	0,58	0,54	0,54	0,54	0,53	0,55	0,53
1	2,12	2,11	2,08	1,03	0,68	0,63	0,59	0,56	0,58	0,55	0,54	0,55	0,54	0,63	0,57	0,56

ANI programı kullanarak OBP algoritma katsayıları olan ÖK ve momentumun(M) en iyi eğitim sonucunu veren değerlerini bulmak için 3/14/1 topolojisinde 1000 iterasyonluk eğitimler yapılmıştır. Her eğitimde aynı topoloji ve katsayıların kullanılmasına karşın her eğitim sonunda birbirine farklı ama yakın hata değerleri gözlemlenmiştir. Buna göre, Çizelge 2.4'de de görüldüğü gibi M değeri 0,8 iken ÖK 0,7-1 arasında, M değeri 0,9 değerinde iken de ÖK 0,5-0,9 arasında OBP'nin en az AE sonucu verdiği gözlenmiştir. Bu nedenle eğitimde yukarıda belirtilen ÖK ve M değerleri arasında değişiklik yapılarak en iyi eğitim sonucuna ulaşılmaya çalışılmıştır.

CPEP'de kullanılan OBP algoritma katsayıları olan ÖK ve M'nin en iyi eğitim sonucu veren değerlerini bulmak için 3/14/1 topolojisinde 1000 iterasyonluk

Buna göre, Çizelge 2.6’de görüldüğü gibi M’nin 1 değeri diğer değerlere göre çok daha yüksek hata vermiştir. Diğer sütunların ise daha düşük ve yakın hata değerleri verdiği görülmüştür. Buna göre, M değeri 0,1 ve 0,9 arasında değişirken ÖK değeri de 0,1-1 arasında değiştirilerek en iyi eğitim sonucuna ulaşmaya çalışılmıştır.

Çizelge 2.6. BBP katsayılarının tespit analizi

ÖK	M’ye göre Mutla hata sonuçları									
	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
0,1	2,15	2,18	2,07	2,08	2,08	2,07	2,04	1,89	2,33	20,06
0,2	2,04	2,21	2,02	2,05	1,97	2,23	1,95	2,28	2,26	7,27
0,3	2,11	1,84	1,94	2,02	1,91	1,99	2,16	2,22	2,21	17,07
0,4	2,04	1,98	1,97	1,96	1,95	2,03	2,00	1,96	2,22	8,21
0,5	2,04	2,13	2,05	1,95	1,98	1,99	1,91	2,16	2,00	19,31
0,6	2,08	2,09	2,06	2,03	1,95	1,95	1,89	2,15	2,33	9,50
0,7	2,11	2,09	1,99	2,04	1,96	1,92	1,99	2,16	1,99	19,57
0,8	2,63	2,11	2,09	2,03	1,94	2,26	1,86	2,17	1,95	7,26
0,9	2,09	2,03	2,03	1,95	2,16	1,98	1,95	2,22	1,93	19,78
1	2,21	2,26	2,00	2,05	2,16	1,98	2,04	2,22	2,02	17,37

Durdurma ölçütü olarak iterasyon sayısı kullanılmıştır. Ağın hata değeri ANI programında AE (Absolute Error), Ağ Hatası (Network Error) ve Hata Gelişimi (Error Improvement) grafikleriyle görülebilmektedir. C# ile kodlanmış programda ise toplam hata grafiği ile ağın eğitim sırasındaki hata durumu görülebilmektedir.

Eğitim sonunda test seti de ağa sunulur. Eğitim seti ve test setinin çıkışları beklenen gerçek çıkış değerinden çıkarılarak AE değeri bulunur. Eğitim ve test setinde $AE \leq 0,5$ olan satır sayısı [5-8] ve bulunan en küçük ve en büyük hata değerleri ağın performansını belirlemektedir. Ayrıca korelasyon katsayısı ve regresyonda eğitimde performans ölçütü olarak kullanılmıştır.

3. BULGULAR VE TARTIŞMA

Bu tezde 96 deney sonucu içeren veri setinin önce %80'i eğitim, kalan %20'si test seti; daha sonra %50'si eğitim, %50'si test seti olarak ayrılmıştır. Yapılan eğitimler sonucunda her iki durumda her bir algoritmanın en başarılı eğitim sonuçları bulunmuştur. Daha sonra bu sonuçlarla CPEP'de yapılan en başarılı eğitim sonucu karşılaştırılmıştır.

3.1. Termoelektrik Başlık Verileri İle Yapay Sinir Ağı Analizi

Termoelektrik başlık üzerinde yapılan deneyler sonucunda elde edilen veri setinden seçilen I, Th ve E bilgileri giriş sütunu, Tc bilgisi ise çıkış sütunu olarak belirlenmiştir. 3 giriş değerinin girildiği bir giriş katmanı, 14 nörondan oluşan bir gizli katman ve eğitim sonucunda hesaplanan Tc bilgisini ifade eden 1 adet çıkış katmanından oluşmaktadır. Çıkış sütunu 0-1 arasında, E bilgileri ± 1 arasında, I ve Th bilgileri ise 1 veya -1 olacak şekilde kategorize edilmiştir. Giriş ve çıkış katmanında aktivasyon fonksiyonu olarak sigmoid kullanılmıştır. İleri beslemeli, geri yayımlı ÇKA modeli için 7 adet algoritma seçilmiştir.

Eğitim ve test seti çıktıları beklenen çıkıştan çıkarılarak AE bulunur. AE değeri $\pm 0,5$ aralığında olan satır sayısı [1-4], maksimum AE değeri, K ve R katsayıları ağıın başarısını belirlemektedir.

3.1.1. %80'i eğitim, %20'si test seti olan veri setinin YSA ile analizi

Algoritmaların en başarılı eğitim sonucunu sağlayan değişkenleri ve değerleri Çizelge 3,1'de verilmiştir. Yapılan her eğitimde ağırlık [-1,1] aralığında rastgele verilmiştir. Çizelge 3,1'de verilen QK hızlı yayılım algoritma katsayısını, ÖK öğrenme katsayısını, M ise momentumu ifade etmektedir.

Çizelge 3.1. Eğitimde kullanılan algoritmaların değişken değerleri

	QP	CGD	QN	LMQN	LM	OBP	BBP	CPEP
Ağ yapısı	3/14/1	3/14/1	3/14/1	3/14/1	3/14/1	3/14/1	3/14/1	3/14/1
QK	1,75	-	-	-	-	-	-	-
ÖK	0,5	-	-	-	-	0,5	0,5	1
M	-	-	-	-	-	0,8	0,8	0,8
İterasyon sayısı	300 000	64 360	39 000	149 100	239	300 000	300 000	300 000

ÇKA modelinde Çizelge 3.1'e göre yapılan eğitim sonucunda Çizelge 3.2'de görülen en büyük ve en küçük AE değerleri elde edilmiştir. Çizelgede Beklenen (BTc), veri setindeki Tc çıkış sütun değerini; çıkış (ÇTc) ise eğitim sonunda ağın hesapladığı Tc değerini; K Korelasyon katsayısını; R Regresyonu ifade etmektedir. EGT, Eğitim setinden bir giriş olduğu; TST ise test setinden bir giriş olduğu anlamına gelmektedir.

Çizelge 3.2. Algoritmalarından elde edilen performans değerleri

Kullanılan algoritma	QP			CGD			QN			LMQN		
	ÇTc	BTc	AE	ÇTc	BTc	AE	BTc	ÇTc	AE	BTc	ÇTc	AE
EGT	-24,1	-24,1	0	-24,9	-24,9	0	-18,1	-18,1	8,8E-07	-20	-20	0
TST	-7,4	4,9	12,3	-22,6	-35,79	13,19	8,4	-37	45,4	-3,30	-36,9	33,6
K	0,989666			0,993156			0,976537			0,954950		
R	0,978374			0,986321			0,951217			0,910812		

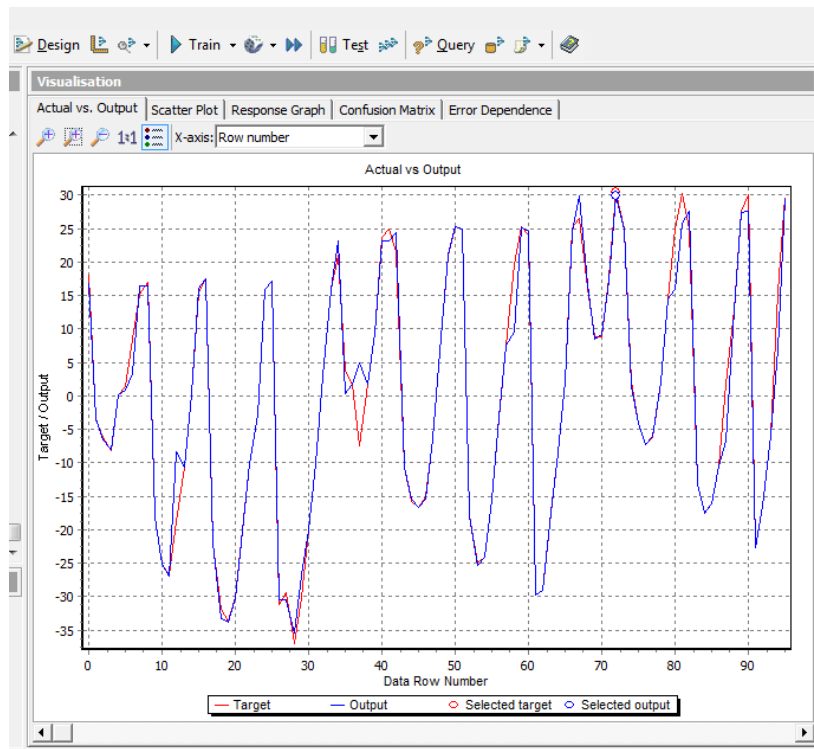
Kul. algoritma	LM			OBP			BBP			CPEP		
	ÇTc	BTc	AE	ÇTc	BTc	AE	BTc	ÇTc	AE	BTc	ÇTc	AE
EGT	-24,9	-24,8	0,01	-7,2	-7,2	0	-24,9	-24,9	0	8,9	8,8	0,1
TST	-22,6	-13,2	-9,4	-10,1	-33,1	23	15,9	21,7	5,8	-2,8	10,6	-13,4
K	0,995762			0,991892			0,996257			-		
R	0,991323			0,983657			0,9925			-		

QP algoritması ile yapılan eğitim sonunda AE değerlerine göre eğitim ve test seti sonuçlarının sayısı Çizelge 3.3'de görülmektedir. Veri setinde 73 sonucun AE

değerinin 0,5'ten küçük olduğu görülmüştür. Bu sonuçların 4 tanesinin test setine ait olduğu görülmüştür (Şekil 3.1).

Çizelge 3.3. QP eğitiminin performans sonuçları

AE değeri	Eğitim seti sonuç sayısı	Test seti sonuç sayısı	Veri seti sonuç sayısı
$AE \leq 0,5$	69	4	73
$0,5 < AE < 2$	6	2	8
$2 \leq AE < 3$	3	1	4
$3 \leq AE < 4$	-	3	3
$4 \leq AE < 5$	-	1	1
$AE \geq 5$	-	7	7
	Toplam : 78	Toplam : 18	Toplam : 96

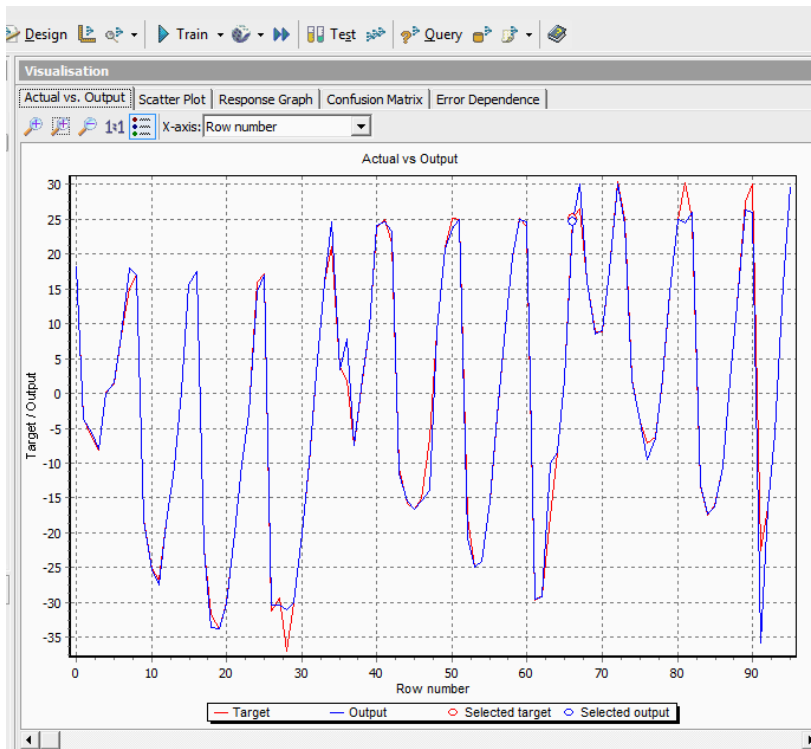


Şekil 3.1. QP'nin ANI ile eğitiminden elde edilen beklenen ve çıkış eğrileri

CGD algoritması ile yapılan eğitim sonunda AE değerlerine göre eğitim ve test seti sonuçlarının sayısı Çizelge 3.4'de görülmektedir. Veri setinde toplam 72 sonucun AE değerinin 0,5'ten küçük olduğu görülmüştür. Bu sonuçların 1 tanesi test setine ait olduğu görülmüştür (Şekil 3.2).

Çizelge 3.4. CGD eğitim performans çizelgesi

AE değeri	Eğitim seti sonuç sayısı	Test seti sonuç sayısı	Veri seti sonuç sayısı
$AE \leq 0,5$	71	1	72
$0,5 < AE < 2$	6	6	12
$2 \leq AE < 3$	-	3	3
$3 \leq AE < 4$	-	2	2
$4 \leq AE < 5$	-	1	1
$AE \geq 5$	-	6	6
	Toplam: 77	Toplam: 19	Toplam: 96



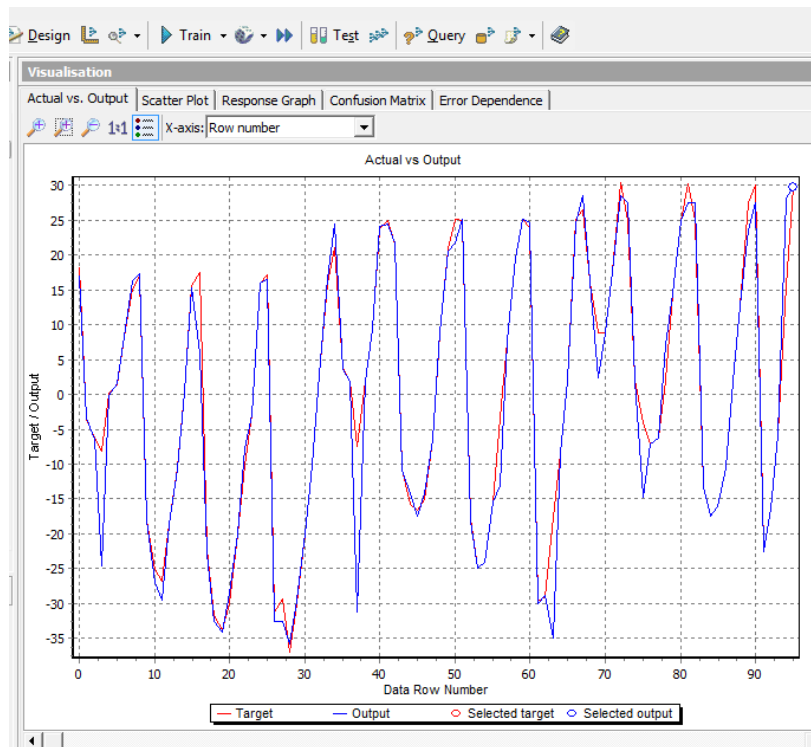
Şekil 3.2. CGD'nin ANI ile eğitiminden elde edilen beklenen ve çıkış eğrileri

QN algoritması ile yapılan eğitim sonunda AE değerlerine göre eğitim ve test seti sonuçlarının sayısı Çizelge 3.5'de görülmektedir. Veri setinde toplam 73 sonucun

AE değerinin 0,5'ten küçük olduğu görülmüştür. Bu sonuçların 1 tanesinin test setine ait olduğu görülmüştür (Şekil 3.3).

Çizelge 3.5. QN eğitim performans çizelgesi

AE değeri	Eğitim seti sonuç sayısı	Test seti sonuç sayısı	Veri seti sonuç sayısı
AE <= 0,5	72	1	73
0,5 < AE < 2	3	2	5
2 <= AE < 3	3	1	4
3 <= AE < 4	-	2	2
4 <= AE < 5	-	1	1
AE >= 5	-	11	11
	Toplam: 78	Toplam: 18	Toplam: 96



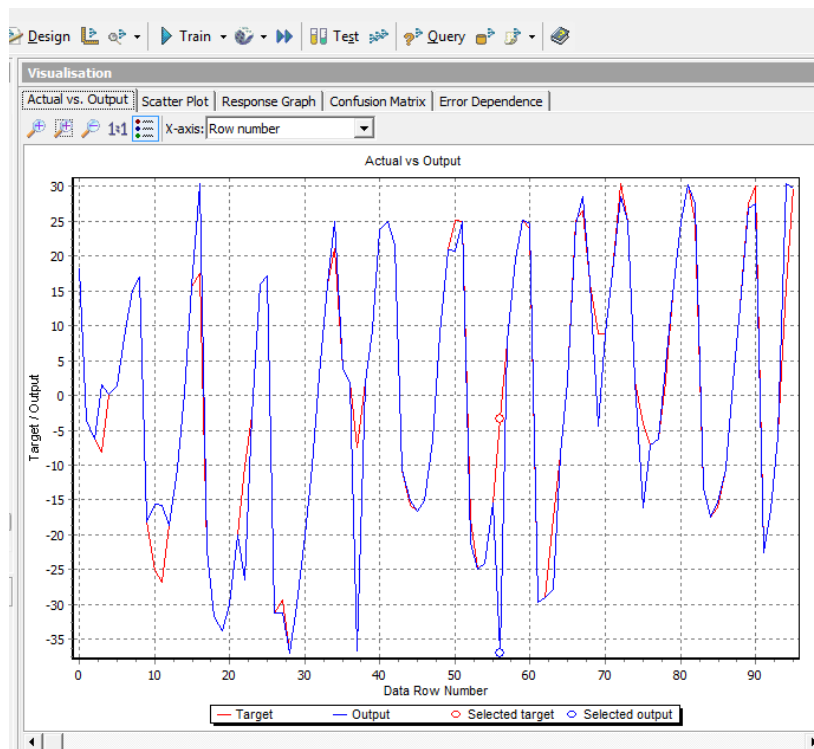
Şekil 3.3. QN'nin ANI ile eğitiminden elde edilen beklenen ve çıkış eğrileri

LMQN algoritması ile yapılan eğitim sonunda AE değerlerine göre eğitim ve test seti sonuçlarının sayısı Çizelge 3.6'da görülmektedir. Veri setinde toplam 74 sonucun

AE değerinin 0,5'ten küçük olduğu görülmüştür. Bu sonuçların 1 tanesinin test setine ait olduğu görülmüştür (Şekil 3.4).

Çizelge 3.6. LMQN eğitim performans çizelgesi

AE değeri	Eğitim seti sonuç sayısı	Test seti sonuç sayısı	Veri seti sonuç sayısı
$AE \leq 0,5$	73	1	74
$0,5 < AE < 2$	2	3	5
$2 \leq AE < 3$	2	1	3
$3 \leq AE < 4$	-	2	2
$4 \leq AE < 5$	-	1	1
$AE \geq 5$	-	11	11
	Toplam: 77	Toplam: 19	Toplam: 96



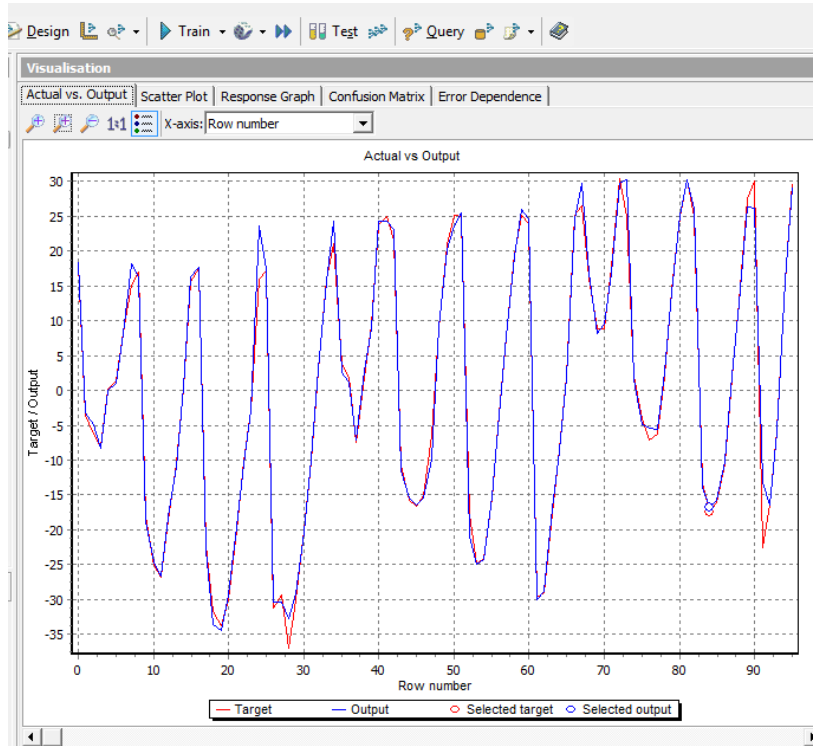
Şekil 3.4. LMQN'nin ANI ile eğitiminden elde edilen beklenen ve çıkış eğrileri

LM algoritması ile yapılan eğitim sonunda AE değerlerine göre eğitim ve test seti sonuçlarının sayısı Çizelge 3.7'de görülmektedir. Veri setinde toplam 52 sonucun

AE değerinin 0,5'ten küçük olduğu görülmüştür. Bu sonuçların 2 tanesinin test setine ait olduğu görülmüştür.

Çizelge 3.7. LM eğitim performans çizelgesi

AE değeri	Eğitim seti sonuç sayısı	Test seti sonuç sayısı	Veri seti sonuç sayısı
$AE \leq 0,5$	50	2	52
$0,5 < AE < 2$	27	8	35
$2 \leq AE < 3$	-	-	-
$3 \leq AE < 4$	-	4	4
$4 \leq AE < 5$	-	2	2
$AE \geq 5$	-	3	3
	Toplam: 77	Toplam: 19	Toplam: 96

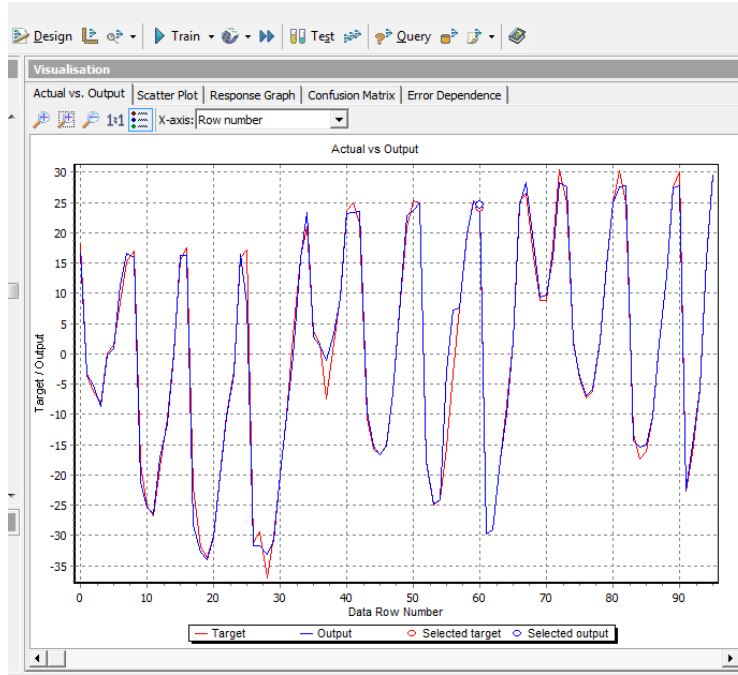


Şekil 3.5. LM'nin ANI ile eğitiminden elde edilen beklenen ve çıkış eğrileri

OBP algoritması ile yapılan eğitim sonunda AE değerlerine göre eğitim ve test seti sonuçlarının sayısı Çizelge 3.8'de görülmektedir. Veri setinde toplam 66 sonucun AE değerinin 0,5'ten küçük olduğu görülmüştür. Bu sonuçların 3 tanesinin test setine ait olduğu görülmüştür (Şekil 3.6).

Çizelge 3.8. OBP eğitim performans çizelgesi

AE değeri	Eğitim seti sonuç sayısı	Test seti sonuç sayısı	Veri seti sonuç sayısı
$AE \leq 0,5$	63	3	66
$0,5 < AE < 2$	10	-	10
$2 \leq AE < 3$	4	3	7
$3 \leq AE < 4$	-	4	4
$4 \leq AE < 5$	-	-	-
$AE \geq 5$	-	9	9
	Toplam: 77	Toplam: 19	Toplam: 96

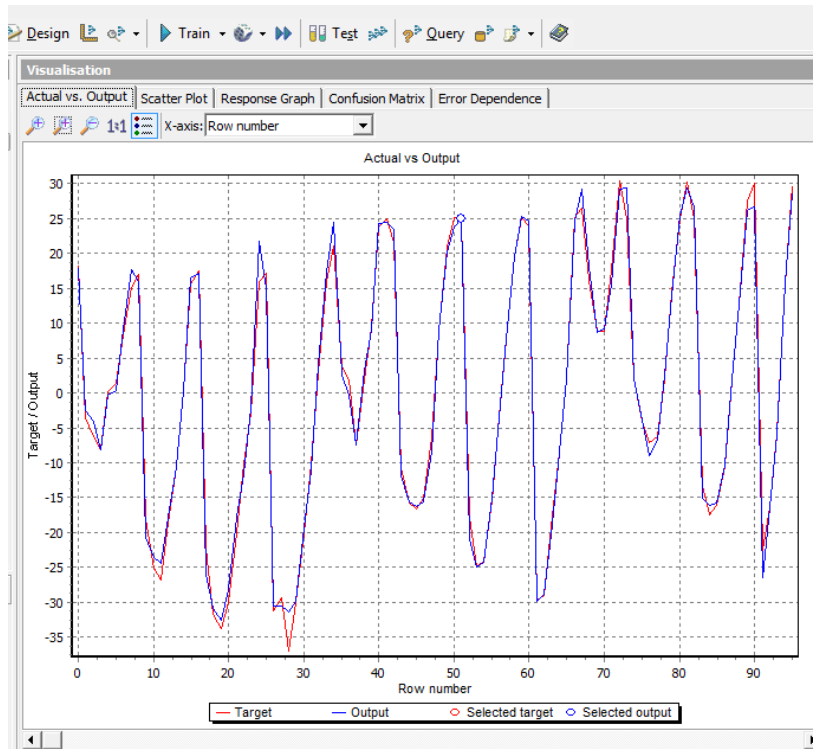


Şekil 3.6. OBP'nin ANI ile eğitiminden elde edilen beklenen ve çıkış eğrileri

BBP algoritması ile yapılan eğitim sonunda AE değerlerine göre eğitim ve test seti sonuçlarının sayısı Çizelge 3.9'da görülmektedir. Veri setinde toplam 43 sonucun AE değerinin 0,5'ten küçük olduğu görülmüştür. Bu sonuçların 1 tanesinin test setine ait olduğu görülmüştür (Şekil 3.7).

Çizelge 3.9. BBP eğitim performans çizelgesi

AE değeri	Eğitim seti sonuç sayısı	Test seti sonuç sayısı	Veri seti sonuç sayısı
$AE \leq 0,5$	42	1	43
$0,5 < AE < 2$	29	5	34
$2 \leq AE < 3$	4	8	12
$3 \leq AE < 4$	1	3	4
$4 \leq AE < 5$	-	1	1
$AE \geq 5$	-	2	2
Toplam: 76	Toplam: 20	Toplam: 96	

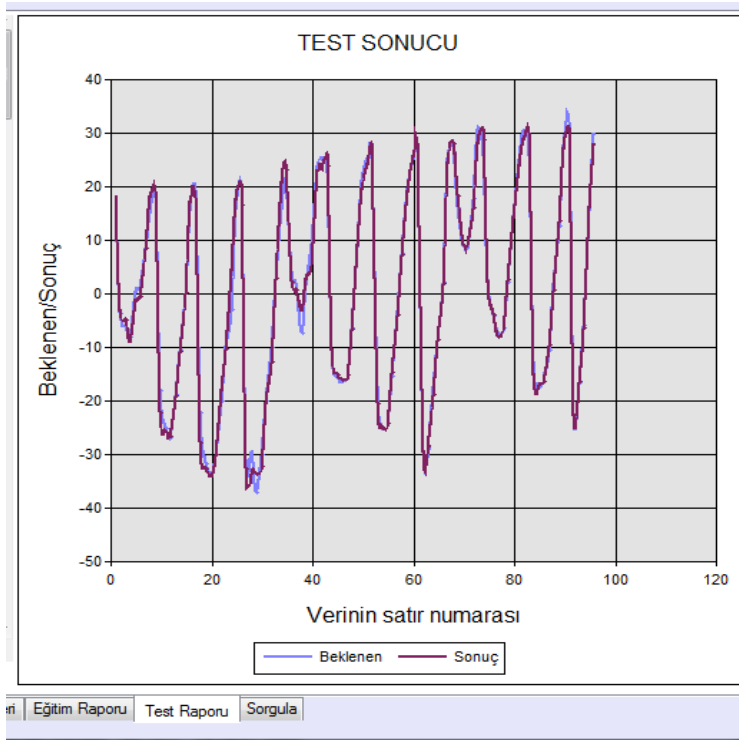


Şekil 3.7. BBP'nin ANI ile eğitiminden elde edilen beklenen ve çıkış eğrileri

CPEP'de yapılan eğitim sonunda AE değerlerine göre eğitim ve test seti sonuçlarının sayısı Çizelge 3.10'da görülmektedir. Veri setinde toplam 53 sonucun AE değerinin 0,5'ten küçük olduğu görülmüştür. Bu sonuçların 5 tanesinin test setine ait olduğu görülmüştür (Şekil 3.8).

Çizelge 3.10. CPEP'in performans çizelgesi

AE değeri	Eğitim seti sonuç sayısı	Test seti sonuç sayısı	Veri seti sonuç sayısı
$AE \leq 0,5$	48	5	53
$0,5 < AE < 2$	17	3	20
$2 \leq AE < 3$	14	6	20
$3 \leq AE < 4$	-	-	-
$4 \leq AE < 5$	-	3	3
$AE \geq 5$	-	3	3
	Toplam: 76	Toplam: 20	Toplam: 96



Şekil 3.8. CPEP sonuç grafiği

3.1.2. %50'si eğitim, %50'si test girişi olan veri setinin YSA ile analizi

Yapılan denemeler sonucunda en başarılı sonuç veren değişken değerleri Çizelge 3.11'de verilmiştir.

Çizelge 3.11. Eğitimde kullanılan değişken değerleri

	QP	CGD	QN	LMQN	LM	OBP	BBP	CPEP
Ağ yapısı	3/14/1	3/14/1	3/11/1	3/10/1	3/10/1	3/14/1	3/14/1	3/14/1
QK	1,75	-	-	-	-	-	-	-
ÖK	1	-	-	-	-	0,5	0,8	1
M	-	-	-	-	-	0,8	1	0,8
W	±0,3	±0,3	±0,3	±0,3	±0,3	±0,3	±0,3	±1
AF	Sigmoid	Sigmoid	Hiperbolik tanjant	Hiperbolik tanjant	Sigmoid	Sigmoid	Sigmoid	Sigmoid
İterasyon sayısı	300 000	900	1500	3500	94	300 000	300 000	300 000

ÇKA modelinde Çizelge 3.11'e göre yapılan eğitim sonucunda Çizelge 3.12'de görülen en büyük ve en küçük AE değerleri elde edilmiştir.

Çizelge 3.12. Algoritmalarından elde edilen performans değerleri

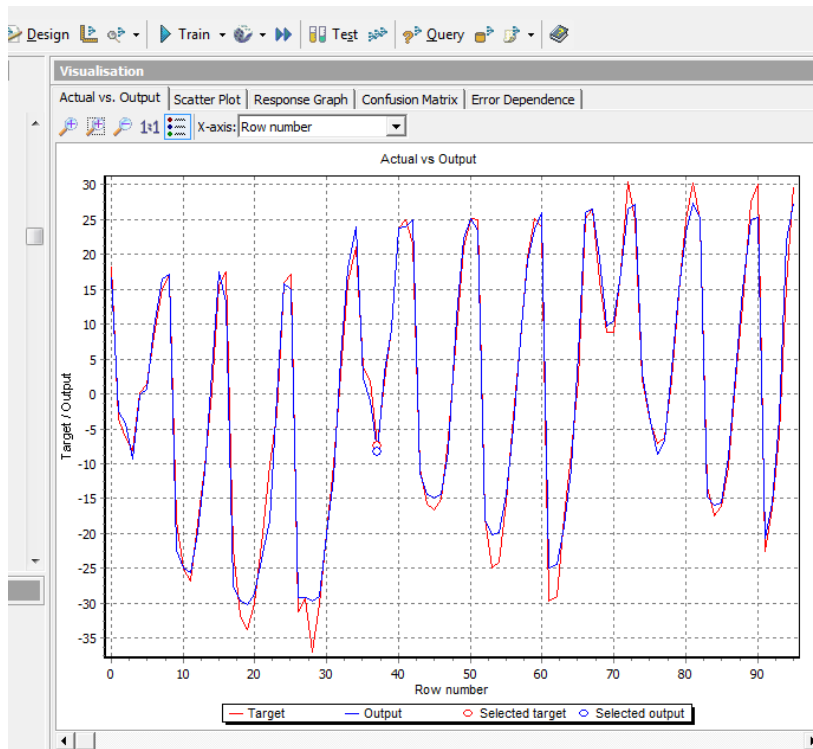
Kullanılan algoritma	QP			CGD			QN			LMQN			
	ÇTc	BTc	AE	ÇTc	BTc	AE	BTc	ÇTc	AE	BTc	ÇTc	AE	
EGT	24,9	24,9	0	20,9	20,9	0	-3,3	-3,29	0,01	30,2	30,19	0,01	
TST	1,8	29,1	27,3	-	29,7	-20,4	9,3	-2,9	13,3	16,2	0,3	19,8	19,5
K	0,971639			0,978012			0,989947			0,976972			
R	0,96213			0,955728			0,979695			0,954057			

Kul. algoritma	LM			OBP			BBP			CPEP			
	ÇTc	BTc	AE	ÇTc	BTc	AE	BTc	ÇTc	AE	BTc	ÇTc	AE	
EGT	-	24,1	-24,1	0	9,4	9,4	0	24,9	24,9	0	-29,09	-29,10	0,01
TST	-2,9	8,7	11,6	2,9	17,37	14,4 7	15,6	27,98	12,38	-2,89	6,96	-9,86	
K	0,992784			0,971998			0,970367			-			
R	0,98546			0,944299			0,959661			-			

QP algoritması ile yapılan eğitim sonunda AE değerlerine göre eğitim ve test seti sonuçlarının sayısı Çizelge 3.13'de görülmektedir. Veri setinde toplam 54 sonucun AE değerinin 0,5'ten küçük olduğu görülmüştür. Bu sonuçların 7 tanesinin test setine ait olduğu görülmüştür (Şekil 3.9).

Çizelge 3.13. QP eğitim performans çizelgesi

AE değeri	Eğitim seti sonuç sayısı	Test seti sonuç sayısı	Veri seti sonuç sayısı
$AE \leq 0,5$	47	7	54
$0,5 < AE < 2$	3	10	13
$2 \leq AE < 3$	-	6	6
$3 \leq AE < 4$	-	7	7
$4 \leq AE < 5$	-	3	3
$AE \geq 5$	-	13	13
	Toplam : 50	Toplam : 46	Toplam : 96

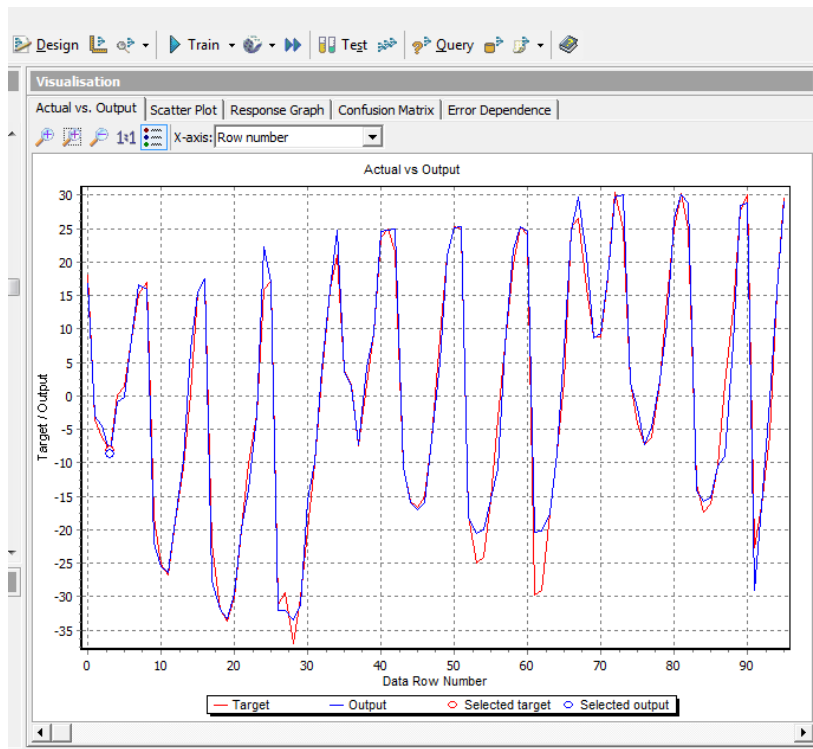


Şekil 3.9. QP'nin ANI ile eğitiminden elde edilen beklenen ve çıkış eğrileri

CGD algoritması ile yapılan eğitim sonunda AE değerlerine göre eğitim ve test seti sonuçlarının sayısı Çizelge 3.14'de görülmektedir. Veri setinde toplam 47 sonucun AE değerinin 0,5'ten küçük olduğu görülmüştür. Bu sonuçların 8 tanesinin test setine ait olduğu görülmüştür (Şekil 3.10).

Çizelge 3.14. CGD eğitim performans çizelgesi

AE değeri	Eğitim seti sonuç sayısı	Test seti sonuç sayısı	Veri seti sonuç sayısı
$AE \leq 0,5$	39	8	47
$0,5 < AE < 2$	9	11	20
$2 \leq AE < 3$	1	3	4
$3 \leq AE < 4$	1	8	9
$4 \leq AE < 5$	-	5	5
$AE \geq 5$	-	11	11
	Toplam: 50	Toplam: 46	Toplam: 96

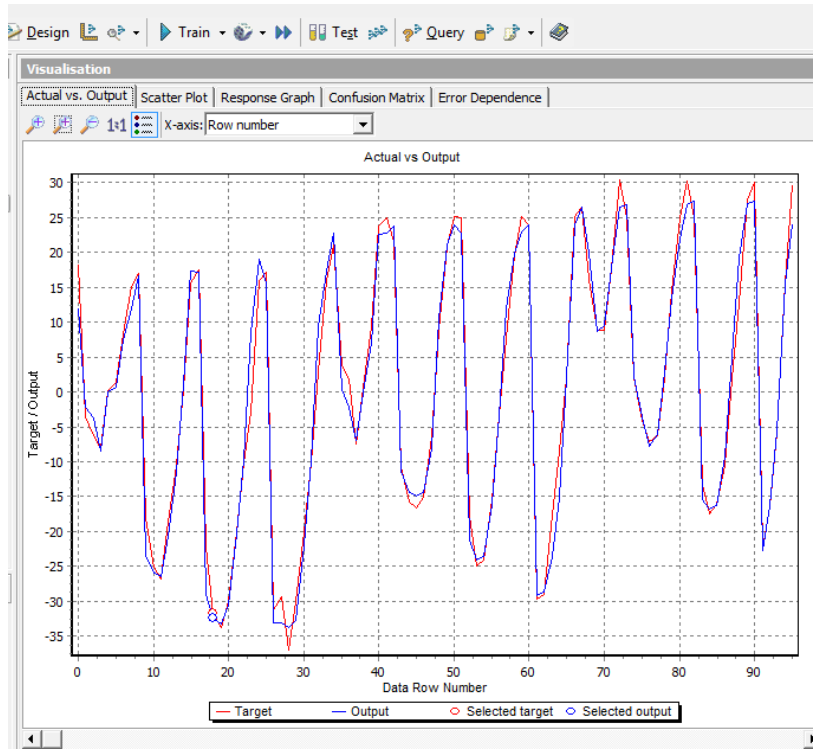


Şekil 3.10. CGD'nin ANI ile eğitiminden elde edilen beklenen ve çıkış eğrileri

QN algoritması ile yapılan eğitim sonunda AE değerlerine göre eğitim ve test seti sonuçlarının sayısı Çizelge 3.15'de görülmektedir. Veri setinde toplam 45 sonucun AE değerinin 0,5'ten küçük olduğu görülmüştür. Bu sonuçların 7 tanesinin test setine ait olduğu görülmüştür (Şekil 3.11).

Çizelge 3.15. QN eğitim performans çizelgesi

AE değeri	Eğitim seti sonuç sayısı	Test seti sonuç sayısı	Veri seti sonuç sayısı
$AE \leq 0,5$	38	7	45
$0,5 < AE < 2$	6	19	25
$2 \leq AE < 3$	3	8	11
$3 \leq AE < 4$	1	8	9
$4 \leq AE < 5$	-	3	3
$AE \geq 5$	-	3	3
	Toplam: 48	Toplam: 48	Toplam: 96

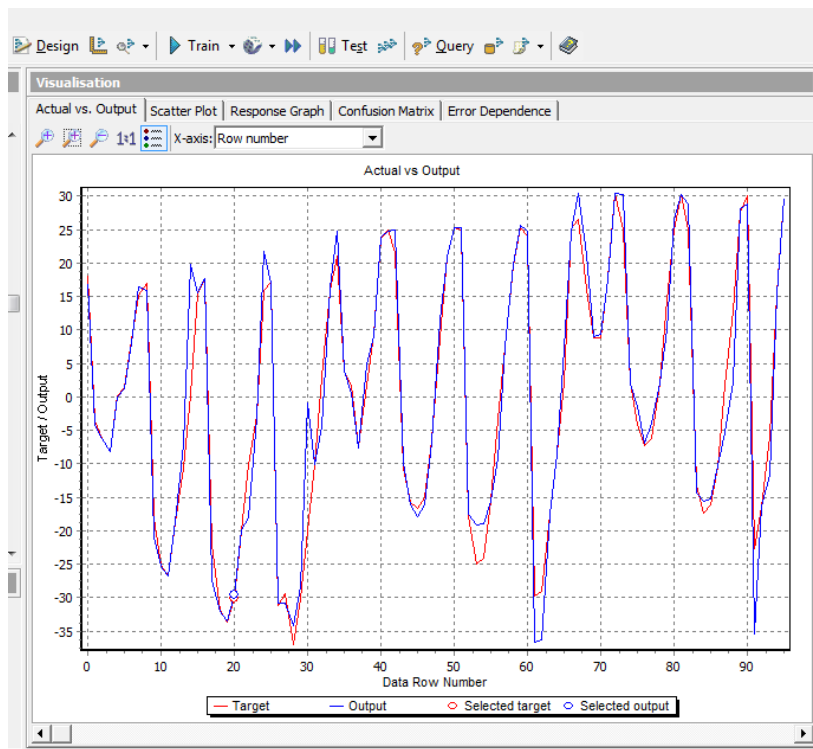


Şekil 3.11. QN'nin ANI ile eğitiminden elde edilen beklenen ve çıkış eğrileri

LMQN algoritması ile yapılan eğitim sonunda AE değerlerine göre eğitim ve test seti sonuçlarının sayısı Çizelge 3.16'de görülmektedir. Veri setinde toplam 50 sonucun AE değerinin 0,5'ten küçük olduğu görülmüştür. Bu sonuçların 41 tanesinin eğitim setine ait olduğu görülmüştür (Şekil 3.12).

Çizelge 3.16. LMQN eğitim performans çizelgesi

AE değeri	Eğitim seti sonuç sayısı	Test seti sonuç sayısı	Veri seti sonuç sayısı
$AE \leq 0,5$	41	9	50
$0,5 < AE < 2$	8	8	16
$2 \leq AE < 3$	1	5	6
$3 \leq AE < 4$	-	5	5
$4 \leq AE < 5$	-	1	1
$AE \geq 5$	-	18	18
	Toplam: 50	Toplam: 46	Toplam: 96

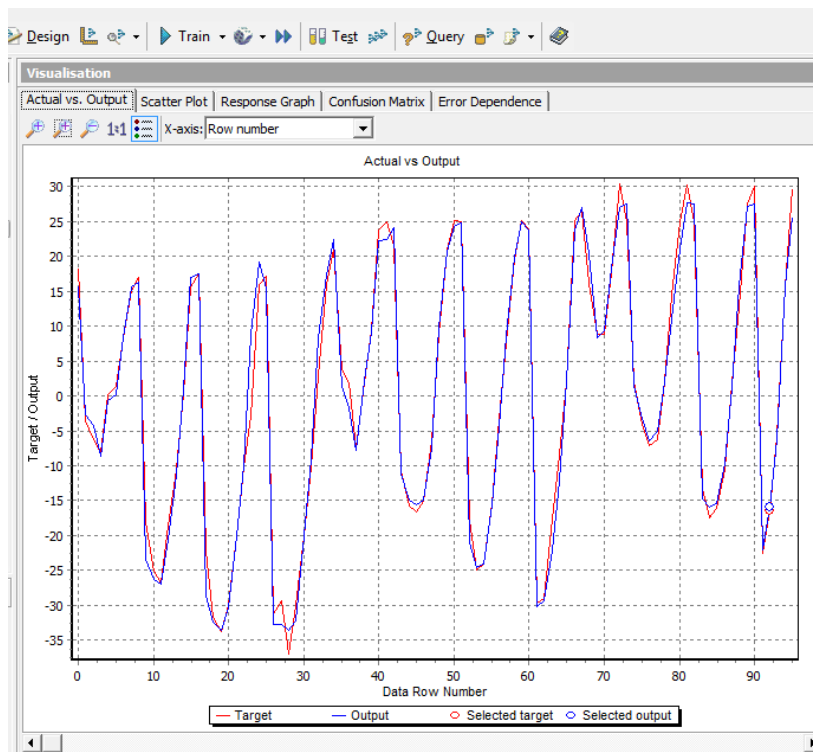


Şekil 3.12. LMQN'nin ANI ile eğitiminden elde edilen beklenen ve çıkış eğrileri

LM algoritması ile yapılan eğitim sonunda AE değerlerine göre eğitim ve test seti sonuçlarının sayısı Çizelge 3.17'de görülmektedir. Veri setinde toplam 37 sonucun AE değerinin 0,5'ten küçük olduğu görülmüştür. Bu sonuçların 13 tanesinin test setine ait olduğu görülmüştür (Şekil 3.13).

Çizelge 3.17. LM eğitim performans çizelgesi

AE değeri	Eğitim seti sonuç sayısı	Test seti sonuç sayısı	Veri seti sonuç sayısı
$AE \leq 0,5$	24	13	37
$0,5 < AE < 2$	20	12	32
$2 \leq AE < 3$	2	9	11
$3 \leq AE < 4$	-	2	9
$4 \leq AE < 5$	-	3	3
$AE \geq 5$	-	4	4
	Toplam: 51	Toplam: 45	Toplam: 96

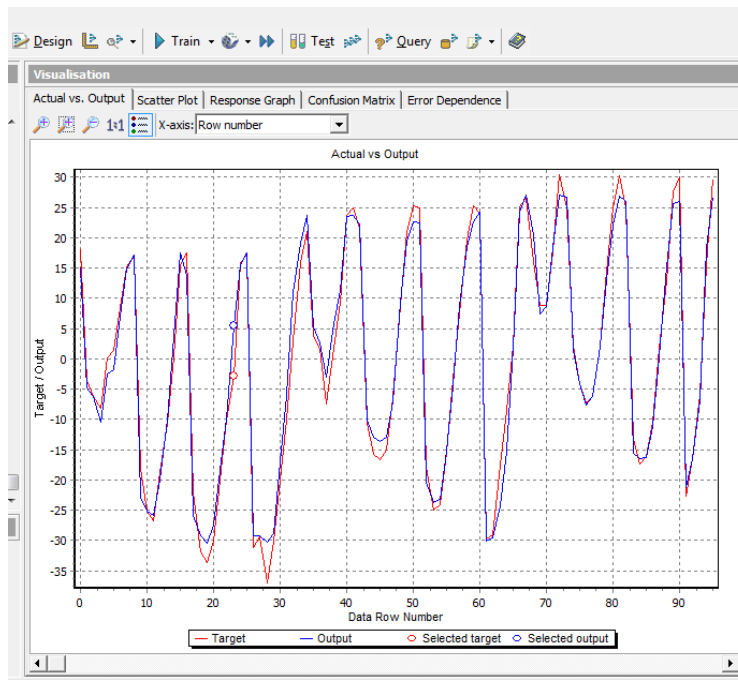


Şekil 3.13. LM'nin ANI ile eğitiminden elde edilen beklenen ve çıkış eğrileri

OBP algoritması ile yapılan eğitim sonunda AE değerlerine göre eğitim ve test seti sonuçlarının sayısı Çizelge 3.18'de görülmektedir. Veri setinde toplam 49 sonucun AE değerinin 0,5'ten küçük olduğu görülmüştür. Bu sonuçların 6 tanesinin test setine ait olduğu görülmüştür (Şekil 3.14).

Çizelge 3.18. OBP eğitim performans çizelgesi

AE değeri	Eğitim seti sonuç sayısı	Test seti sonuç sayısı	Veri seti sonuç sayısı
$AE \leq 0,5$	43	6	49
$0,5 < AE < 2$	6	11	17
$2 \leq AE < 3$	-	11	11
$3 \leq AE < 4$	-	2	2
$4 \leq AE < 5$	-	5	5
$AE \geq 5$	-	12	12
	Toplam: 49	Toplam: 47	Toplam: 96

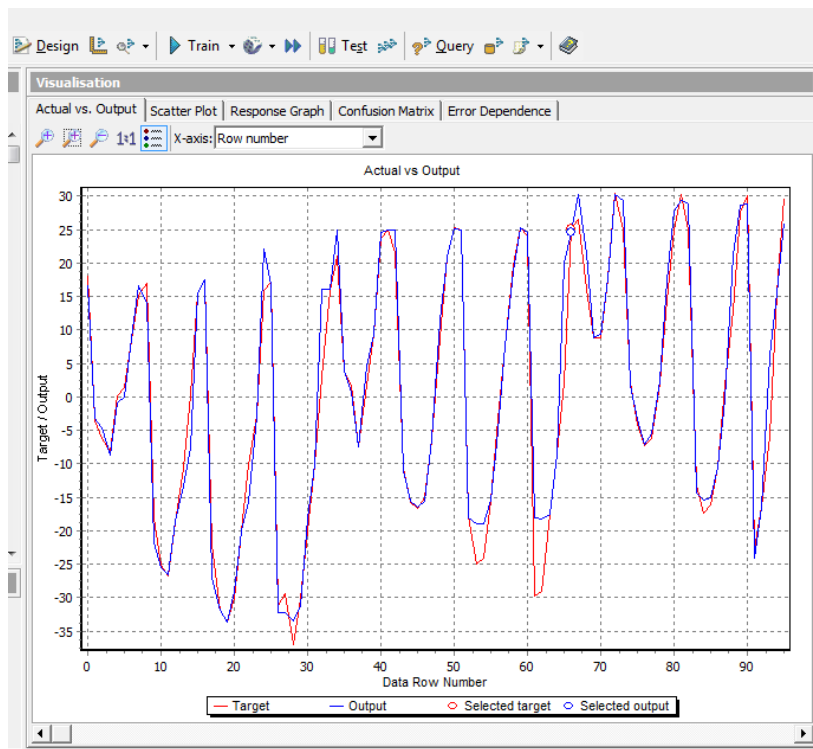


Şekil 3.14. OBP'nin ANI ile eğitiminden elde edilen beklenen ve çıkış eğrileri

BBP algoritması ile yapılan eğitim sonunda AE değerlerine göre eğitim ve test seti sonuçlarının sayısı Çizelge 3.19'da görülmektedir. Veri setinde toplam 54 sonucun AE değerinin 0,5'ten küçük olduğu görülmüştür. Bu sonuçların 11 tanesinin test setine ait olduğu görülmüştür (Şekil 3.15).

Çizelge 3.19. BBP eğitim performans çizelgesi

AE değeri	Eğitim seti sonuç sayısı	Test seti sonuç sayısı	Veri seti sonuç sayısı
$AE \leq 0,5$	43	11	54
$0,5 < AE < 2$	6	13	19
$2 \leq AE < 3$	-	3	3
$3 \leq AE < 4$	-	6	6
$4 \leq AE < 5$	-	4	4
$AE \geq 5$	-	10	10
	Toplam: 49	Toplam: 47	Toplam: 96

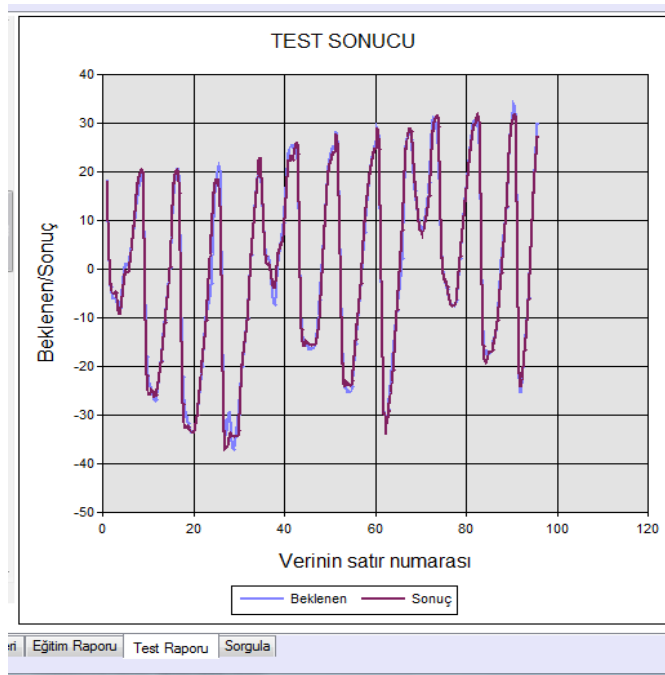


Şekil 3.15. BBP'nin ANI ile eğitiminden elde edilen beklenen ve çıkış eğrileri

CPEP'de yapılan eğitim sonunda AE değerlerine göre eğitim ve test seti sonuçlarının sayısı Çizelge 3.20'de görülmektedir. Veri setinde toplam 43 sonucun AE değerinin 0,5'ten küçük olduğu görülmüştür. Bu sonuçların 13 tanesinin test setine ait olduğu görülmüştür (Şekil 3.16).

Çizelge 3.20. CPEP performans çizelgesi

AE değeri	Eğitim seti sonuç sayısı	Test seti sonuç sayısı	Veri seti sonuç sayısı
$AE \leq 0,5$	30	13	43
$0,5 < AE < 2$	13	14	27
$2 \leq AE < 3$	5	8	13
$3 \leq AE < 4$	-	6	6
$4 \leq AE < 5$	-	3	3
$AE \geq 5$	-	4	4
	Toplam: 48	Toplam: 48	Toplam: 96



Şekil 3.16. CPEP'in eğitiminden elde edilen beklenen ve çıkış eğrileri

3.1.3. CPEP'de tüm deney sonuçlarının test edilmesi

96 deney sonucunun 48'i eğitim, kalanı test girişi olarak ayarlanmıştır. Önce eğitim girişleri ağa gösterilmiş daha sonra test girişleri ağa sorularak ağın başarısı ölçülmüştür. Bu başarının tüm veri seti için geçerliliğini görmek için ağa ne eğitim ne de test seti olarak sorulmayan 260 adet giriş test seti olarak sorulmuş ve sonuçlar kaydedilmiştir (Çizelge 3.21).

Çizelge 3.21. Ağın genel performansı

AE değeri	Deney sonuç sayısı
$AE \leq 0,5$	82
$0,5 < AE < 2$	111
$2 \leq AE < 3$	28
$3 \leq AE < 4$	18
$4 \leq AE < 5$	11
$AE \geq 5$	10
Toplam	260

458 adet deney sonucundan oluşan veri setinin 96 sonuçtan oluşan örneklem bir veri seti alınmıştır. Veri setinin eğitiminden sonra kalan 260 sonuç ağa test seti olarak sorulmuştur. Bunun sonucunda ağın AE değerinin çoğunlukla 2'nin altında olduğu ve verdiği en büyük AE değerinin de 9,86 olduğu bulunmuştur. Ağın AE değeri 2'nin altında verdiği satır sayısına göre %74 bir başarı elde edilmiştir.

3.2. Tartışma

QP, CGD, QN, LMQN, LM, OBP, BBP algoritmaları kullanılarak eğitimler yapılmıştır. Her eğitimde W değeri istenen değer aralığında rastgele atanmaktadır. Bu nedenle ağ yapısı, M, ÖK, W, iterasyon sayısı ve algoritma aynı olan eğitimlerin birbirinden farklı AE sonuçları verebildiği görülmüştür. ANI ve CPEP'de yapılan eğitimlerde en başarılı sonuç veren ağlar seçilmiştir.

Her bir algoritmada eğitim girişleri belirtilen ağ yapısıyla eğitime sokulduktan sonra test girişleri ağa sunulmuştur. Bu yolla elde edilen eğitim ve test sonuçları eğitim ve test çıktılarıyla karşılaştırılarak her sonuç için bir AE (mutlak hata) değeri bulunmuştur. Eğitimin başarı ölçütleri olarak $AE = \pm 0,5$ arasında olan sonuç sayısı, K ve R katsayıları ve en büyük AE değeri kullanılmıştır.

Bu tezde ilk olarak veri setinin %80'i eğitim girişi, %20'si test girişi olan eğitimler yapılmıştır. Yapılan eğitimlerde AE değeri 0,5'ten küçük olan sonuçların QP'de $4/73$ 'ü, CGD'de $1/72$ 'si, QN'de $1/73$ 'ü, LMQN'de $1/74$ 'ü, LM'de $2/52$ 'si, OBP'de $3/66$ 'sı, BBP'de $1/43$ 'ünün test sonucu olduğu görülmüştür. QP, CGD, QN, LMQN ve OBP'de test sonuçlarının çoğunun AE değeri, 5 ve 5'ten büyüktür. Bu sonuç bize QP, CGD, QN, LMQN ve OBP algoritmalarıyla yapılan eğitimlerin ezberlediğini göstermektedir. Ancak LM ve BBP'de test sonuçların AE değerleri 1-5 arasındadır. Bu nedenle LM ve BBP algoritmalarıyla yapılan eğitimler ezberlememiştir. K ve R katsayı değerlerine göre de LM ve BBP'de en başarılı eğitimler yapılmıştır. CPEP'de yapılan eğitimler sonucunda AE değeri 0,5'ten küçük olan sonuçların $5/53$ 'ünün test sonucu olduğu görülmüştür. Test sonuçlarının AE değerleri BBP ve LM'ye benzemektedir. Bu nedenle eğitim ezberlememiştir.

Eğitimin ezberlememesi için veri setinin %50'si eğitim girişi, %50'si test girişi olan eğitimler yapılmıştır. Yapılan eğitimlerde AE değeri 0,5'ten küçük sonuçların QP'de $7/54$ 'ü, CGD'de $8/47$ 'si, QN'de $7/45$ 'i, LMQN'de $9/50$ 'si, LM'de $13/37$ 'si, OBP'de $6/49$ 'u, BBP'de $11/54$ 'ünün test sonucu olduğu görülmüştür. Bu sonuç algoritmaların %80'i eğitim, %20'si test girişi olan eğitimlere göre daha başarılı olduğunu göstermektedir. AE = $\pm 0,5$ arasındaki sonuç sayı ile 5 ve 5'ten büyük sonuç sayısına göre en başarılı eğitim LM algoritmasından elde edilmiştir. K ve R katsayı değerlerine göre de LM algoritması en başarılı eğitim sonucunu vermiştir. CPEP'de AE = $\pm 0,5$ arasındaki sonuçların $13/43$ 'ünün test sonucu olduğu görülmüştür. Ayrıca AE değeri 5 ve 5'ten büyük sonuç sayısı $4/48$ 'dir. Bu sonuç bize eğitim başarısının LM ile benzer olduğunu göstermektedir.

4. SONUÇ VE ÖNERİLER

Termoelektrik modüllerden oluşan bir sistemde çıkış kontrolünde karşılaşılabilen sistemin çıkışını olumsuz etkileyen ortamın sıcaklığı, akım, gerilim gibi faktörler sonucu yanlış ölçümler elde edilebilmesi, ölçüm seti için yapılan deneylerin zaman alması, bu verilere göre sistem davranışını yorumlamanın zor olması ve ölçüm sonuçlarının kontrol edilmesi için defalarca ölçüm yapılması gibi olumsuzlukların önüne geçmek için YSA'nın ileriye dönük tahmin yeteneği kullanılmıştır. Termoelektrik modüllerden oluşan termoelektrik başlıktan elde edilen deney verileri kullanılarak eğitimler yapılmıştır. Eğitimde ANI ve CPEP kullanılmıştır. CPEP kullanılmasının nedeni hem eğitimin kaynak kodlarına erişmek hem de geliştirilerek ilerde gelişmiş bir eğitim simülasyon programı tasarlamaktır. ANI kullanılmasının nedeni ise hataları dinamik olarak grafiksel göstermesinin yanında eğitim sonucunu özet tablo şeklinde giriş çıkış ve hataları ortalamaları ile birlikte vermesidir [35-39].

Termoelektrik başlığın tasarım nedeni beyin hipotermiyasını sağlamaktır. Hipotermiya, vücut sıcaklığının tehlikeli boyutlarda düşük olmasıdır. Genel olarak insana zarar veren bir olgu olarak algılsa da, Hipotermiya bazı durumlarda hayat kurtarıcı olmaktadır. Beyin, kandaki oksijen düzeylerindeki düşüşe son derece duyarlı bir organdır. Kanın vücut dışındaki bir cihaza aktarılması sırasında, beyne giden oksijen miktarında değişiklikler olabilmektedir. Oksijen düzeyindeki değişikliklere bağlı olarak beyinde oluşabilecek hücre hasarını engellemek amacıyla vücut sıcaklığı düşürülür. Vücut sıcaklığındaki her 1 derecelik düşüş, beyin oksijen ihtiyacını % 10 civarında azaltır. Ameliyat sırasında oluşturulan Hipotermiya sayesinde beyin, daha az oksijene maruz kalsa dahi kendini koruyabilir [40-41]. Beyin hipotermiyasındaki temel problem beyin hızla (ilk 1 saat) gereken sıcaklığa kadar soğutulması ve bu sıcaklıkta kontrollü bir şekilde uzun süre tutulmasıdır. Kabul edilebilir sıcaklık hatası $\pm 0,5$ °C'dir. Termoelektrik başlığın sıcaklık kontrolü için 4 adet modülden oluşan özel bir ölçüm seti geliştirilerek deneyler yapılmış ve bir veri seti oluşturulmuştur [1].

Bu tezde, elde edilen veri setinin YSA ile bir benzetimi yapılmıştır. Bu benzetimle termoelektrik başlıkta kullanılan esnek termoelektrik modüllerin bir modeli oluşturulmuştur. Böylece veri seti oluşturulurken yapılan deneylerdeki, ortam sıcaklığı, okuyucu hatası, ölçüm sırasında yapılan anahtarlardan ve termoelektrik modüllere besleme için kullanılan güç kaynağında, özellikle yüksek akımlardan dolayı meydana gelen dalgalanmalardan kaynaklanan ölçüm hatalarının da önüne geçilmiştir. Termoelektrik sistem tasarımı yapılırken uzun zaman alan deneyler yerine kullanılabilir.

YSA ile veri setinin benzetimi için 564 adet deney sonucundan tekrarlanan deneyler çıkarılarak 458 adet deney sonucu kullanılmıştır. Bu deney sonucundan da tüm veri setini örnekleyecek 96 deney sonucu seçilmiştir [1]. Bu deney sonuçlarının eğitiminde ANI ve CPEP programları kullanılmıştır. Eğitim setinin veri sayısı arttırıldıkça YSA'nın ileriye yönelik tahmin yapamadığı gibi eğitim için kendine gösterilen deney sonuçlarını da ezberlediği görülmüştür. Bu nedenle eğitim ve test seti 48'er veriden oluşacak şekilde eğitimler yapılmış ve ağın bu şekilde ezberlemediği görülmüştür. Yapılan eğitimler sonucunda her iki programda en başarılı sonuç veren eğitimler karşılaştırılmıştır. Buna göre CPEP programının ANI programına yakın bir başarı gösterdiği görülmüştür.

CPEP programında en başarılı eğitim yapılan ağa daha önce sorulmamış 260 deney sonucu sorulmuştur. Bunun sonucunda ağın, sistemin çıkış sıcaklığını çoğunlukla 2°C'nin altında bir hata ile tahmin ettiği ve en çok bir deney sonucunda 9,86 °C'lık hata verdiği görülmüştür. 40A ile yapılan deneylerde ölçüm hatası olma ihtimali yüksektir [1] ve 9,86'lık hatayı 40A ile yapılmış deney sonucunda vermektedir. Eğitim sonunda beklenen performans, hatanın en çok 0,5 °C ve altında olmasıdır. Bu da ağın amaca yakın bir başarıda çalıştığını göstermektedir. Örneklem veri setindeki eğitim setinin veri sayısı arttırılarak, eğitime sokulan verilerin eğitim sırası değiştirilerek yapılan eğitimlerde daha başarılı sonuçlar elde edilebilir.

KAYNAKLAR

1. Güler, İ., Ahıska, R., Yavuz, H., “Neuro-Fuzzy Inference System For Control Of A Thermoelectric Brain Cooler”, *Journal of Thermoelectricity, Journal of Thermoelectricity*, 31-38 (2008).
2. Katz, L., Younga, M., A.S., Franka, J.E., Wanga, Y., Park, K., “Regulated Hypothermia reduces brain oxidative stres after hypoxic-ischemia”, *Brain Research*, 1017 85-91 (2004).
3. Gal, R., Cundrle, I., Zimova, I., Smrcka, M., “Mild hypothermia therapy for patients with severe brain injury”, *Clinical Neurology And Eurosurgey*, 104 318-321 (2002).
4. Clifton, G.L., “Hypothermia and severe brain injury”, *J. Neurosurgery*, 93(4): 718-719 (2000).
5. Hill, T., O’Connor, M., Remus, W., “Neural Networks Models for Time Series Forecasts”, *Management Sciences*, Cilt 42, No 7,1082-1092 (1996).
6. Sharda, R., Patil, R.B., “Connectionist Approach to Time Series Prediction: An Emprical Test”, *Journal of Intelligent Manufacturing*, Cilt 3, 317-323 (1992).
7. Tang, Z., Almeida, C., Fishwick, P.A., “Time Series Forecasting Using Neural Networks vs Box-Jenkins Methodology”, *Simulation*, Cilt 57, No 5, 303-310 (1991).
8. Zhang, G., Patuwo, B.E., Hu, M.Y., “Forecasting with Artificial Neural Networks: The State of the Art”, *Inter. Journal of Forecasting*, Cilt 14, 35-62 (1998).
9. Ahıska, R., Patent, EP, A61, F7\00 (1993).
10. Kapıdere, M., “Mikrodenetleyicili Termohipoterm Tıp Cihazı Tasarımı ve Gerçekleştirilmesi”, *Gazi Üniversitesi Fen Bilimleri Enstitüsü*, Doktora Tezi (2005).
11. Ahıska, R., Ciylan, B., Savas, Y., Güler, İ., “Standart Termoelektrik Modülün Z Parametresinin Ölçülmesi için Yeni Yöntem ve Sistem”, *Gazi Üniversitesi Mühendislik Fakültesi Dergisi*, 19(4):467-473 (2004).
12. Ahıska, R., Güler, İ., Savas, Y., “Termoelektrik soğutucunun özelliklerinin araştırılması”, *Politeknik Dergisi*, 2 (3):89-94 (1999).

13. Ahıska, R. “New Method for Investigation of Dynamic Parameters of Thermoelectric Modules” Doga-Tr., *Turkish Journal of Electrical Engineering & Computer Sciences*, Vol.1 No.4, 2007.
14. Benjamin, W ., “Artifisal Neural Networks Concepts and Theory”, *IEEE Computer Society Press*, Washington, 45 (1992).
15. Öztemel, E., “Yapay Sinir Ağları”, *Papatya Yayıncılık*, İstanbul, 13-113 (2003).
16. Mahmut, M., Güngör, M., “Askı Madde Konsantrasyonu ve Miktarının Yapay Sinir Ağları ile Belirlenmesi”, *İMO Teknik Dergi*, 219: 3267-3282 (2004).
17. McCullogh, WS., and Pitts, W., “A logical calculus of ideas imminent in nervous activity”, *Bull. Math. Biophysics*, 5: 115-33 (1943).
18. Saraç, T., “Yapay Sinir Ağları”, *Seminer Projesi*, Ankara, 22-75 (2004)
19. İnternet : Yapay Zeka “Yapay Sinir Ağları” , www.ahmetkakici.com/category/yapay-sinir-aglari.html, (2009).
20. İnternet : Yapay Zeka “Yapay Zeka Nedir?” www.yapay-zeka.org/modules/wiwimod/index.php?page=ANN (2009).
21. İnternet : Drakos “Computer Based Learning Unit” <http://www.csc.kth.se/~orre/snns-manual/UserManual/node147.html> (1995).
22. Champanard A. J., “AI game development: synthetic creatures with learning and reactive behaviors”, *New Riders*, United States of America, 244 (2004).
23. İnternet : Statistica “Electronic Statistics TextBook” <http://www.statsoft.com/textbook/statistics-glossary/q/button/q/#Quick%20Propagation> (2011)
24. Karakış, R., “Yapay Sinir Ağları ve Lojistik Regresyon Yöntemleri ile Meme Kanseri Koltuk Altı Lenf Modu Durumunun Belirlenmesi”, Yüksek Lisans Tezi, *Gazi Üniversitesi Bilişim Enstitüsü*, Ankara , 20-27 (2009)
25. Haykin, S., *Neural Networks: A Comprehensive Foundation*, New York, Macmillan College Publishing Company Inc, (1994).
26. Chaudhuri, B.B., Bhattacharya, U., “Efficient Training and Improved Performance of Multilayer Perceptron in Pattern Classification”, *Neurocomputing*, Vol 34, 11-27 (2000).
27. Rumelhart, D.E., Hinton, G.E., Williams, R.J., “Learning Representations by Back-Propagating Errors”, *Nature*, Vol 323, 533-536 (1986).

28. Hagan, M. T., Menhaj, M. B., “Training feedforward Networks with the Marquardt algorithm”, *IEEE Transactions on Neural Networks*, 5(6):989-993 (1994).
29. Bolat, S., Kalenderli, Ö., “Levenberg-Marquardt algoritması kullanılan yapay sinir ağı ile elektrot biçim optimizasyonu”, *International XII. Turkish Symposium on Artificial Intelligence and Neural Networks – TAINN (2003)*, Çanakkale, 2-4 (2003).
30. Le Cun, Y., “Efficient learning and second order methods”, *Tutorial presented at Neural Information Processing System 93, Denver*, 1-71 (1993).
31. Demuth, H., Beale, M., Hagan, M., “Neural Network Toolbox 6 User’s Guide, The MathWorks”, Inc., Natick, *Massachusetts*, 156-228 (1992).
32. Kollias, S., Anantassiou, D., “An adaptive least squares algorithm for the efficient training of artificial neural networks”, *IEEE Transactions on Circuits and Systems*, 36(8):1092-1101 (1989).
33. İnternet : FAQ Archives “Batch Back Propagation” <http://www.faqs.org/faqs/ai-faq/neural-nets/part2/section-16.html> (2011)
34. Özdamar, K., “Spss ile Biyoistatistik”, *Kaan Kitapevi*, İstanbul, 419-460 (2003).
35. Shanthi, D., Sahoo, G., Saravanan, N., “Designing an Artificial Neural Network Model for the Prediction of Thrombo-embolic Stroke”, *International Journals of Biometric and Bioinformatics (IJBB), Volume (3) : Issue (1)*, 10-17 (2009).
36. Raghu Prasad, B.K., Eskandari, H., Venkatarama Reddy B.V., “Prediction of compressive strength of SCC and HPC with high volume fly ash using ANN” , *Volume 23, Issue 1*, 117-128 (2009).
37. Iswaran, N., Percy, D.F., “Conflict analysis using Bayesian neural Networks and generalized linear models”, *Journal of the Operational Research Society*, 61: 332 –341 (2010).
38. Sidda Reddy, B., Suresh Kumar, J., Kumar Reddy V., Padmanabhan, G. “Application of Soft Computing for the Prediction of Warpage of Plastic Injection Molded Parts” , *JOURNAL OF Engineering Science and Technology Review*, 2 :56-62 (2009).
39. Pin-Chang, C., Chih-Yao, L., Hung-Teng C., “An Empirical Study of the Artificial Neural Network for Currency Exchange Rate Time Series Prediction”, *The Sixth ISSN 2009*, 56 : 543–549 (2009).
40. İnternet : Wikimedia Foundation “Hipotermi” <http://tr.wikipedia.org/wiki/Hipotermi> (2011)

41. İnternet : Anesteziik “Hipoterminin Yararları” <http://www.anesteziik.com/anesteziik.com/index.php?entry=entry110212-235019> (2011)

EKLER

EK-1. CPEP eğitim programı kodları

```

using System;
using System.Collections;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.OleDb;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading;
using System.Windows.Forms;
using MyExcel = Microsoft.Office.Interop.Excel;
using Owc11 = Microsoft.Office.Interop.Owc11;
using System.Diagnostics;

namespace dene
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        int progress;
        char[] egtstmi;
        bool testediliyor;
        public string giris,ara,cikis="";
        string bir,iki,uc,dort="";//ara katmandaki neuron sayıları
        static int satirsayisi, sutunsayisi;
        public int form1satirsayisi, form1sutunsayisi;
        static double[][] katman,katmantst;
        static double cikis_son_neuron, ogrnmorn_lrnrate, quick_const;
        static int satir, strtst;
        //Excelden veri okumak için
        MyExcel.Application uygulama;
        MyExcel.Workbook kitap;
        MyExcel.Worksheet sayfa;
        static double[,] tstgirisler,sorgula;
        static int testsayisi, egtsayisi;
        static double[,] egitimgirisler;
        static double[] tgrstst,cvr_tgrstst;
        static int katmansayisihesap;
        static double[][][] agirlik_tum;
        static double[][] delta;
        static double[][] deltaq;
    }
}

```

EK-1. (Devam) CPEP eğitim programı kodları

```

static double[][] eski_gradient;
static double hata_localgrediet, hata_toplam;
double[] bektmgrs, cvr_bektmgrs;
//grafik için
ArrayList MSEgrfk;
//BACK PROPOGATION VE QUICKPROPOGATION DEĞİŞKENLERİ
int dur_iter, bolum_sonuc, satir_hesap, dur;//bolumden sonra kalanında döngüde
çalışması için "dur"
int iterasyon_sayisi, kalan_sonuc;
int DENEME_SAY;
//FONKSİYONLARDA KULLANILAN ORTAK DEĞİŞKENLER
int i, neuron, j;
//delta
double topla_delta;
public void egitm_test_satir_ata()
{
strttst = 0;//testte kullanılan beklenen sutunu için
//Exceldeki eğitim ve test seti ataması yapıyorum çıkış sutunu da dahil
richTextBox1.Text = "";
//////////gerek yok gibi silmeyi unutma
uygulama = new MyExcel.Application();
kitap = uygulama.Workbooks.Add(true);
sayfa = (MyExcel.Worksheet)uygulama.ActiveSheet;
int rowindex;
int colindex;
int egt_tsttip;//0 ise eğitim , 1 ise test
int egt = 0; int test = 0;
testsayisi = Convert.ToInt16(Math.Ceiling(satirsayisi * numericUpDown2.Value /
100));
egtsayisi = satirsayisi - testsayisi;
egitimgirislere = new double[egtsayisi, sutunsayisi];
tstgirislere = new double[testsayisi, sutunsayisi];
sorgula = new double[1, sutunsayisi];
bektmgrs = new double[satirsayisi];
cvr_bektmgrs = new double[satirsayisi];
egttstmi = new char[satirsayisi];
richTextBox1.AppendText("\n Test Sayısı: " + testsayisi + "\n Eğitim sayısı : " +
egtsayisi + "\n");
label25.Text = "Test Sayısı: " + testsayisi + "\n Eğitim sayısı : " + egtsayisi + "\n";
egt_tsttip = 0;
for (rowindex = 0; rowindex < satirsayisi; rowindex++)
{
git:
if (egt + test <= satirsayisi)
{

```

EK-1. (Devam) CPEP eğitim programı kodları

```

if (egt_tsttip == 0)
{
    egt_tsttip = 1;
    if (egt >= egtsayisi)
    {
        egt_tsttip = 1;
        goto git;
    }
    egtstmi[rowindex] = 'E';
    richTextBox1.AppendText("E---");
    for (colindex = 0; colindex < sutunsayisi; colindex++)
    {
        DataGridViewCell cell = dataGridView1[colindex, rowindex];//http://csharp.net-
        informations.com/excel/csharp-excel-datagridview.htm
        egitimgiriler[egt, colindex] = Convert.ToDouble(cell.Value);
        richTextBox1.AppendText(egitimgiriler[egt, colindex].ToString() + "*");
    }
    bektmgrs[strtst++] = egitimgiriler[egt, colindex - 1];
    ++egt;
    richTextBox1.AppendText("\n");
}
else if (egt_tsttip == 1)
{
    egt_tsttip = 0;
    if (test >= testsayisi)
    {
        egt_tsttip = 0;
        goto git;
    }
    egtstmi[rowindex] = 'T';
    richTextBox1.AppendText("T---");
    for (colindex = 0; colindex < sutunsayisi; colindex++)
    {
        DataGridViewCell cell = dataGridView1[colindex, rowindex];//http://csharp.net-
        informations.com/excel/csharp-excel-datagridview.htm
        tstgiriler[test, colindex] = Convert.ToDouble(cell.Value);
        richTextBox1.AppendText(tstgiriler[test, colindex].ToString() + "*");
    }
    bektmgrs[strtst++] = tstgiriler[test, colindex - 1]; //sadece çıkış sütununu tutar
    ++test;
    richTextBox1.AppendText("\n");}}}}
private void katman_sayisi_al()
{
    katmansayisihesap = Convert.ToInt16(numericUpDown4.Value) + 2;//Çıkış katmanı
    ve giriş için iki ekledim
    katmantst = katman = new double[katmansayisihesap][];

```

EK-1. (Devam) CPEP eğitim programı kodları

```

katmantst[0] = katman[0] = new double[sutunsayisi - 1]; //Giriş katmanı
katmantst[katmansayisihesap - 1] = katman[katmansayisihesap - 1] = new
double[1]; //Çıkış katmanı
try
{ //Gizli katmanlar
katmantst[1]=katman[1] = new double[Convert.ToInt16(textBox2.Text)];
katmantst[2]=katman[2] = new double[Convert.ToInt16(textBox3.Text)];
katmantst[3]=katman[3] = new double[Convert.ToInt16(textBox4.Text)];
katmantst[4]=katman[4] = new double[Convert.ToInt16(textBox5.Text)];
}
catch
{
if (textBox2.Enabled == false) { textBox2.Text = ""; }
if (textBox3.Enabled == false) { textBox3.Text = ""; }
if (textBox4.Enabled == false) { textBox4.Text = ""; }
if (textBox5.Enabled == false) { textBox5.Text = ""; }
}}
double range;
private void agirlik_ata()
{
//Ağırlık ataması yapıyorum
agirlik_tum = new double[katmansayisihesap - 1][[]];
eski_adim = new double[katmansayisihesap - 1][[]];
Random rand = new Random();
for (int katman_a = 0; katman_a < katmansayisihesap - 1; ++katman_a)
{
agirlik_tum[katman_a] = new double[(katman[katman_a + 1].Length)][[]];
eski_adim[katman_a] = new double[(katman[katman_a + 1].Length)][[]];
for (int neurons = 0; neurons < katman[katman_a + 1].Length; ++neurons)
{
agirlik_tum[katman_a][neurons] = new double[katman[katman_a].Length];
eski_adim[katman_a][neurons] = new double[katman[katman_a].Length]; //son sutun
cikis oldugundan geri kalan giriş
for (int agirlik_ktmn = 0; agirlik_ktmn < katman[katman_a].Length;
++agirlik_ktmn)
{
agirlik_tum[katman_a][neurons][agirlik_ktmn] = (range * (rand.Next()% 1000) /
500) - range;
richTextBox2.AppendText(katman_a.ToString() + ". katman" + neurons.ToString()
+ ". neron" + agirlik_ktmn.ToString() + ".
agirlik"+agirlik_tum[katman_a][neurons][agirlik_ktmn].ToString()+"\n");}}}}
public void cikis_hesap(double[,] girisler,int csatir)
{
cikis_son_neuron = 0;
for (i = 0; i < katmansayisihesap - 1; i++)

```


EK-1. (Devam) CPEP eğitim programı kodları

```

{
for (neuron = 0; neuron < katman[i + 1].Length; neuron++)
{
for (j = 0; j < katman[i].Length; j++)
{//NET HESAPLANIYOR
if (i == 0)//ilk katman
{
katman[i + 1][neuron] += girisler[csatir, j] * agirlik_tum[i][neuron][j];
}
else if (i < katmansayisihesap - 1)//diğer katmanlar
{
katman[i + 1][neuron] += katman[i][j] * agirlik_tum[i][neuron][j];
}
}
else
MessageBox.Show("ileri hesapta hata");}
//ÇIKIŞ HESAPLANIYOR
katman[i + 1][neuron] = 1 / (1 + Math.Exp(-1 * katman[i + 1][neuron]));
}}
cikis_son_neuron = katman[i][0];//neron çıkışı hesaplanıyor.}
public void delta_sayi_belirle()
{
for (i = katmansayisihesap - 1; i > 1; i--)//son ve ara katmanlar arasındaki hata oranı
{
delta[i - 1] = new double[katman[i - 1].Length];
deltaq[i - 1] = new double[katman[i - 1].Length];
eski_gradient[i - 1] = new double[katman[i - 1].Length];
}}
public void delta_hesap()
{
eski_gradient[katmansayisihesap - 1][0] = delta[katmansayisihesap - 1][0];
delta[katmansayisihesap - 1][0] = cikis_son_neuron * (1 - cikis_son_neuron) *
hata_localgrediet;
for (i = katmansayisihesap - 1; i > 1; i--)//son ve ara katmanlar arasındaki hata oranı
{
for (neuron = 0; neuron < katman[i - 1].Length; neuron++)
{topla_delta = 0;
for (j = 0; j < katman[i].Length; j++)//ilerde çıktı katmanı iki ve daha çok yapılabilir
{topla_delta+=agirlik_tum[i - 1][j][neuron] * delta[i][j];}
eski_gradient[i - 1][neuron] = delta[i - 1][neuron];
delta[i-1][neuron]= topla_delta * katman[i - 1][neuron] * (1 - katman[i -
1][neuron]);
deltaq[i-1][neuron]+=delta[i - 1][neuron];
}}}
}
double momentum;
public void agirlik_guncelle_onlinebackprop()

```

EK-1. (Devam) CPEP eğitim programı kodları

```

{
for (i = 0; i < katmansayisihesap - 1; i++)
{
for (neuron = 0; neuron < katman[i + 1].Length; neuron++)
{
for (j = 0; j < katman[i].Length; j++)
{
if (i == 0)//giriş katmanı
{
eski_adim[i][neuron][j]=momentum*eski_adim[i][neuron][j]+ogrnmorn_lrnrate *
egitimgirisler[satir, j] * delta[i + 1][neuron];
agirlik_tum[i][neuron][j] += eski_adim[i][neuron][j];
}
else if (i < katmansayisihesap - 1)
{
eski_adim[i][neuron][j] = momentum * eski_adim[i][neuron][j] + ogmorn_lrnrate
* katman[i][j] * delta[i + 1][neuron];
agirlik_tum[i][neuron][j] += eski_adim[i][neuron][j];
}}}}
public void agirlik_guncelle_batchbackprop()
{
for (i = 0; i < katmansayisihesap - 1; i++)
{
for (neuron = 0; neuron < katman[i + 1].Length; neuron++)
{
for (j = 0; j < katman[i].Length; j++)
{
if (i == 0)//giriş katmanı
{
eski_adim[i][neuron][j] += momentum * eski_adim[i][neuron][j] + ogmorn_lrnrate
* egitimgirisler[satir, j] * delta[i + 1][neuron];
}
else if (i < katmansayisihesap - 1)
{
eski_adim[i][neuron][j] += momentum * eski_adim[i][neuron][j] + ogmorn_lrnrate
* katman[i][j] * delta[i + 1][neuron];
}
if (satir == satir_hesap - 1)
{
agirlik_tum[i][neuron][j] += eski_adim[i][neuron][j];
eski_adim[i][neuron][j] = 0;
}}}} }
double[][][] eski_adim;
static double yeni_gradient, yeni_adim,quick_factor;
public void agirlik_guncelle_quickprop()

```

EK-1. (Devam) CPEP eğitim programı kodları

```

{
quick_factor=quick_const/(1+quick_const);
for (i = 0; i < katmansayisihesap - 1; i++)
{
for (neuron = 0; neuron < katman[i + 1].Length; neuron++)
{
for (j = 0; j < katman[i].Length; j++)
{
yeni_adim =yeni_gradient= 0;
if (i == 0)//giriş katmanı
{
yeni_adim += ogrnmorn_lrnrte * delta[i + 1][neuron] * egitimgirisler[satir, j];
}
else if (i < katmansayisihesap - 1)
{
//yeni_gradient = -1 * deltaq[i + 1][neuron] * katman[i][j];
yeni_adim += ogrnmorn_lrnrte * delta[i + 1][neuron] * katman[i][j];
}
if (eski_adim[i][neuron][j] > 0
&&(!double.IsPositiveInfinity(eski_adim[i][neuron][j])))
{
if (delta[i+1][neuron]>=(quick_factor * eski_gradient[i + 1][neuron]))
{
yeni_adim += quick_const * eski_adim[i][neuron][j];
}
else
{
yeni_adim += delta[i + 1][neuron] / (eski_gradient[i + 1][neuron] - delta[i +
1][neuron]) * eski_adim[i][neuron][j];
}}
else if (eski_adim[i][neuron][j] <
0&&(!double.IsNegativeInfinity(eski_adim[i][neuron][j])))
{
if (delta[i + 1][neuron] <= (quick_factor * eski_gradient[i + 1][neuron]))
{
yeni_adim += quick_const * eski_adim[i][neuron][j];
}
else
{
yeni_adim += delta[i + 1][neuron] / (eski_gradient[i + 1][neuron] - delta[i +
1][neuron]) * eski_adim[i][neuron][j];
}}
eski_adim[i][neuron][j] = yeni_adim;
agirlik_tum[i][neuron][j] += yeni_adim;

```

EK-1. (Devam) CPEP eğitim programı kodları

```

}}}}
private void ciz_MSE()
{
objspacem.Clear();
objspacem.EnableEvents = true;
int miktar=MSEgrfk.Count;
object[] mse = new object[miktar];

int say_default = 0;
    foreach (object s in MSEgrfk)//array objeye dönüştürülüyor
    { mse[say_default++] = s; }

    object[] strCategory = new object[miktar];
    object[] satir_dizisi = new object[miktar];
for (say_default = 0; say_default < miktar; say_default++)
{
satir_dizisi[say_default] =strCategory[say_default]=say_default + 1;}
Microsoft.Office.Interop.Owc11.ChChart objChart = objspacem.Charts.Add(0);
Microsoft.Office.Interop.Owc11.ChSeries series1;
objChart.Type =
Microsoft.Office.Interop.Owc11.ChartChartTypeEnum.chChartTypeScatterSmoothLine;
objChart.HasTitle = true;
objChart.Title.Caption = " EĞİTİM SONUCU ";
objChart.HasLegend = true;
objChart.Axes[0].HasTitle = true;
objChart.Axes[0].Title.Caption = "Hata miktarı";
objChart.Axes[1].HasTitle = true;
objChart.Axes[1].Title.Caption = "İterasyon sayısı";
objChart.Legend.Position =
Microsoft.Office.Interop.Owc11.ChartLegendPositionEnum.chLegendPositionBottom;
series1 = objChart.SeriesCollection.Add(0);
series1.Caption = "Hata eğrisi";
objChart.SeriesCollection[0].Thickness = 10;
series1.SetData(Microsoft.Office.Interop.Owc11.ChartDimensionsEnum.chDimCategories, -1, strCategory);
series1.SetData(Microsoft.Office.Interop.Owc11.ChartDimensionsEnum.chDimXValues, -1, satir_dizisi);
series1.SetData(Microsoft.Office.Interop.Owc11.ChartDimensionsEnum.chDimYValues, -1, mse);
}
private void ciz_test()
{

```

EK-1. (Devam) CPEP eğitim programı kodları

```

objspace.EnableEvents = true;
int miktar = satirsayisi;
int say_default = 0;
object[] bektg = new object[miktar];
object[] prgsnc = new object[miktar];
object[] strCategory = new object[miktar];
object[] satir_dizisi = new object[miktar];
foreach (object s in cvr_bektmgrs)//double objeye dönüştürülüyor
{
    bektg[say_default++] = s; }
say_default = 0;
foreach (object s in cvr_tgrstst)//double objeye dönüştürülüyor
{
    prgsnc[say_default++] = s; }
for (say_default = 0; say_default < miktar; say_default++)
{
    satir_dizisi[say_default] = strCategory[say_default] = say_default + 1;
}
Microsoft.Office.Interop.Owc11.ChChart objChart = objspace.Charts.Add(0);
Microsoft.Office.Interop.Owc11.ChSeries series1;
Microsoft.Office.Interop.Owc11.ChSeries series2;
objChart.Type =
Microsoft.Office.Interop.Owc11.ChartChartTypeEnum.chChartTypeScatterSmoothLine;
objChart.HasTitle = true;
objChart.Title.Caption = " TEST SONUCU ";
objChart.HasLegend = true;
objChart.Legend.Position =
Microsoft.Office.Interop.Owc11.ChartLegendPositionEnum.chLegendPositionBottom;
objChart.Axes[0].HasTitle = true;
objChart.Axes[0].Title.Caption = "Beklenen/Sonuç";
objChart.Axes[1].HasTitle = true;
objChart.Axes[1].Title.Caption = "Verinin satır numarası";
series1 = objChart.SeriesCollection.Add(0);
series1.Caption = "Beklenen";
series2 = objChart.SeriesCollection.Add(1);
series2.Caption = "Sonuç";
series1.SetData(Microsoft.Office.Interop.Owc11.ChartDimensionsEnum.chDimCategories, -1, strCategory);
series1.SetData(Microsoft.Office.Interop.Owc11.ChartDimensionsEnum.chDimXValues, -1, satir_dizisi);
series1.SetData(Microsoft.Office.Interop.Owc11.ChartDimensionsEnum.chDimYValues, -1, bektg);

```

EK-1. (Devam) CPEP eğitim programı kodları

```

series2.SetData(Microsoft.Office.Interop.Owc11.ChartDimensionsEnum.chDimCategories, -1, strCategory);
series2.SetData(Microsoft.Office.Interop.Owc11.ChartDimensionsEnum.chDimXValues, -1, satir_dizisi);
series2.SetData(Microsoft.Office.Interop.Owc11.ChartDimensionsEnum.chDimYValues, -1, prgsnc);
}
private void textBox1_TextChanged(object sender, EventArgs e)
{
groupBox11.Enabled = groupBox1.Enabled = true;
}
private void numericUpDown1_ValueChanged(object sender, EventArgs e)
{
numericUpDown2.Value = 100 - numericUpDown1.Value;
}
private void numericUpDown2_ValueChanged(object sender, EventArgs e)
{
numericUpDown1.Value = 100 - numericUpDown2.Value;
egitm_test_satir_ata();//eğitim ve test satırları girişe göre atamıyor.
}
private void numericUpDown4_ValueChanged(object sender, EventArgs e)
{
switch (Convert.ToInt16(numericUpDown4.Value))
{
case 1: { groupBox1.Enabled = true; groupBox2.Enabled = false; textBox2.Text = "1"; textBox3.Text = ""; groupBox3.Enabled = false; textBox4.Text = ""; groupBox4.Enabled = false; textBox5.Text = ""; break; }
case 2: { groupBox2.Enabled = true; groupBox3.Enabled = false; textBox2.Text = "1"; textBox3.Text = "1"; textBox4.Text = ""; groupBox4.Enabled = false; textBox5.Text = ""; break; }
case 3: { groupBox3.Enabled = true; groupBox2.Enabled = true; textBox2.Text = "1"; textBox3.Text = "1"; textBox4.Text = "1"; groupBox4.Enabled = false; textBox5.Text = ""; break; }
case 4: { groupBox4.Enabled = true; groupBox2.Enabled = true; textBox2.Text = "1"; textBox3.Text = "1"; textBox4.Text = "1"; textBox5.Text = "1"; groupBox3.Enabled = true; break; }
default:
{
break;
}
}
}
bir = textBox2.Text;
iki = textBox3.Text;
uc = textBox4.Text;
dort = textBox5.Text;

```

EK-1. (Devam) CPEP eğitim programı kodları

```

etiket14());
}
private void Form1_Load(object sender, EventArgs e)
{
// TODO: This line of code loads data into the 'deneDataSet3.Tablo1' table. You can
move, or remove it, as needed.
this.tablo1TableAdapter1.Fill(this.deneDataSet3.Tablo1);
tabControl1.SelectTab(tabPage5);
tabControl1.Refresh();
richTextBox6.Text = "";
saniye = 00;
dak = 00;
saat = 00;
satirsayisi = form1satirsayisi;
sutunsayisi = form1sutunsayisi;
richTextBox6.AppendText("Ağda Toplam \n\n" + satirsayisi + " tane satir ,\n\n" +
sutunsayisi + " tane sütun\n bulunmaktadır.\n\nAktivasyon fonksiyonu :\n Sigmoid");
egitm_test_satir_ata();
bir = "1"; cikis = "1"; iki = "0"; uc = "0"; dort = "0";
etiket14());
System.Windows.Forms.Control.CheckForIllegalCrossThreadCalls = false;
MSEgrfk = new ArrayList();
//SORGULA TABI
for (int sayim = 0; sayim < dataGridView1.Columns.Count-1; sayim++)//sorgula tabı
dolduruluyor
{
dataGridView4.Columns.Add(dataGridView1.Columns[sayim].Name,
dataGridView1.Columns[sayim].Name);
}
dataGridView4.Rows.Add();
label6.Text = dataGridView1.Columns[sutunsayisi - 1].Name + "Değeri :";
}
private void numericUpDown3_Click(object sender, EventArgs e)
{
radioButton3.Checked = true;
}
private void textBox8_Click(object sender, EventArgs e)
{
radioButton2.Checked = true;
}
private void radioButton3_CheckedChanged(object sender, EventArgs e)
{
numericUpDown3.Visible = true; textBox8.Visible = false;
}
private void radioButton2_CheckedChanged(object sender, EventArgs e)

```

EK-1. (Devam) CPEP eğitim programı kodları

```

{
numericUpDown3.Visible = false; textBox8.Visible = true;
}
private void timer1_Tick(object sender, EventArgs e)
{
Thread.Sleep(100);
if (dur_iter == bolum_sonuc && dur == 0) { satir_hesap = kalan_sonuc; dur = 1;
} //iterasyon durdurma kriteri
for (satir = 0; satir < satir_hesap; ++satir)
{
katman_sifirla_ikibyt(katman);
++DENEME_SAY;
itersaysonuc.Text = DENEME_SAY.ToString();
itersaysonuc.Refresh();
cikis_hesap(egitimgirisler,satir);
hata_localgrediet =egitimgirisler[satir,sutunsayisi-1] - cikis_son_neuron;
hata_toplam += Math.Pow(hata_localgrediet,2);
delta_hesap();
if (radioButton4.Checked)//online back prop. güncelle
{
agirlik_guncelle_onlinebackprop();
}
else if (radioButton5.Checked)//online back prop. güncelle
{
agirlik_guncelle_batchbackprop();
}
else if (radioButton1.Checked)
{
if (DENEME_SAY <= satir_hesap)
{
momentum = 0.8;
agirlik_guncelle_onlinebackprop}
else
{ agirlik_guncelle_quickprop(); }
}
if (satir == satir_hesap - 1)
{
hata_toplam = 0.5 * hata_toplam; MSEsonuc.Text =
hata_toplam.ToString("0.#####");
MSEgrfk.Add(hata_toplam);
hata_toplam = 0;}
progressBar1.Value +=1;}
++dur_iter;
if (dur_iter > bolum_sonuc)

```


EK-1. (Devam) CPEP eğitim programı kodları

```

{ timer1.Stop(); timer2.Stop(); label18.Visible = progressBar1.Visible = false;
objspacem.EnableEvents = true;ciz_MSE(); objspacem.Visible =
true;MessageBox.Show("timer iş bitti" + DENEME_SAY.ToString() + "kez
çalıştı");
label26.Text = iterasyon_sayisi + " iterasyondaki\n\n Toplam hata:\n" +
MSEsonuc.Text + "\n\n Geçen süre\n" + label16.Text;}}
double ayni;
int egt_say,tst_say; DataRow data_satir; Color renk; DataGridViewCellStyle
style_backcolor = new DataGridViewCellStyle();
private void timer3_Tick(object sender, EventArgs e)
{
progressBar2.Value += 1;
katman_sifirla_ikibyt(katman);
data_satir = deneDataSet.Tables[0].NewRow();
deneDataSet.Tables["Tablo1"].Rows.Add(data_satir);
if (egttstmi[strtst].Equals('E'))
{
DataGridViewCellStyle stil = new DataGridViewCellStyle();
stil.ForeColor = Color.DarkGreen; // yeni stil tanımlanıyor...
dataGridView2.Rows[strtst].DefaultCellStyle = stil;
cikis_hesap(egitimgirisler,egt_say++);
tgrstst[strtst] = cikis_son_neuron;}
else if (egttstmi[strtst] == 'T')
{
DataGridViewCellStyle stil = new DataGridViewCellStyle();
stil.ForeColor = Color.Brown; // yeni stil tanımlanıyor...
dataGridView2.Rows[strtst].DefaultCellStyle = stil;
cikis_hesap(tstgirisler,tst_say++);
tgrstst[strtst] = cikis_son_neuron;
}
if (checkBox1.Checked)
{
cvr_bektmgrs[strtst] = bektmgrs[strtst] * 67.4 - 37;
cvr_tgrstst[strtst] = tgrstst[strtst]*67.4-37;
}
else
{
cvr_bektmgrs[strtst] = bektmgrs[strtst];
cvr_tgrstst[strtst] =tgrstst[strtst];
}
data_satir["BEKLENEN"] =egttstmi[strtst]+"/ "+cvr_bektmgrs[strtst].ToString();
data_satir["ÇIKIŞ"] =cvr_tgrstst[strtst];
data_satir["HATA"] =cvr_bektmgrs[strtst] -cvr_tgrstst[strtst];
//data_satir["STRTÜR"] = egttstmi[strtst];
Thread.Sleep(10);

```

EK-1. (Devam) CPEP eğitim programı kodları

```

tablo1TableAdapter.Update(data_satir);
++strtst;
if (strtst >= satirsayisi)
{
strtst = 0; timer3.Stop(); progressBar2.Visible = label19.Visible = false;
objspace.Visible = true;ciz_test(); /*timer4.Start(); */MessageBox.Show("test işlemi
tamamlandıtimer3stop"); }
}
public void etiket14()
{
ara = bir + " - " + iki + " - " + uc + " - " + dort ;
label14.Text = "Giriş Katmanı Ara Katman Çıkış katmanı\n " + giris + " " + ara +
"" + cikis;
label14.Update();
label24.Text="Ağ yapısı\n"+giris+"-"+ara+"-"+cikis;
label24.Update();
}
private void textBox2_TextChanged(object sender, EventArgs e)
{
bir = textBox2.Text;
etiket14();
}
private void textBox3_TextChanged(object sender, EventArgs e)
{
iki = textBox3.Text;
etiket14();
}
private void textBox4_TextChanged(object sender, EventArgs e)
{
uc = textBox4.Text;
etiket14(); }
private void textBox5_TextChanged(object sender, EventArgs e)
{
dort = textBox5.Text;
etiket14();}
private void pictureBox1_Click(object sender, EventArgs e)
{
Form2 form2 = new Form2();
Form1.ActiveForm.Hide();
form2.ShowDialog();
}
private void pictureBox2_Click(object sender, EventArgs e)
{
Application.Exit();
}
}

```

EK-1. (Devam) CPEP eğitim programı kodları

```

int kapat =0;
private void Form1_FormClosing(object sender, FormClosingEventArgs e)
{
if (kapat==0)
{
DialogResult dugmesec = MessageBox.Show("Belgeyi kaydetmek istermisiniz?",
"QUICK PROPOGATION İLE ..... EĞİTİMİ",
MessageBoxButtons.YesNoCancel, MessageBoxIcon.Asterisk);
if (dugmesec == DialogResult.Yes)
{
//test sonucundaki datagridview2 yi excel dosyasına aktarsın*****
e.Cancel = true; kapat =0;
}
else if (dugmesec == DialogResult.No) { Application.Exit(); kapat = 1; }
else { e.Cancel = true; kapat =0; }
}
else { Application.Exit(); }}
private void pictureBox3_Click(object sender, EventArgs e)
{
tabControl1.SelectTab(tabPage2);
tabControl1.Refresh();
salise = saniye = dak = 0; progress = 0;
agyapisisonuc.Text = giris + "-" + ara + "-" + cikis;
objspacem.Visible = false; objspace.Visible = false;
deneDataSet.Clear();label18.Visible = progressBar1.Visible = true;
//*****toplam hata kontrolünü sildin onu unutma
MSEgrfk.Clear();
// katman_sifirla_tekbyt();
testediliyor = false;
hata_toplam = DENEME_SAY = 0;
dur_iter = 1; dur = 0;
iterasyon_sayisi = Convert.ToInt32(numericUpDown3.Value); kalan_sonuc = new
Int32();
progressBar1.Minimum = 0;//progressBarımızın ilk değeri
progressBar1.Maximum = iterasyon_sayisi;//progressBarımızın son değeri
progressBar1.Step = 1;//Kaçar kaçır artacağı
progressBar1.Value = 0;//Mevcut değerinin ne olacağı
timer2.Start();//program saati
katman_sayisi_al();//girilen katman sayısı alınıyor.
range =Convert.ToDouble(textBox9.Text);
agirlik_ata();//ilk ağırlık ataması yapıyorum.
katman_sifirla_ucbyt(eski_adim,'S');
delta = new double[katmansayisihesap][];
deltaq = new double[katmansayisihesap][];
eski_gradient = new double[katmansayisihesap][];

```

EK-1. (Devam) CPEP eğitim programı kodları

```

ogrnorn_lrnrate = Convert.ToDouble(textBox7.Text);
delta[katmansayisihesap - 1] = new double[1];
    deltaq[katmansayisihesap - 1] = new double[1];
    eski_gradient[katmansayisihesap - 1] = new double[1];//çıkış katmanında bir
neuron olduğundan "1" verdim.
    delta_sayi_belirle();
if (radioButton3.Checked)//İTERASYON SAYISINA GÖRE DURDURMA
{
if (iterasyon_sayisi >= egtsayisi)
{
satir_hesap = egtsayisi; kalan_sonuc = iterasyon_sayisi % egtsayisi; bolum_sonuc =
(iterasyon_sayisi - kalan_sonuc) / egtsayisi + 1;
MessageBox.Show("bolum" + bolum_sonuc.ToString() + " kalan" +
kalan_sonuc.ToString());}
else { bolum_sonuc = 1; satir_hesap = iterasyon_sayisi; dur = 1; }//kalan_sonuc
olmadığı için aşağıdaki durlu if deyimine gerek yok
}
timer1.Start();//ReportProgress metodu ile hesapladouble.NaN değeri, raporlama
amacı ile ProgressChanged olayına gönderiyoruz.
}
private void pictureBox4_Click(object sender, EventArgs e)
{
timer1.Stop(); timer2.Stop(); timer3.Stop();//tsagirlk_guncelle.Suspend();
tsdelta_hesap.Suspend(); tscikis_hesap.Suspend();
}
private void katman_sifirla_tekbyt(double[] ktmn)
{
for (int i = 0; i < katmansayisihesap; i++)
{
ktmn[i] = 0;} }
private void katman_sifirla_ikibyt(double[][] ktmn)
{
for (int i = 0; i < katmansayisihesap; i++)
{
for (int y = 0; y < katman[i].Length; y++)
{ktmn[i][y] = 0;} }
}
private void katman_sifirla_ucbyt(double[][][] ktmn,char s)
{
for (int i = 0; i < katmansayisihesap-1; i++)
{
for (int y = 0; y < katman[i + 1].Length; y++)
{
for (int z = 0; z < katman[i].Length; z++)
{

```

EK-1. (Devam) CPEP eğitim programı kodları

```

if (s.Equals('S'))
{ktmn[i][y][z] = 0;}
else if(s.Equals('Y'))
{
richTextBox5.AppendText(i.ToString() + ". katman" + y.ToString() + ". neron" +
z.ToString() + ". ağırlık " + agirlik_tum[i][y][z].ToString() + "\n");
}}}}
private void pictureBox5_Click(object sender, EventArgs e)
{//sorgula datagrid sutun isimleri veriliyor
katman_sifirla_ikibyt(katman);//katmanı sıfırlıyor
progressBar2.Visible = label19.Visible = true;
progressBar2.Minimum = 0;//progressBarımızın ilk değeri
progressBar2.Maximum = satirsayisi;//progressBarımızın son değeri
progressBar2.Step = 1;//Kaçar kaçır artacağı
progressBar2.Value = 0;//Mevcut değerinin ne olacağı
tabControl1.SelectTab(tabPage4);
tabControl1.Refresh();
testediliyor = true;
//Ağırlıkların son hali yazılıyor
katman_sifirla_ucbyt(agirlik_tum, 'Y');
//TEST EDİLİYOR
deneDataSet.Clear();
objspace.Clear();
objspace.Visible = false;
dataGridView2.ForeColor = Color.Black;
tgrstst = new double[satirsayisi];
cvr_tgrstst = new double[satirsayisi];
egt_say = tst_say=strtst = 0;
timer3.Start();
}
int saniye, dak, salise,saat;
private void timer2_Tick(object sender, EventArgs e)
{
saat = (dak == 60 ? saat + 1 : saat);
dak = (saniye == 60 ? dak + 1 : dak);
saniye = (saniye == 60 ? 00 : saniye + 1);
dak = (dak == 60 ? 00 : dak);
label16.Text = saat.ToString() + ":" + dak.ToString() + ":" + saniye.ToString();}
private void radioButton4_CheckedChanged(object sender, EventArgs e)
{
if (radioButton4.Checked || radioButton5.Checked)
{
algoritmasonuc.Text = radioButton5.Text;
if (radioButton4.Checked) algoritmasonuc.Text = radioButton4.Text;
momentum = Convert.ToDouble(textBox1.Text);
}
}

```

EK-1. (Devam) CPEP eğitim programı kodları

```

label11.Enabled = textBox6.Enabled = false; label20.Enabled = textBox1.Enabled =
true;}
else
{quick_const = Convert.ToDouble(textBox6.Text);
label20.Enabled = textBox1.Enabled = false; label11.Enabled = textBox6.Enabled =
true;}}
private void radioButton5_CheckedChanged(object sender, EventArgs e)
{if (radioButton4.Checked || radioButton5.Checked)
{algoritmasonuc.Text = radioButton5.Text;
if (radioButton4.Checked) algoritmasonuc.Text = radioButton4.Text;
momentum = Convert.ToDouble(textBox1.Text);
label11.Enabled = textBox6.Enabled = false; label20.Enabled = textBox1.Enabled =
true;}
else
{algoritmasonuc.Text = radioButton1.Text;
quick_const = Convert.ToDouble(textBox6.Text);
label20.Enabled = textBox1.Enabled = false; label11.Enabled = textBox6.Enabled =
true;}}
private void radioButton1_CheckedChanged(object sender, EventArgs e)
{if (radioButton4.Checked || radioButton5.Checked)
{algoritmasonuc.Text = radioButton5.Text;
if (radioButton4.Checked) algoritmasonuc.Text = radioButton4.Text;
momentum = Convert.ToDouble(textBox1.Text);
label11.Enabled = textBox6.Enabled = false; label20.Enabled = textBox1.Enabled =
true;}
else{algoritmasonuc.Text = radioButton1.Text;
quick_const = Convert.ToDouble(textBox6.Text);
label20.Enabled = textBox1.Enabled = false; label11.Enabled = textBox6.Enabled =
true;}}
double cvr_tgirstst;
private void button1_Click(object sender, EventArgs e)
{for (int colindex = 0; colindex < sutunsayisi-1; colindex++)
{DataGridViewCell cell = dataGridView4[colindex,0];
if (colindex.Equals(sutunsayisi - 2))
sorgula[0, colindex] = (Convert.ToDouble(cell.Value) -35 ) / 43;
else{sorgula[0, colindex] = Convert.ToDouble(cell.Value);}}
katman_sifirla_ikibyt(katman);
cikis_hesap(sorgula,0);
if (checkBox1.Checked)
{ cvr_tgirstst = cikis_son_neuron*67.4 - 37;}
else
{ cvr_tgirstst = cikis_son_neuron;}textBox10.Text = cvr_tgirstst.ToString(); } }

```

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : BİLEN, Gamze Hatice
Uyruğu : T.C.
Doğum tarihi ve yeri : 17.09.1985 Gölbaşı
Medeni hali : Bekar
Telefon : (312)4852317
e-mail : gamze.7483@hotmail.com.tr

Eğitim Derece

Eğitim Birimi

Mezuniyet tarihi

Lisans	Gazi Üniversitesi/Teknik Eğitim	2008
Lise	Dikmen And. Lisesi	1999

İş Deneyimi Yıl

Yer

Görev

2008-2010	Kızılırmak Ç.P.L	Öğretmen
2011-	Gölbaşı End. Mes.Lisesi	Öğretmen

Yabancı Dil

İngilizce

Hobiler

Bilgisayar teknolojileri