

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**YALITKAN MADDELERDE ELEKTRİKSEL DELİNME
DAYANIMININ YAPAY SINIR AĞLARI İLE BELİRLENMESİ**

**T.C. YÜKSEKÖĞRETİM KU
DOKÜMANTASYON MERKEZİ**

**YÜKSEK LİSANS TEZİ
Elk. Müh. Melike OĞUZ
504991409**

101409

**Tezin Enstitüye Verildiği Tarih : 11 Haziran 2001
Tezin Savunulduğu Tarih : 29 Haziran 2001**

**Tez Danışmanı : Doç. Dr. Özcan KALENDERLİ
Diğer Jüri Üyeleri : Prof.Dr. Kevork MARDİKYAN
Doç. Dr. Serhat ŞEKER**

Özcan
K. Mardikyan
Şeker

HAZİRAN 2001

ÖNSÖZ

Yüksek lisans tezimin konusu; “Yalıtkan Maddelerde Elektriksel Delinme Dayanımının Yapay Sinir Ağları ile Belirlenmesi”dir. Bu çalışmada, yapay sinir ağları tanıtılmış ve yöntemin katı yalıtkan maddelerin delinme dayanımının hesabına uygulaması yapılmıştır.

Tüm araştırmalarımda bana yardımcı olan yüksek lisans tezimin de danışmanı Sayın Doç. Dr. Özcan Kalenderli’ye, yardımlarını benden esirgemeyen aileme ve arkadaşlarıma teşekkür ederim.

Haziran 2001

MELİKE OĞUZ

İÇİNDEKİLER

KISALTMALAR	vi
TABLO LİSTESİ	viii
ŞEKİL LİSTESİ	ix
SEMBOL LİSTESİ	xii
ÖZET	xiii
SUMMARY	xiv
1. GİRİŞ	1
2. YAPAY SİNİR AĞLARI	2
2.1. Giriş	2
2.2. Yapay Sinir Ağlarının Temel İlkeleri	3
2.3. Yapay Sinir Ağlarının Tarihsel Gelişimi	5
2.4. Basit Nöron Yapısı	6
2.5. Aktivasyon (Uyarma) İşlevi	7
2.6. Yapay Sinir Ağı Yapıları	9
2.6.1. Tek katmanlı ve çok katmanlı sinir ağları	10
2.7. Yapay Sinir Ağlarında Öğrenme Algoritmaları	11
2.7.1. Eğitici öğrenme algoritmaları	13
2.7.1.1. Perseptron öğrenme kuralı	13
2.7.1.2. Widrow-Hoff öğrenme kuralı	
En küçük ortalama hata karesi öğrenme kuralı	
(LMS – Least Mean Square Error)	18
2.7.1.3. Grossberg öğrenme kuralı	20
2.7.1.4. BP – (Backpropagation)	
Geriye yayılım algoritması	20
2.7.1.5. Momentum katsayılı geriye yayılım algoritması	
BPM – Backpropagation with momentum	23
2.7.1.6. Delta Bar Delta algoritması – DBD	24
2.7.1.7. Decoupled Momentum algoritması – DM	25
2.7.1.8. Rprop algoritması	26
2.7.1.9. Adaptif geriye yayılım algoritması	
ABP– Adaptive Backpropagation	27
2.7.1.10. Değiştirilmiş adaptif geriye yayılım algoritması	
MABP – Modified Adaptive Backpropagation	27
2.7.1.11. Adaptif kazançlı geriye yayılım algoritması	
BPG – Backpropagation with Adaptive Gains	28

2.7.1.12. Newton'un ikinci dereceden türev alınan öğrenme algoritması	28
2.7.1.13. İkinci dereceden türev alınan Gauss-Newton öğrenme algoritması	28
2.7.1.14. Levenberg-Marquardt öğrenme algoritması	29
2.7.1.15. Quickprop öğrenme algoritması	29
2.7.1.16. Birleşik gradyan azalması algoritması CGD - Conjugate Gradient Descent	30
2.7.1.17. Eğitici öğrenme algoritması uygulaması - Sayısal Örnek	30
2.7.2. Eğitici öğrenme Algoritmaları	32
2.8. Yapay Sinir Ağlarının Kullanım Nedenleri	34
2.9. Yapay Sinir Ağlarının Dezavantajları	34
2.10. Yapay Sinir Ağlarının Potansiyel Uygulama Alanları	35
3. HATANIN GERİYE YAYILIMI ALGORİTMASI	37
3.1. Giriş	37
3.2. Hata Minimizasyonu	37
3.3. Yapay Sinir Ağlarında Öğrenme ve Ezberleme Problemi	38
3.4. BP Algoritmasında Ağırlıkların Yenilenmesi	39
3.4.1. Aktivasyon işlevinin türevleri	41
3.4.2. Çıkış tabakasındaki ağırlıkların yenilenmesi	41
3.4.3. Ara tabaka ağırlıklarının yenilenmesi	43
3.5. BP Algoritmasında Saklı Katman Nöronlarının Sayısının Belirlenmesi	44
3.6. Geriye Yayılım Algoritmasının Uygulama Alanları	45
3.6.1. Geriye yayılım algoritması ile veri sıkıştırma	45
3.6.2. Geriye yayılım algoritması ile karakter algılama	46
3.6.3. Geriye yayılım algoritması ile yalıtkan tanılaması	47
3.6.4. Üniversal motor dinamiğinin tanılanması	48
3.6.5. Elektrik tesislerinde mesafe rölesi ve mesafe korumasında geriye yayılım algoritmasının kullanımı	49
3.6.6. İşlev yaklaşıklama	50
3.6.7. İletim hatlarında taşıma kapasitesinin bulunması	51
3.6.8. Elektrokardiyogram verilerinin tanılanması	51
4. YAPAY SİNİR AĞLARININ ELEKTRİKSEL DELİNME DAYANIMI DEĞERLERİNİN SAPTANMASINDA KULLANILMASI	53
4.1. Giriş	53
4.2. Katı Yalıtkanlarda Delinme Türleri	53
4.3. Polyester Film Tabakalarında Elektriksel Delinme Dayanımı	54
4.4. Giriş Değerleri Olarak Sinir Ağına Tanıtılacak Deneysel Verilerin Elde Edilmesi	55

5. BİLGİSAYAR UYGULAMASI VE SONUÇLARI	57
5.1. Uygulama	57
5.1.1. Hata limit değerinin değiştirilmesi ile gözlenen değişiklikler	59
5.1.2. Saklı katmandaki nöron sayısı artırılarak yapılan denemeler	65
5.1.3. Momentum değerinin artırılması ile elde edilen sonuçlar	67
5.1.4. Öğrenme katsayısının artırılması ile elde edilen sonuçlar	71
5.1.5. İdeal parametre değerleri belirlendikten sonra yapılan uygulamalar	77
6. SONUÇLAR VE ÖNERİLER	88
KAYNAKLAR	90
ÖZGEÇMİŞ	94



KISALTMALAR

ABP	:Adaptive Back Propagation (Adaptif geriye yayılım algoritması)
ACE	:Adaptive Critic Element
AHN	:Adaptive Hamming Net (Adaptif Hamming ağı)
ART	:Adaptive Resonance Theory (Adaptif rezonans teorisi)
ASE	:Adaptive Search Element
ASMOD	:Adaptive Spline Modelling of Observation Data
BAM	:Bidirectional Associative Memory
BFN	:Block Feedback Network
BON	:Binary Oscillator Network
BP	:Back Propagation (Geriye yayılım algoritması)
BPG	:Back Propagation with Adaptive Gains (Adaptif kazançlı geriye yayılım algoritması)
BPM	:Back Propagation with Momentum (Momentum katsayılı geriye yayılım algoritması)
CGD	:Conjugate Gradient Descent (Birleşik gradyan azalması)
CMAC	:Cerebellar Model Articulation Controller
CPN	:Counter Propagation Network
DBD	:Delta-Bar-Delta
DBD-M	:Delta-Bar-Delta with Adaptive Decoupled Momentum
DPB	:Dynamic Back Propagation
DEKF	:Decoupled Extended Kalman Filter
DM	:Decoupled Momentum
DNMRAC	:Direct Neural Model Reference Adaptive Control
DNU	:Dynamic Neural Unit
DP	:Dynamic Programming
EBPA	:Extended Back Propagation Algorithm (Geliştirilmiş geriye yayılım algoritması)
FEL	:Feedback Error Learning
FIM	:Forward and Inverse Modelling
FLN	:Functional Link Network
FNN	:Feedforward Neural Network
GARIC	:Generalized Approximate Reasoning-Based Intelligent Control
GL	:Generalized Learning
GRL	:Genetic Reinforcement Learning
GLVQ	:Generalized Learning Vector Quantization
GRNN	:General Regression Neural Network
HONN	:Higher Order Neural Network
HS	:History Stack
HSII	:Hyper-Spherical Surface Interactive Interconnection Learning Method
IGLS	:Integrated Gradient and Least Squares
INC/NN	:Inverse Nonlinear Control using Neural Network
KPN	:Knapsack Packing Networks

LM	:Levenberg-Marquardt
LMS	:Least mean square (En küçük ortalama hata karesi kuralı)
LRLS	:Learning by Recursive Least Squares
MABP	:Modified Adaptive Back Propagation (Değiştirilmiş adaptif geriye yayılım algoritması)
MARS	:Multivariate Adaptive Regression Splines
MLANC	:Maximum Likelihood Adaptive Neural Controller
MLP	:Multilayer Perceptron (Çok katmanlı perseptron)
MNN	:Multilayer Neural Network (Çok katmanlı sinir ağı)
NFL	:Neural Feedback Linearization
NIMC	:Neural Internal Model Control
NN	:Neural Network (Sinir ağı)
OANNC	:On-line Adaptive Neural Network Based Controller
OLL	:Optimization Layer by Layer
OLS	:Orthogonal Least Squares
PID	:Proportional Integrate and Derivative
PT	:Polyester
QR-WRLS	:QR-Weighted Recursive Least Squares
RBFN	:Radial Basis Function Network (Radyal bazlı işlev ağı)
RL	:Reinforcement Learning
RNN	:Recurrent Neural Network
RPN	:Ridge Polynomial Network
RTRL	:Real Time Recurrent Learning
SL	:Specialized Learning
SNN	:Single Layer Neural Network (Tek katmanlı sinir ağı)
SOM	:Self Organizing Map (Özdenetimli harita)
SPHS	:Schroeder Phase Harmonic Signal
TD	:Temporal Difference
TDL	:Tapped Delay Lines
YSA	:Yapay Sinir Ağı

TABLO LİSTESİ

	<u>Sayfa No</u>
Tablo 2.1. Yapay sinir ağlarında kullanılan öğrenme algoritmalarının sınıflandırılması	13
Tablo 3.1. İletim hatlarında taşıma kapasitesinin bulunmasında geriye yayılım algoritması ile gözlenen hata dağılımı değerleri.....	51
Tablo 4.1. Farklı kalınlıklarda polyester folyeler için delinme dayanımının sıcaklıkla değişimi (E_d kV/mm).....	56
Tablo 5.1. Farklı sıcaklık değerlerinde yapay sinir ağı ile elde edilen delinme dayanımı değerleri (E_d kV/mm).....	86
Tablo 5.2. Farklı polyester folye kalınlıklarında yapay sinir ağı ile elde edilen delinme dayanımı değerleri (E_d kV/mm).....	86
Tablo 5.3. Farklı polyester folye kalınlıklarında ve sıcaklık değerlerinde deney ve sinir ağı uygulaması ile elde edilen delinme dayanımı değerleri (E_d kV/mm).....	87

ŞEKİL LİSTESİ

	Sayfa No
Şekil 2.1 :Sinir hücresi yapısı.....	2
Şekil 2.2 :YSA'nın tarihsel gelişimi.....	4
Şekil 2.3 :Basit nöron yapısı.....	6
Şekil 2.4.a :Lineer aktivasyon işlevi.....	7
Şekil 2.4.b :Lineer eşik aktivasyon işlevi.....	7
Şekil 2.4.c :Basamak aktivasyon işlevi.....	7
Şekil 2.4.d :Lojistik sigmoid aktivasyon işlevi.....	7
Şekil 2.4.e :Hiperbolik tanjant tipi aktivasyon işlevi.....	7
Şekil 2.5 :İleri beslemeli yapay sinir ağı yapısı.....	9
Şekil 2.6 :Geri beslemeli yapay sinir ağı yapısı.....	10
Şekil 2.7 :Eğitici öğrenme algoritması şeması.....	12
Şekil 2.8 :Perseptron yapısı.....	14
Şekil 2.9 :Adaline yapısı.....	15
Şekil 2.10 :Çözüm uzayı.....	16
Şekil 2.11 :2 Adaline birimi girişi olan Madaline Yapısı.....	17
Şekil 2.12 :Madaline yapısı çözüm uzayı.....	17
Şekil 2.13 :Çok katmanlı perseptron.....	18
Şekil 2.14 :Instar ve Outstar modeli.....	20
Şekil 2.15 :Nonlineer hücre modeli.....	22
Şekil 2.16 :BP Algoritması blok akış şeması.....	23
Şekil 2.17 :Tek katmanlı perseptron kullanılan sayısal örnek.....	31
Şekil 2.18 :Özdenetimli Harita.....	33
Şekil 2.19 :Bilgi Akışı.....	35
Şekil 3.1 :Hata minimizasyonunun geometrik yorumu.....	38
Şekil 3.2 :Eğitimin durdurulması için uygun durdurma bölgesi.....	39
Şekil 3.3 :BP sinir ağı yapısı.....	40
Şekil 3.4 :Yapay sinir ağları ile veri sıkıştırma.....	46
Şekil 3.5 :A harfinin ikili sayı düzeninde kodlanması.....	47
Şekil 3.6 :Giriş katmanı, saklı katman ve çıkış katmanından oluşan ağ yapısı.....	50
Şekil 3.7 :İşlev yaklaşımı.....	51
Şekil 4.1 :Katı yalıtkanlarda delinme dayanımının gerilim uygulama süresine göre değişimi	54
Şekil 4.2 :Deneyde kullanılan elektrot tipi.....	55
Şekil 5.1 :BP algoritması akış şeması.....	57
Şekil 5.2 :Lojistik aktivasyon işlevinin türevi.....	58
Şekil 5.3 :3 nöronlu 1 saklı katmanlı sinir ağında işlev yaklaşımı şekli (E=0.001).....	60
Şekil 5.4 :3 nöronlu 1 saklı katmanlı sinir ağında işlev yaklaşımı şekli (E=0.004).....	61
Şekil 5.5 :3 nöronlu 1 saklı katmanlı sinir ağında işlev yaklaşımı şekli (E=0.01).....	62

Şekil 5.6	:3 nöronlu 1 saklı katmanlı sinir ağında işlev yaklaşımı şekli ($E=0.01$).....	63
Şekil 5.7	:3 nöronlu 1 saklı katmanlı sinir ağında işlev yaklaşımı şekli ($E=0.001$, $R=150000$).....	64
Şekil 5.8	:5 nöronlu 1 saklı katmanlı sinir ağında işlev yaklaşımı şekli ($E=0.001$, $R=150000$).....	65
Şekil 5.9	:7 nöronlu 1 saklı katmanlı sinir ağında işlev yaklaşımı şekli ($E=0.001$, $R=150000$).....	66
Şekil 5.10	:10 nöronlu 1 saklı katmanlı sinir ağında işlev yaklaşımı şekli ($E=0.001$, $R=150000$).....	67
Şekil 5.11	:Saklı katman nöron sayısı 10, $\eta=0.001$ ve $m=0.2$ iken elde edilen eğitim sonuçları.....	68
Şekil 5.12	:Saklı katman nöron sayısı 10, $\eta=0.001$ ve $m=0.4$ iken elde edilen eğitim sonuçları.....	69
Şekil 5.13	:Saklı katman nöron sayısı 10, $\eta=0.001$ ve $m=0.6$ iken elde edilen eğitim sonuçları.....	69
Şekil 5.14	:Saklı katman nöron sayısı 10, $\eta=0.001$ ve $m=0.9$ iken elde edilen eğitim sonuçları.....	70
Şekil 5.15	:Saklı katman nöron sayısı 10, $\eta=0.001$ ve $m=0.99$ iken elde edilen eğitim sonuçları.....	71
Şekil 5.16	:10 nöronlu tek saklı katmanlı sinir ağında $\eta = 0.05$ seçilmesi halinde elde edilen sonuçlar.....	73
Şekil 5.17	:Saklı katman nöron sayısı 10, $\eta=0.03$ ve $m=0.8$ iken elde edilen eğitim sonuçları.....	74
Şekil 5.18	:10 nöronlu tek saklı katmanlı sinir ağında $\eta = 0.015$ seçilmesi durumu.....	74
Şekil 5.19	:10 nöronlu tek saklı katmanlı sinir ağında $\eta = 0.009$ seçilmesi durumu.....	75
Şekil 5.20	:10 nöronlu tek saklı katmanlı sinir ağında $\eta = 0.005$ seçilmesi durumu.....	75
Şekil 5.21	:26 nöronlu tek saklı katmanlı sinir ağında $\eta = 0.97$ seçilmesi durumu.....	77
Şekil 5.22	:Eğitim süresince hata karesinin değişimi.....	78
Şekil 5.23	:23 μ m PT folye kalınlığında elde edilen deney sonuçlarının ağa tanıtılması ile elde edilen grafikte $\eta=0.001$, $m=0.8$, $E=0.001$, $H=10$ alınmıştır.....	79
Şekil 5.24	:36 μ m PT folye kalınlığında elde edilen deney sonuçlarının ağa tanıtılması ile elde edilen grafikte $\eta=0.001$, $m=0.8$, $E=0.001$, $H=10$ alınmıştır.....	80
Şekil 5.25	:50 μ m PT folye kalınlığında elde edilen deney sonuçlarının ağa tanıtılması ile elde edilen grafikte $\eta=0.001$, $m=0.8$, $E=0.001$, $H=10$ alınmıştır.....	80
Şekil 5.26	:100 μ m PT folye kalınlığında elde edilen deney sonuçlarının ağa tanıtılması ile elde edilen grafikte $\eta=0.001$, $m=0.8$, $E=0.001$, $H=10$ alınmıştır.....	81

Şekil 5.27	:200 μ m PT folye kalınlığında elde edilen deney sonuçlarının ağa tanıtılması ile elde edilen grafikte $\eta=0.001$, $m=0.8$, $E=0.001$, $H=10$ alınmıştır.....	81
Şekil 5.28	:(-40°C) sıcaklıkta ve değişen folye kalınlıklarında elde edilen delinme dayanımı değerleri.....	82
Şekil 5.29	:(-20°C) sıcaklıkta ve değişen folye kalınlıklarında elde edilen delinme dayanımı değerleri.....	83
Şekil 5.30	:(0°C) sıcaklıkta ve değişen folye kalınlıklarında elde edilen delinme dayanımı değerleri.....	83
Şekil 5.31	:(20°C) sıcaklıkta ve değişen folye kalınlıklarında elde edilen delinme dayanımı değerleri.....	83
Şekil 5.32	:(40°C) sıcaklıkta ve değişen folye kalınlıklarında elde edilen delinme dayanımı değerleri.....	84
Şekil 5.33	:(60°C) sıcaklıkta ve değişen folye kalınlıklarında elde edilen delinme dayanımı değerleri.....	84
Şekil 5.34	:(80°C) sıcaklıkta ve değişen folye kalınlıklarında elde edilen delinme dayanımı değerleri.....	84
Şekil 5.35	:(100°C) sıcaklıkta ve değişen folye kalınlıklarında elde edilen delinme dayanımı değerleri.....	85
Şekil 5.36	:(120°C) sıcaklıkta ve değişen folye kalınlıklarında elde edilen delinme dayanımı değerleri.....	85
Şekil 5.37	:Artan sıcaklık değerleri ve yalıtkan kalınlığına göre elektriksel delinme dayanımı eğrileri.....	87

SEMBOL LİSTESİ

α	:Momentum katsayısı
β	:Eğim parametresi
D	:Uzaklık
E	:Hata işlevi
e	:Hata terimi
η	:Öğrenme oranı
g	:Kazanç değeri
H	:Hessian matrisi
I	:Birim matris
K_t	:Tork katsayısı
L_a	:Armatür endüktansı
φ	:Hata işlevinin ağırlık değerlerine göre türevi
ψ	:Aktivasyon işlevi
R_a	:Armatür direnci
σ	:Pozitif değerli küçük bir sayı
t	:Zaman
T_g	:Üretilen tork
θ	:Referans (bias) değeri
w	:Ağırlık katsayısı
x	:Giriş değeri
y	:Çıkış değeri
y_d	:Hedeflenen çıkış değeri

YALITKAN MADDELERDE ELEKTRİKSEL DELİNME DAYANIMININ YAPAY SİNİR AĞLARI İLE BELİRLENMESİ

ÖZET

İnsan beyni, beynin özel işlevlerini yerine getirebilmek için paralel bir ağ mimarisi içinde düzenlenmiş hücrelerden oluşur. Beyin, birçok işlemi paralel olarak işleyebilmektedir. Yapay sinir ağları, insan beyni çalışma işlevine benzetim yapılarak sayısal bilgisayarların karşılaştıkları zorluklara çözüm sağlayabilmek amacıyla geliştirilmiştir.

Yapay sinir ağlarında, biyolojik nöronların benzetimi, ağ içerisinde kullanılan giriş çıkış değerleri, aktivasyon işlevleri ve bağlantı ağırlıkları arasındaki ilişkiler düzenlenerek gerçekleştirilir. Bir yapay nörona, diğer nöronlardan ulaşan tüm girdiler, bağlantı ağırlıkları ile çarpılarak bu nöron üzerinde toplam bir etki oluştururlar. Nöron, bu toplam etkiyi aktivasyon işlevine göre yorumlayıp sonuçta bir tepki ya da çıkış üretmektedir.

Yapay sinir ağlarının en yaygın kullanılan algoritması olan geriye yayılım algoritması, denetim altında bir eğitim ile çıkış hatasının alt katmanlara yayılmasını sağlayarak giriş verisine karşılık gelen bir çıkış vektörü bulmayı amaçlar. Hata değeri istenilen seviyeye düşene dek ağırlık değerleri minimizasyon şartına göre adapte edilir.

Yapay sinir ağları uygulamalarında, performans artışı ve yerel minimuma takılmayı engelleyebilmek için, saklı katman nöron sayısı arttırılmalı, öğrenme oranı azaltılmalı, farklı ağırlık katsayıları kullanarak değişik denemeler yapılmalıdır. Duyarlı sonuçlar için, giriş veri kümesinin yeterince büyük olması gerekmektedir.

“Yalıtkan Maddelerde Elektriksel Delinme Dayanımının Yapay Sinir Ağları ile Belirlenmesi” başlıklı tez çalışmasında deneysel olarak farklı sıcaklık ve yalıtkan madde kalınlıklarında elde edilen delinme dayanımı değerleri, geriye yayılım algoritması kullanılarak yapay sinir ağına tanıtılmış ve deneme yanılma yolu ile en uygun çıkışı verecek ağ mimarisi geliştirilmiştir. Hata limit değeri, saklı katman nöron sayısı, momentum katsayısı, öğrenme oranı parametreleri değiştirilerek denemeler yapılmış ve optimum sonuçlar sunulmuştur. Bu yaklaşımla katı yalıtkan maddelerin delinme geriliminin ve delinme dayanımının belirlenmesinde, pahalı, zaman alıcı ve zahmetli deneysel çalışmalara kuramsal destek verilmiştir.

DETERMINATION OF ELECTRICAL BREAKDOWN STRENGTH OF SOLID INSULATING MATERIALS WITH ARTIFICIAL NEURAL NETWORKS

SUMMARY

Human brain is composed of specialized cells which are organized in a parallel way network structure to be able to maintain the functions of the brain. Brain processes have parallel aspects in their structures, although process structure of a numerical computer is rather different. As a solution of such problems artificial neural networks are developed.

The biological neuron is to be resembled by the relationship between input and output values, activation function and connection weights in artificial neural networks. The inputs received by a unit from all other units are multiplied by the connection weights. This forms a total effect on the unit after it is treated upon activation function to form an output.

The popularly used backpropagation algorithm aims to find out an output vector corresponding to the input vector by supervised training and by output error propagation upon hidden layers. The connection weight parameters are trained until the error function reaches the desired minimum level.

In neural network applications to increase performance and to prevent local minimum strike, number of hidden layer neurons should be increased, learning rate should be decreased, trials with randomly different weights should be made.

In the study of the M.Sc. thesis named “Determination of Electrical Breakdown Strength of Solid Insulating Materials with Artificial Neural Networks”, backpropagation algorithm is used. In the application part of the study, experimental results obtained in different temperature and insulator thickness values, are introduced to the neural network and best network structure for the problem is developed by error and trial method. Error minimization is aimed by parameter updates. In this point of view, a new approach is introduced to the breakdown strength value determination of solid insulating materials.

1. GİRİŞ

Bu tezin amacını, yapay sinir ağıları yardımıyla katı yalıtkan maddelerde elektriksel delinme dayanımının hesabı oluşturmaktadır.

Hesaplarda yapay sinir ağına giriş verileri olarak, daha önce deneysel yollardan, farklı sıcaklık değerlerinde, farklı kalınlıklarda katı yalıtkan maddeler için ölçülmüş delinme dayanımı değerleri kullanılmıştır.

Bu kapsamda, bu tezin içinde araştırılması gereken iki ana konu ortaya çıkmaktadır. Bu konular, yapay sinir ağıları ve yapay sinir ağlarının delinme dayanımı hesabında nasıl kullanılabileceği konularıdır.

Tezin birinci bölümünde yapay sinir ağlarının tanıtımı, tarihsel gelişimi, eğitme algoritmaları, yapay sinir ağlarının kullanımının sebepleri, yapay sinir ağlarının klasik yazılımlarla karşılaştırılması, üstünlükleri ve dezavantajları ayrıntılı olarak tartışılmıştır.

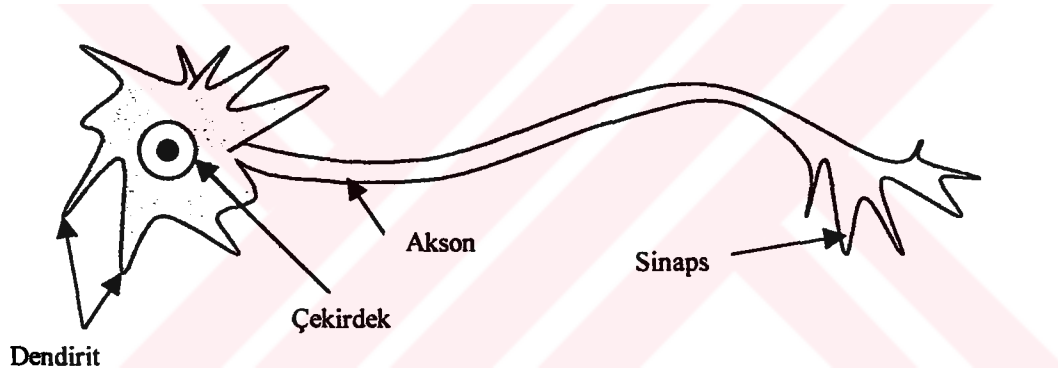
Tezin ikinci bölümünde, yapay sinir ağları kullanılarak ince katı yalıtkan tabakalarda (PT folyeler için) farklı ortamlarda delinme dayanımının hesabı konusu üzerine uygulamaya geçilmiş, adım adım çalışma algoritması anlatılmıştır.

Tezin sonuç kısmında ise yapay sinir ağları uygulamasında yapay sinir ağının ürettiği sonuçların nasıl yorumlanacağı açıklanmış, görüşlere yer verilmiştir.

2. YAPAY SİNİR AĞLARI

2.1. Giriş

Yapay sinir ağları, insan beyninin öğrenme şeklinin uygulanmaya çalışıldığı bir yöntemdir. Yapay sinir ağları yöntemine insan beyninin modellenmesi de denebilir. Yapay sinir ağlarının temelinde, zeka gerektiren işlemlerden oluşan bilgi işleme işlevi vardır. Bu, bilgisayarda uygun bir programlama dili ile algoritmalar geliştirilerek yapılır. Yapay sinir ağlarının çalışma biçimi, gerçek bir sinir hücresinin çalışma biçimine benzer olarak oluşturulduğundan önce bu yapı incelenecektir.



Şekil 2.1- Sinir hücresi yapısı

Canlılardaki sinir sistemi hücrelerine nöron adı verilir. Biyolojik nöronun dört önemli parçası yapay sinir ağlarında da benzer işlevlerle kullanılmıştır;

Dendrit: Diğer nöronlardan gelen işaretleri algılar.

Soma: Sinir hücresinin gövdesidir. Gelen işaretleri toplar. Bir anlamda ağırlık değerleri ile çarpılan girişlerin toplanması işlemini gerçekleştiren toplayıcı görevini üstlenmiştir.

Akson: Nöronlar arasındaki iletim hattını oluşturur.

Sinaps: Nöronun çıkış işaretini iletir.

Şekil 2.1’de gösterilen biyolojik sinir hücresinde sinir hücresinin dendritlerine ulaşan bilgiler toplanır ve sinir hücresinin sonundaki dendritlere sinir hücresinin aksonu boyunca ulaştırılır. Bir nöronun aksonundan geçen sinir darbeleri, özel

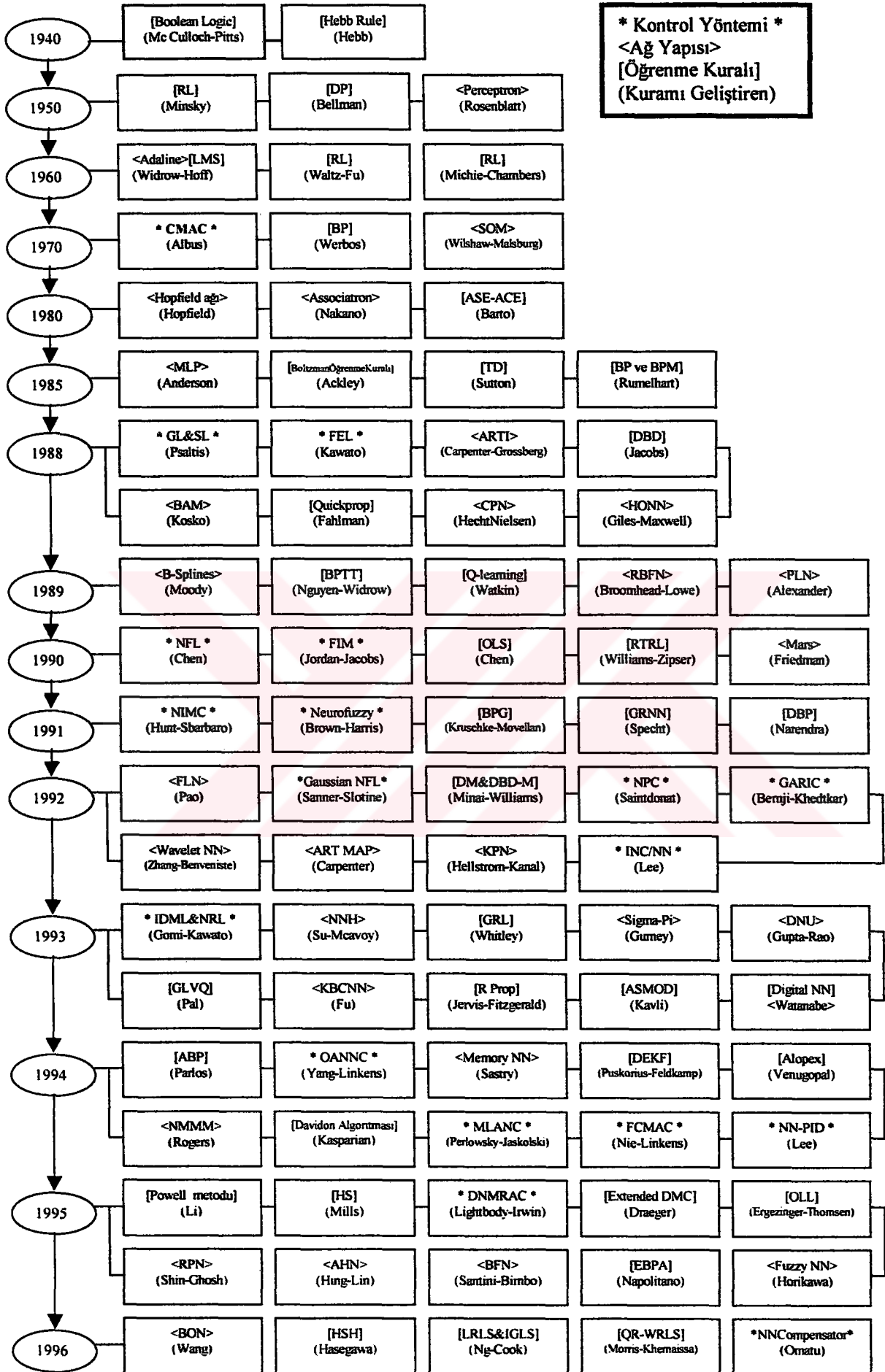
baloncuklardaki kimyasal ileticinin sinapslar üzerinden boşalmasına yol açar. Dendritler bu kimyasal mesajı alıp çekirdeğe iletecek biçimde yapılanmıştır. Bir nöronun dendritlerine gelen sinyaller entegre olarak veya beraber toplanarak bileşik sinyali oluştururlar. Bu bilgiler, eğer gelen uyarı belirli bir eşik değerini aşarsa, diğer sinir hücresine geçirilir. Bu durumda sinir hücresi uyarılmıştır denir. Eğer gelen uyarı çok düşükse bilgi daha ileri taşınmaz. Bu durumda ise sinir pasif bırakılmış olur. İki sinir arasındaki bağlantılar birbiri ile uyumludur. Yani bağlantı yapısı büyük çapta değişebilir. Beynin öğrenme yeteneği de bu uyum sayesinde olmaktadır.

Yapay sinir ağlarında amaçlanan, beynin çalışma sistemine benzer çalışmak olduğundan, doğal olarak varolan teknolojik yetersizliklerden dolayı büyük çaplı basitleştirmelere gidilmiştir. Herşeyden önce biyolojik sinir hücrelerine ilişkin işlemlerin tam olarak kopyalanması mümkün değildir.

2.2. Yapay Sinir Ağlarının Temel İlkeleri

Yapay sinir ağlarının dayandığı temel ilkeler şöyle sıralanabilir.

- i. Nöron adı verilen ve bilgiyi işleyen bir çok birim içerirler.
 - ii. Nöronlar arasında kurulan bağlantı hatları aracılığıyla işaretlerin bir nörondan diğerine taşınması sağlanır.
 - iii. Nöronlar arasındaki her bir bağlantı hattının kendine özgü bir ağırlık (weight) değeri vardır. Tipik bir sinir ağında taşınan işaret ile bu ağırlık değerinin çarpımı yolu ile toplam enerji hesaplanır.
 - iv. Her bir nöronun yer aldığı ağın girişine uyguladığı, genelde doğrusal karakterde olmayan bir aktivasyon (uyarma) işlevi vardır ve çıkış işareti bu işlevden yararlanılarak bulunur.
- Sinir ağının karakteri ise;
- i. Nöronların bağlantı yolları (mimari yapı),
 - ii. Bağlantıya ilişkin olarak ağırlık (weight) değerlerinin hesaplanma yöntemi ki bu aynı zamanda ağın öğrenme algoritması demektir,
 - iii. Aktivasyon işlevi ile belirlenir [1].



Şekil 2.2 - YSA'nın tarihsel gelişimi

2.3. Yapay Sinir Ağlarının Tarihsel Gelişimi

Yapay sinir ağlarının tarihsel gelişimi Şekil 2.2’de gösterilmiştir. Yapay sinir ağları kavramı ilk olarak 1940lı yılların ortalarında oluşturulmaya başlandı.

McCulloch ve Pitts, 1943 yılında basit yapılı lojik fonksiyonlarla ilk yapay sinir ağını kurdular. Daha sonra Hebb, 1949 yılında kendi adıyla anılan öğrenme kuralını yapay sinir ağlarına uyguladı. 1954 yılında Minsky (RL - Reinforcement Learning) Takviyeli Öğrenme Kuralı’nı oluşturdu. 1957 yılında Bellman, Dinamik Programlama’yı (DP - Dynamic Programming) literatüre tanıttı. Bir yıl sonra da Frank Rosenblatt, yapay sinir ağlarının en çok kullanılan terimi olan “perseptron”u literatüre tanıttı.

Daha sonra (LMS - Least Mean Square) En Küçük Ortalama Hata Karesi adı verilen yöntem, diğer bir adıyla Gradyan Azalması Metodu, Widrow ve Hoff tarafından geliştirildi. Bu algoritma basit lineer nörona uygulanarak (adaptive linear neuron) Adaline algoritması geliştirildi.

1974 yılında Werbos, Geriye Yayılma (BP-Backpropagation) Algoritması’nı ortaya attı. Willshaw ve Malsburg 1976 yılında Özdenetimli Harita (SOM - Self Organizing Feature Map) adıyla kümeleme algoritmasını kurdular. 1982 yılında Hopfield kendi adıyla anılan Hopfield Ağı’nı geliştirdi. 1982 yılında Kohonen, özdenetimli harita ağı kuramını beyindeki oluşumların karşılaştırmalı haritasını çıkartabilmek amacıyla ortaya koydu. Kohonen’in oluşturduğu ağ, Willshaw ve Malsburg’un oluşturduğu SOM ağına çok benzer niteliktedir. Aynı yıl Parker, BP, Geriye Yayılım Algoritması’nı tartışmaya açtı. Genelleştirilmiş Delta Öğrenme Kuralı olarak da bilinen BP algoritması, 1986 yılında Rumelhart ve McClelland tarafından tekrar gündeme getirilip popüler yapıldı.

1980’li yılların ortalarında termodinamik modellerin gelişimi gözlendi. Smolensky’nin 1986’daki Uyum Teorisi (Harmony Theory) ile Boltzman Makinası bu modellere birer örnektir. Ackley Hinton ve Sejnowski 1985 yılında Boltzman Makinası’nı kullanarak bir öğrenme kuralı geliştirdiler.

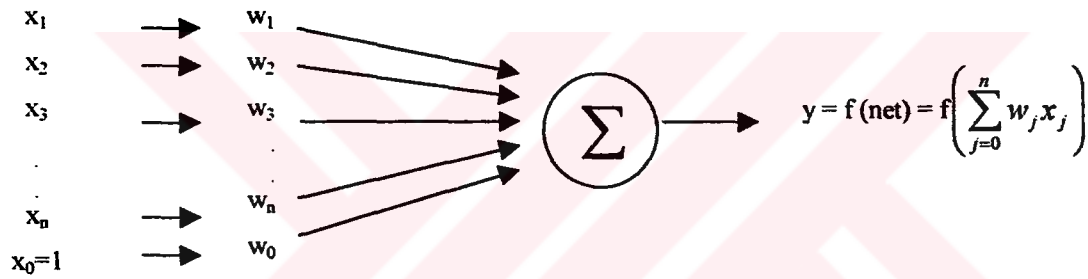
1986 yılında Rumelhart ve McClelland’ın yazdıkları kitap yayınlandıktan sonra yapay sinir ağlarına ilgi birden arttı. Ertesi yıl IEEE ilk yapay sinir ağları konferansını düzenledi. Uluslararası Yapay Sinir Ağları Kuruluşu (International

Neural Network Society - INNS) da aynı yıl faaliyete geçerek 1988 yılında ilk bildirisini yayınladı.

1988 yılında Broomhead ve Lowe, Radyal Bazlı İşlev Ağı (Radial Basis Function Network) adıyla yeni bir sinir ağı yapısı geliştirdiler. Ağ mimarilerinin değişik kombinasyonları olarak ileri beslemeli ve geri beslemeli yapılar tanıtıldı (Almeidal 1987, Pineda 1989, Williams ve Zipser 1989) [2].

2.4. Basit Nöron Yapısı

En basit şekli ile bir nöronun giriş ve çıkışları dikkate alınarak elde edilen matematik modeli Şekil 2.3'deki gibidir.



Şekil 2.3 – Basit nöron yapısı

Nörona uygulanan her bir giriş değeri $x_1, x_2, x_3, \dots, x_n$ kendisi ile ilgili ağırlık $w_1, w_2, w_3, \dots, w_n$ değerleriyle çarpılır. Bu ikili çarpımın toplamı;

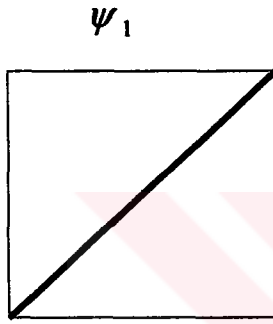
$$net(x, w) = x_0 w_0 + x_1 w_1 + x_2 w_2 + x_3 w_3 + \dots + x_n w_n = w_0 + \sum_{j=1}^n x_j w_j \quad (2.1)$$

şeklinde ifade edilir. Buradaki w_0 ağırlık değeri, $x_0=1$ sabit girişine karşılık gelen referans (bias) değeridir, θ işareti ile de gösterilir. Ağırlıklı toplam, yani $net(x, w)$ değerinin hesabından sonra bu değer bir aktivasyon işlevi üzerinden geçirilmesi ile nöronun çıkışı üretilir. Nöronun aktivasyon işlevi ψ ile gösterilecektir.

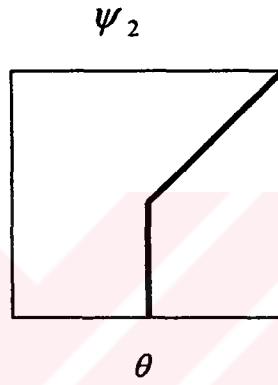
2.5. Aktivasyon (Uyarma) İşlevi

Aktivasyon işlevi, çıkış değerlerinin aşağı yukarı dalgalanmasını sınırlar, çıktı işaretlerinin taban-tavan sınırları arasındaki dalgalanmasını kabul edilebilir sonlu bir değere sınırlar. Bir nöron çıkışının normalleştirilmiş dalgalanma aralığı $[0, 1]$ aralığı veya alternatif olarak $[-1, 1]$ aralığıdır.

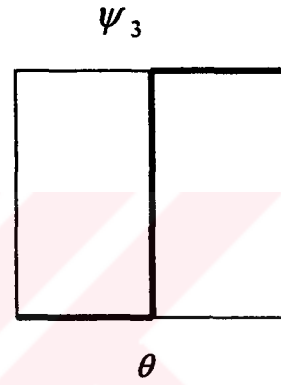
Şekil 2.4.a'da görülen şekle karşılık gelen işlev, giriş ve çıkış değerleri arasındaki lineer ilişkiyi göstermektedir. Bu tip aktivasyon işlevine sahip ağlar nispeten basit hesaplama durumlarında kullanılırlar.



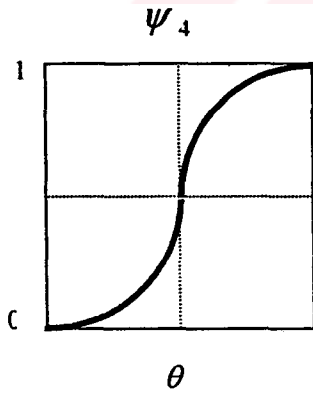
Şekil 2.4.a - Lineer aktivasyon işlevi



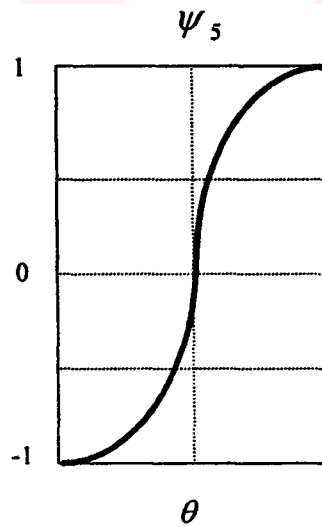
Şekil 2.4.b - Lineer eşik aktivasyon işlevi



Şekil 2.4.c - Basamak aktivasyon işlevi



Şekil 2.4.d – Lojistik sigmoid aktivasyon işlevi



Şekil 2.4.e – Hiperbolik tanjant tipi aktivasyon işlevi

Şekil 2.4.b'de görülen durum lineer eşik aktivasyon işlevine karşılık gelir. İşlev, θ eşik değerinden düşük olan değerler için sıfır değerini vermektedir. Bu değer üzerinde olan değerler için de sınırlanmamış lineer bir işlev olarak davranış gösterir. Şekil 2.4.c, basamak işlevi veya eşik mantık birimi olarak adlandırılan aktivasyon durumunu göstermektedir. Buradaki fonksiyonel ilişki eşik değeri altında sıfır, üzerinde ise bir olacak şekilde kurulmuştur. İşlevin matematiksel ifadesi aşağıdaki gibidir.

$$\begin{aligned} y &= f(net(x, w)) \\ net(x, w) \geq 0 \text{ ise } y &= 1 \\ net(x, w) < 0 \text{ ise } y &= 0 \end{aligned} \quad (2.2)$$

Şekil 2.4.d lojistik sigmoid işlevi olarak bilinmektedir. Bu işlev önceki işlevlerin birçok özelliğine sahiptir. Sigmoid işlevinin matematiksel ifadesi;

$$y = \frac{1}{1 + e^{-\beta[net(x, w)]}} \quad (2.3)$$

olarak yazılır. β sabiti yarı lineer bölgenin eğimini kontrol eden bir parametredir. Küçük β değerleri için sigmoid işlevinin davranışı lineer aktivasyon işlevi gibi olurken, büyük β değerleri için sigmoid işlevinin davranışı da basamak işlevi gibi olacaktır.

Şekil 2.4.e ise hiperbolik tanjant tipi aktivasyon işlevine aittir. Hiperbolik tanjant işlevinin matematiksel ifadesi;

$$y = \frac{e^{net(x, w)} - e^{-net(x, w)}}{e^{net(x, w)} + e^{-net(x, w)}} = \tanh[net(x, w)] \quad (2.4)$$

olarak verilir.

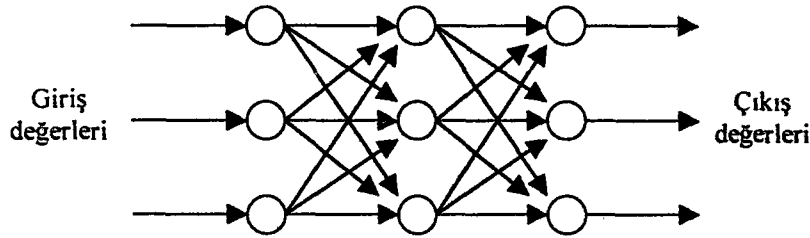
Aktivasyon işlevi seçiminde iki koşul göz önüne alınır. Bu koşullar türevi alınabilirlik ve nonlinearliktir. Genellikle türevi alınabilir aktivasyon işlevleri seçiminin nedeni yapay sinir ağlarında oluşturulan dönüşüm (mapping) işleminin analitik kontrolünü kolaylaştırmaktır. Özellikle gradyan azalması yöntemine göre öğrenmeyi gerçekleştiren algoritmalarda ağ aktivitesinin ağ ağırlıklarına göre

gradyan hesaplaması yapılmaktadır. Lineer eşik işlevi ve basamak işlevleri bu özelliği taşımazlar. Lojistik sigmoid işlevi ve hiperbolik tanjant tipi aktivasyon işlevleri hem nonlinearlik hem de türevi alınabilirlik özelliklerine sahip olduklarından daha çok tercih edilen işlevlerdir. Bu iki işlev arasından birini tercih etmek ise tamamen keyfi bir seçimdir. Bu işlevler arasında doğrusal bir dönüşüm yapmak mümkün olduğundan ağ eğitiminde bu işlevlerin birbirlerinden çok farklı sonuçlar çıkarmaları beklenmemelidir [3].

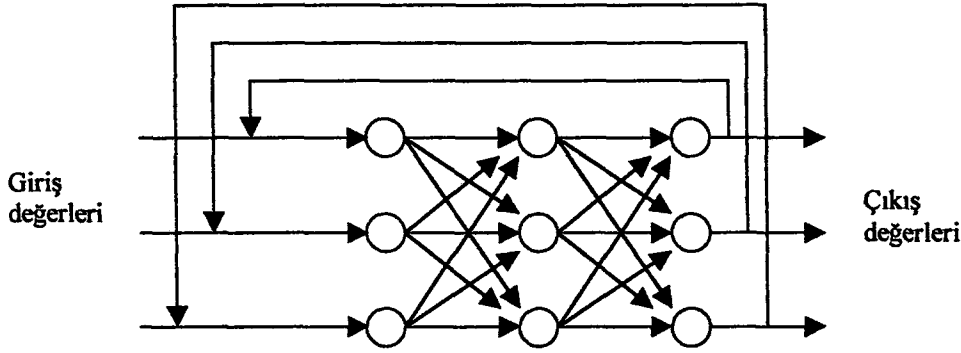
Nonlinear aktivasyon işlevleri seçilerek oluşturulan bir yapay sinir ağının, lineer aktivasyon işlevleri seçilerek oluşturulmuş bir modele göre temel avantajı daha karmaşık dönüşümlerin gerçekleştirilebilmesidir [4].

2.6. Yapay Sinir Ağı Yapıları

Dodd, 1992 yılında ağ yapısının probleme uygun olarak yapılandırılmaması durumunda öğrenmenin çok zor hatta imkansız olacağını açıklamıştır. Ağ yapısı, basit işlem birimleri ve bunların birbirleriyle ilişkilendirilmelerinden oluşur [2]. Temelde iki tip ağ yapısı bulunmaktadır. Bunlar ileri beslemeli yinelemesiz sinir ağı (feedforward neural network-FNN) ve geri beslemeli yinelemeli (recurrent neural network-RNN) sinir ağıdır. Giriş birimlerinden çıkış birimlerine doğru işaretlerin tek yönlü yayıldığı ağ ileri beslemeli olarak bilinirken, herhangi bir nöronun çıktısının tekrar o nörona girdi olarak gönderildiği ağ da geri beslemeli olarak bilinir. Şekil 2.5 ve Şekil 2.6'da ileri beslemeli ve geri beslemeli ağ yapıları gösterilmiştir. Her iki ağ yapısı da tek katmanlı veya çok katmanlı türde olabilir. Geri beslemeli ağ yapıları genellikle optimizasyon problemlerinin çözümünde kullanılırken, ileri beslemeli ağ yapıları işaret işleme, konuşma algılama, istatistiksel öngörme, veri sıkıştırma ve adaptif işlem kontrolü alanlarında kullanılır.



Şekil 2.5 – İleri beslemeli yapay sinir ağı yapısı



Şekil 2.6 – Geri beslemeli yapay sinir ağı yapısı

2.6.1. Tek katmanlı ve çok katmanlı sinir ağları

Tek katmanlı sinir ağında (single layer neural network - SNN) saklı katman (hidden layer) bulunmaz. Sadece giriş katmanı ve çıkış katmanından oluşur. Çok katmanlı sinir ağında ise (multilayer neural network - MNN) en azından bir tane saklı katman vardır. Tek katmanlı sinir ağlarının kapasitelerini zorlayan hesaplama işlevlerini çok katmanlı sinir ağları yapabilmektedir.

İleri beslemeli ve geri beslemeli sinir ağları tek katmanlı veya çok katmanlı yapılarda olabilirler. Örneğin; Radial Basis Function Network (RBFN), B-Spline Ağı, Fonksiyonel Bağlantı Ağı (Functional Link Network - FLN), Adaline ve perseptron ileri beslemeli tek katmanlı yapay sinir ağlarıdır. Çok katmanlı perseptron ise ileri beslemeli çok katmanlı yapay sinir ağıdır. Boltzman Makinası ve BAM (Bidirectional Associative Memories) geri beslemeli çok katmanlı sinir ağları iken ART (Adaptive Resonance Theory), Kohonen ve Hopfield ağları geri beslemeli tek katmanlı ağlardır.

Çok katmanlı yapay sinir ağlarında istenen sonuca ulaşabilmek için saklı katman sayısının optimum düzeyde belirlenmesi gerekir. Her uygulama için öngörülecek bu optimum değer değişkenlik gösterir. Saklı katman sayısının bir seçilmesi ve saklı katmanda uygun sayıda nöron kullanılması ile her nonlineer işlevin çözümünün bulunabileceği savunulmaktadır (Funahashi, 1989; Cybenko, 1989; Hornik, 1989; Funahashi ve Nakamura, 1993) [2]. Bazı problem çözümlerinde ise bir yerine iki tane saklı katman kullanılması daha iyi sonuç doğurmaktadır (Sontag, 1990).

DeVillers ve Barnard (1993) iki saklı katmanlı yapay sinir ağlarında tek saklı katmanlı sinir ağlarına oranla yerel minimuma takılma problemine daha çok rastlandığını belirtmişlerdir. Gizli katmandaki veya katmanlardaki nöron sayısının optimal sayısını veren herhangi bir analitik yöntem şu ana dek geliştirilememiştir. Dolayısıyla bu katmanlardaki nöron sayısı belirsizliğini aşabilmenin tek yolu deneme yanılma yöntemidir. Saklı katmanda düğüm sayısının gereğinden az seçilmesi durumunda ağ, öğrenme işlemi tamamlayamayabilir. Düğüm sayısının gereğinden fazla seçilmesi durumunda ise ağ giriş verilerini ezberleme işlevi kazanır ve yapılması istenen hesaplama çok uzun ve pahalı bir sürece girer [22]. Ezberleme, eğitim kümesi dışında oluşturulacak olan sına kümeleri için giriş ve çıkışlar arasında gerekli ilişkinin kurulamadığı, başarımın yalnızca eğitim kümesinde kullanılan giriş çıkışlarla sınırlı kaldığı durumdur [8].

2.7. Yapay Sinir Ağlarında Öğrenme Algoritmaları

Öğrenme algoritmaları üç ana başlıkta toplanabilir.

- Eğitici Öğrenme (Supervised Learning)
- Takviyeli Öğrenme (Reinforcement Learning) veya Derecelendirmeli Öğrenme
- Eğitici-siz, Kendi Kendine Öğrenme (Unsupervised Learning)

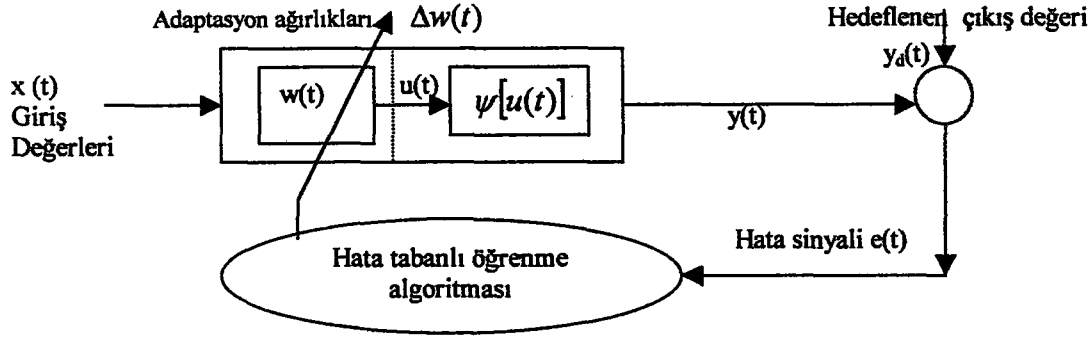
Eğitici öğrenme algoritmasında sisteme verilen girdilere karşılık istenen belirli çıkış değerleri kullanılarak ağı eğitimi sağlanır. Hata tabanlı öğrenme algoritmaları olarak da tanımlanabilen eğitici öğrenme algoritmaları, bir dışsal sinyal (öğretmen) alırlar ve alınan cevapla elde bulunan referans değerini karşılaştırarak bir hata sinyali üretirler. Bu hata sinyaline dayanarak yapay sinir ağları, sistemin işleyişini daha düzenli hale getirmek için sinaptik bağlantılarını düzenlerler. Sinaptik bağlantılar w , giriş değerleri x , ile gösterilecek olunursa, hata tabanlı öğrenme algoritmaları için eşitlik;

$$w(t+1) = w_i(t) + \Delta w_i(t) \quad (2.5)$$

$$\Delta w_i(t) = \eta_i x_i(t) [y_d(t) - y(t)] \quad (2.6)$$

olacaktır. Burada $w_i(t)$, $x_i(t)$ giriş değerlerine karşılık olan ağırlık değerleridir. $\Delta w_i(t)$ parametresi ise belirli bir zaman aralığındaki $w_i(t)$ bağlantı ağırlığındaki

değişmedir. η , öğrenme katsayısıdır. $y_d(t)$ ise sinirden beklenen çıktı, $y(t)$ ise sinirin kendi bulduğu çıktı değeridir. Eğitici öğrenme prosedürü sistematik olarak Şekil 2.7’de gösterilmiştir [5].



Şekil 2.7 – Eğitici öğrenme algoritması şeması

Eğitici öğrenmenin özel bir hali olan takviyeli öğrenme algoritmasında istenen belirli giriş ve çıkış değerleri bulunmamakta, eğitim belirlenen ölçütlere göre ağ çıkışına “iyi” veya “kötü” ölçütleri verilerek gerçekleştirilmektedir. Özellikle kontrol uygulamalarında kullanılır.

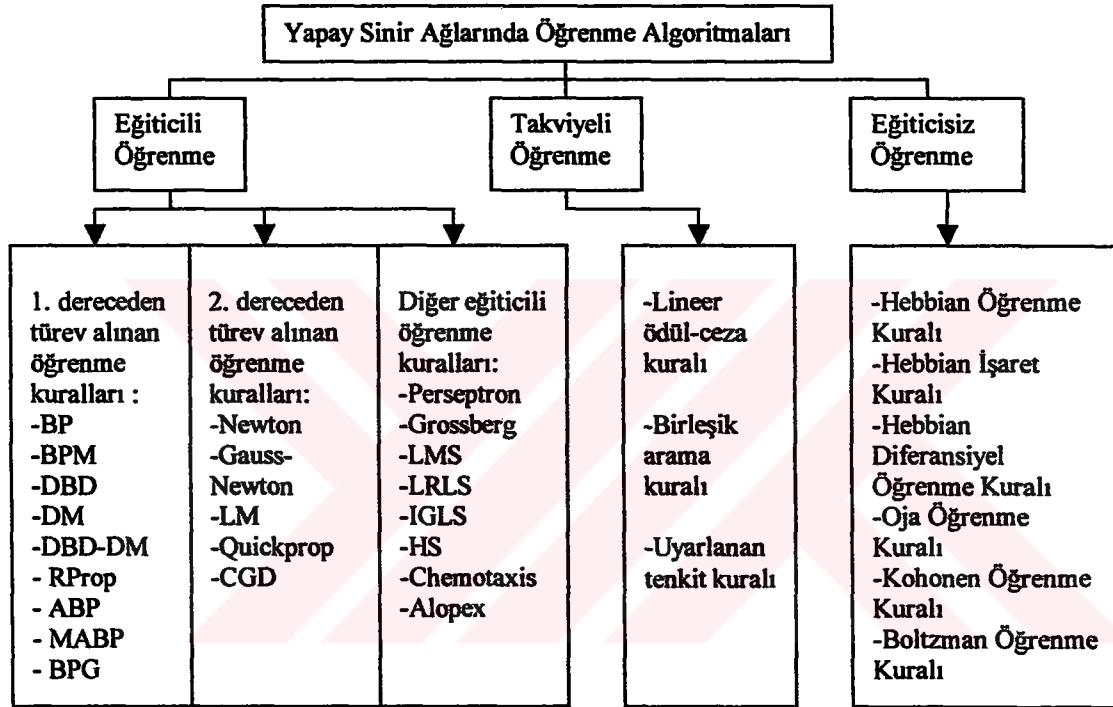
Eğitici öğrenme, özdenetimli öğrenme olarak da tanımlanabilir. Eğitici öğrenme algoritmaları kendi yerel bilgi ve iç kontrol mekanizmalarına dayanan öz denetimli organizasyon ilkelerini içerirler. Bu tip öğrenmede mevcut bir hedef çıktı değeri yoktur. Burada amaç, birbirlerine benzer birimleri belirli değer aralıkları içinde gruplamaktır. Bu tür algoritmalar olasılık yoğunluk işlevlerine, sınıflandırma ve şekil tanıma problemlerine uygulanabilirler.

Eğitici öğrenme algoritması kullanılan bir yapay sinir ağı, öğretmenin sorduğu soruyu cevaplayan ve cevabını öğretmenin kendisine söylediği doğru cevap ile karşılaştıran öğrenciye benzer. Eğitici öğrenme algoritması kullanılan bir yapay sinir ağı, yanında öğretmen yokken konuyu öğretmenin hazırladığı video kasedinden izleyerek öğrenen öğrenciye benzer. Öğretmen, yöntemi, öğrenme kurallarını ve soruları yani giriş vektörlerini hazırlamakta fakat soruların yanıtını yani çıkış vektörlerini vermemektedir. Takviyeli öğrenme algoritması kullanımı hali ise öğretmenin, öğrenciye uygulanan sınavda ayrı ayrı her bir sorudaki başarısına değinmeden genel olarak iyi veya kötü not aldığını söylemesine benzer. Böylece

öğrenci hangi soruda hata yaptığını veya hangi soruyu doğru cevapladığını bilemeyeceğinden performansını geliştirmesi mümkün değildir [2].

Eğitme sırasında eğitme verisinin ağıdan her geçişine öğrenme devri (epoch) denir. Yapay sinir ağları, bir çok öğrenme devrinde ağırlık değişimi ile eğitilir. Eğitme işlemine, giriş örnekleri ve ağırlık değiştirme işleminden yeterli sonuç alınmıncaya kadar devam edilir [6].

Tablo 2.1 – Yapay sinir ağlarında kullanılan öğrenme algoritmalarının sınıflandırılması

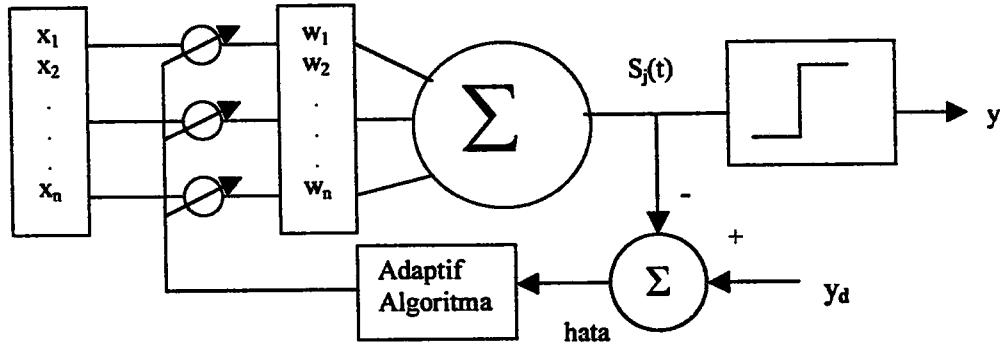


Tablo 2.1’de gösterilen eğitici öğrenme algoritmaları özelliklerine göre incelenecektir.

2.7.1. Eğitici öğrenme algoritmaları

2.7.1.1. Perseptron öğrenme kuralı

Perseptron ilk olarak 1958 yılında F. Rosenblatt tarafından bulunmuştur. Perseptron, ikili veriyi girdi olarak kabul eden ve buna göre sonuç üreten çok basit bir sinir ağıdır. Bu sinir ağı (AND) VE, (OR) VEYA gibi temel mantık işlemcilerini çözebilir. Daha karmaşık mantıksal operasyonlar (XOR problemleri) bu sinir ağı ile çözülememektedir. Şekil 2.8’de perseptron yapısı gösterilmektedir. Şekil 2.8’deki aktivasyon işlevi Şekil 2.4.c’de gösterilen basamak işlevi olarak alınmıştır [14].



Şekil 2.8 – Perseptron yapısı

Rosenblatt'ın tek katmanlı perseptronu şu adımlarla kendini eğitir:

- i. Ağdaki ağırlık değerleri rastgele seçilir.
- ii. Giriş değerleri (x) verilerek

$$S_j(t) = \text{net}(x(t), w(t)) - w_0(t) = x_1(t)w_1(t) + \dots + x_n(t)w_n(t) = \sum_{j=1}^n x_j(t)w_j(t) \quad (2.7)$$

toplamı bulunur.

- iii. Her birimin çıktısı

$$y(t) = \begin{cases} 1 & S_j(t) > Th \\ 0 & S_j(t) < Th \end{cases}$$

Th = eşik değeri (threshold value)

olarak belirlenir. Aktivasyon işlevi olarak basamak işlevi kullanılır.

- iv. Hata değeri hesaplanır.

$$e_j(t) = y_d(t) - y(t) \quad (2.8)$$

y_d : hedeflenen çıkış değeri

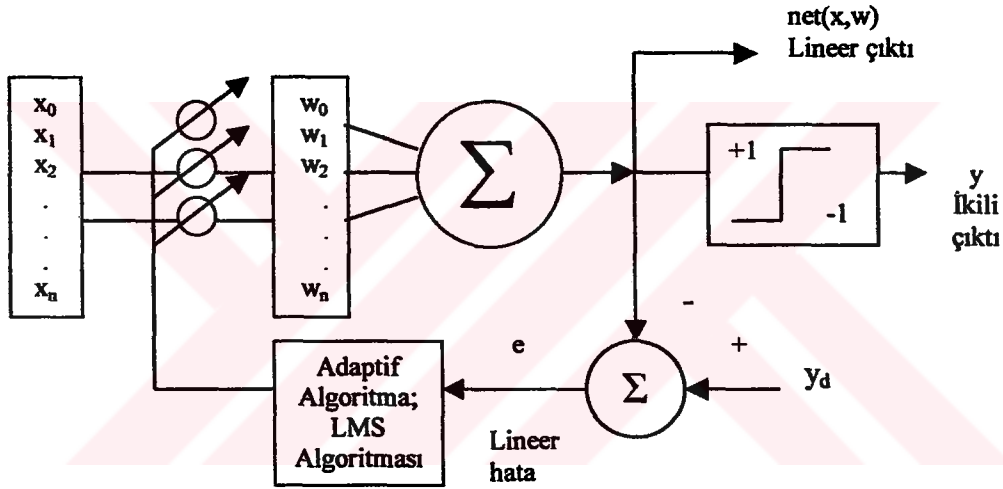
- v. Ağırlık değerleri yenilenir. η : öğrenme katsayısı

$$w_j(t+1) = w_j(t) + \eta x_j(t) e_j(t)$$

Öğrenme katsayısı, giriş dağılımındaki değişiklikleri hızlı bir şekilde adapte edebilecek biçimde büyük, önceki girişlerdeki ortalama eğilimi saklayacak ve kararlı ağırlık ayarlamaları yapacak ölçüde de küçük seçilmelidir. Buradaki birbirine karşıt iki istek optimal biçimde dengelenmelidir.

vi. Hata değeri istenen seviyeye düşene dek adım ii – iv arası tekrarlanır.

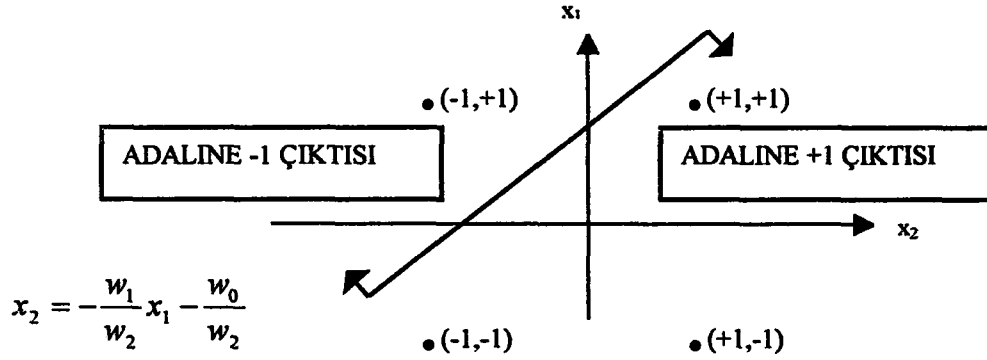
Rosenblatt perseptronunun, Adaline (adaptive linear element) biriminden farkı; giriş işareti olarak $\{1, 0\}$ ikili girdi değerlerini alması ve eşik ağırlık değerini $\{w_0\}$ işleme katmamasıdır. Yapay sinir ağlarının en basit elemanı olan Adaline yapısı, Şekil 2.9'da gösterilmiştir [26].



Şekil 2.9 - Adaline yapısı

Şekil 2.9'da gösterilen Adaline yapısı, n adet ikili (binary) giriş değerine karşılık bir adet ikili çıkış verir. Giriş vektörü olarak 2^n tane olası veri bulunmaktadır. Hedeflenen çıkış değerine göre, lojik işlemci, her veriyi $+1$ veya -1 çıktısı olarak sınıflayacaktır. N adet giriş değerini ikili veride tek bir çıktıya bağlayan 2^{2^n} olası lojik fonksiyon vardır. Adaline bu fonksiyonlardan lineer eşik işlevini ve basamak işlevini algılayabilecek yapıdadır.

İki giriş vektörlü bir adaline elementi göz önüne alınırsa; olası giriş değerleri Şekil 2.10'da gösterildiği gibi olacaktır. Koordinat eksenleri giriş vektörleri bileşenlerini göstermek üzere, Adaline, giriş değerlerini, ağırlık birimlerine göre iki kategoriye ayırır.



Şekil 2.10 – Çözüm uzayı

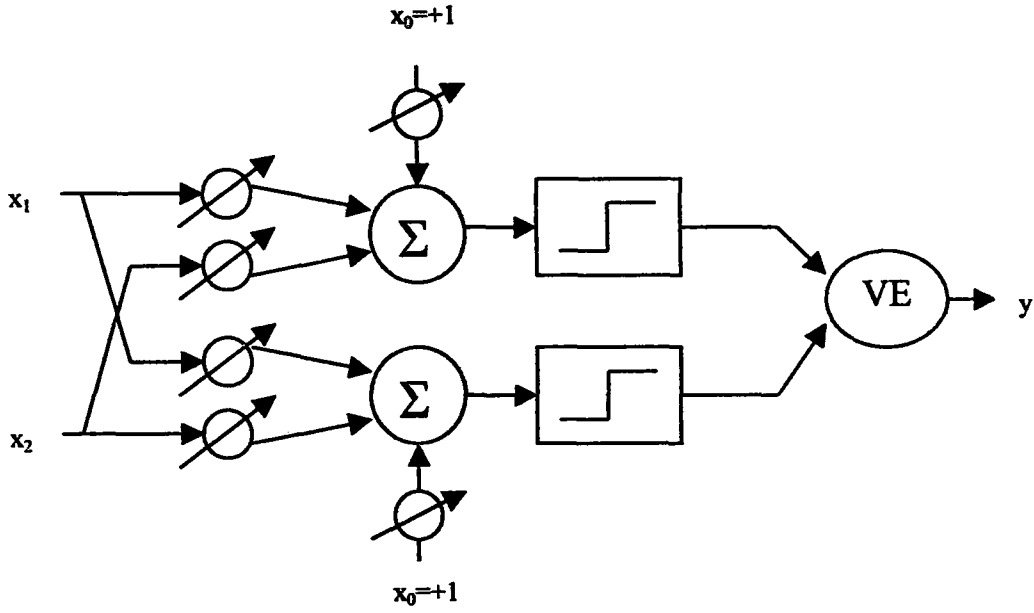
$net(x, w)$ lineer çıktı değerini sifra eşitlersek;

$$net(x, w) = x_1 w_1 + x_2 w_2 + w_0 = 0$$

$$x_2 = -\frac{w_1}{w_2} x_1 - \frac{w_0}{w_2}$$

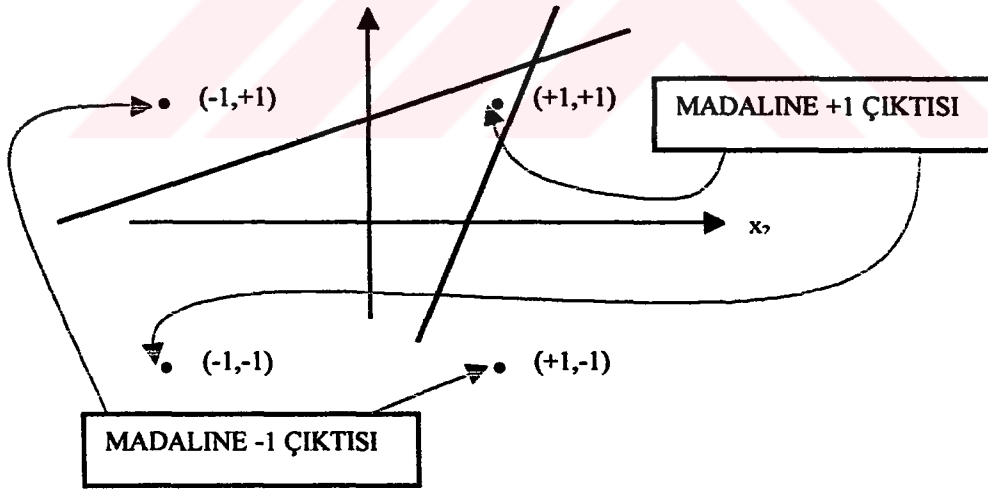
olacaktır. Şekil 2.10'da bu lineer ilişki gözlenmektedir. Dört ağırlık değeri kullanılan Adaline biriminde çözüm uzayı bir düzlem belirtirken, daha fazla ağırlık değeri kullanılması durumunda ise hiperdüzlem belirtecektir. Ağırlık değeri w_0 'ın sıfır değerine eşit olması durumunda, hiperdüzlem homojen olacak ve çözüm uzayının ayırıcı çizgisi orijinden geçecektir [26].

Widrow ve Hoff'un geliştirdikleri Madaline-I yapısı, birden çok adaptif eleman içeren sinir ağı modellerindendir. Ridgway, Hoff ve Glanz, Madaline-I algoritmasının matematiksel analizini beraber geliştirmişlerdir. Madaline-I algoritmasında giriş katmanına adaptif adaline birimleri verilerek bu birimler mantıksal işlemci üzerinden geçirilir. Mantıksal işlemci sinir ağı çıkış değerini üretir. Madaline yapısı Şekil 2.11'de gösterilmiştir.



Şekil 2.11 – 2 Adaline birimi girişi olan Madaline Yapısı

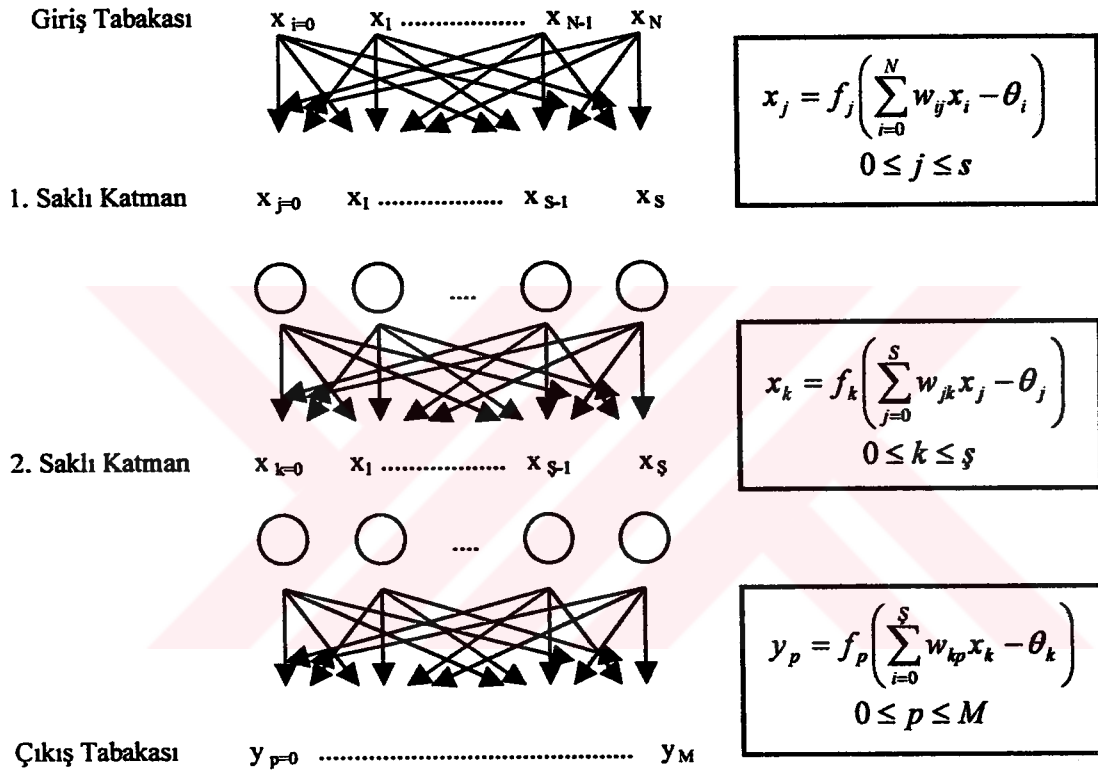
Uygun ağırlık değerleri seçilmesi ile Şekil 2.11’de gösterilen yapının çözüm uzayı, lineer ayrımsallığı olmayan NAND_{x_1} işlevi ile Şekil 2.12’de gösterildiği gibi ayrılacaktır [26].



Şekil 2.12 – Madaline yapısı çözüm uzayı

Çok katmanlı perseptronlar, giriş ve çıkış arasında, bir ya da daha fazla katmanın bulunduğu ileri beslemeli sinir ağlarıdır. Çok katmanlı perseptronlar, tek katmanlı perseptronun pek çok kısıtlamasını ortadan kaldırmıştır. Çok katmanlı perseptronların gücü aslında düğümlerde kullanılan aktivasyon işlevlerine bağlıdır. Şekil 2.13’de çok katmanlı bir perseptron yapısı gösterilmektedir. Çok katmanlı perseptron ağında, N sayıda giriş bileşeni, M sayıda da çıkış bileşeni bulunmaktadır.

Aktivasyon işlevinin seçimi için bir kısıtlama getirilmemiştir. Şekil 2.13’de iki saklı katmanı bulunan bir sinir ağı incelenmektedir. Birinci ve ikinci saklı tabaka çıkış değerleri x_j ve x_k olarak gösterilir. θ_i , θ_j ve θ_k sırasıyla birinci ve ikinci saklı tabakaların ve çıkış tabakasının referans (bias) değerleridir. Giriş katmanının bağlantı ağırlık değeri w_{ij} iken w_{jk} ve w_{kp} değerleri birinci ve ikinci saklı katmanların bağlantı ağırlık değerleridir.



Şekil 2.13 – Çok katmanlı perseptron

2.7.1.2. Widrow-Hoff öğrenme kuralı

En küçük ortalama hata karesi öğrenme kuralı

(LMS – Least Mean Square Error)

LMS yöntemi hata işlevi yüzeyinde negatif gradyan yönünde ilerlediği için “gradyan azalması” adını da alır. Perseptron öğrenme kuralına çok benzer yapıdadır. Aralarındaki farklar:

- Hata e , giriş ve ağırlık değerleri çarpımlarının toplamı olan $net(x, w)$ ile hedeflenen çıkış değeri y_d arasındaki farktır.

$$e_j(t) = y_d(t) - \text{net}(x, w) \quad (2.9)$$

- ii. $\text{net}(x, w)$ lineer toplamı, perseptron çıkış işlevi olan basamak işlevi yerine, lineer veya lineer eşik aktivasyon işlevinden geçirilir. Perseptron algoritması nonlineer iken LMS algoritmasının lineer olduğu Şekil 2.9'dan görülebilir.

Bu yöntemle göre, perseptron benzeri bir ağda, gerçekleşen ve hedeflenen çıkış arasındaki ortalama hata karesinin minimuma indirilmesi amaçtır. Ortalama hata karesi işlevi;

$$E(t) = \frac{1}{2} [y_d(t) - y(t)]^2$$

dir. Buradaki y_d hedeflenen çıkış değeridir. $E(t)$ 'nin minimizasyonu ağırlıklara göre gerçekleşmelidir. LMS yönteminde, hatanın ağırlıklara göre negatif gradyanı hesaplanıp, buna göre ağırlık adaptasyonları gerçekleştirilir.

$$y = \text{net}(x, w) = w_0 + \sum_{j=1}^n x_j w_j \quad (2.10)$$

$$-\frac{\partial E}{\partial w_j} = (y_d(t) - y(t)) \frac{\partial y(t)}{\partial w_j} = (y_d(t) - y(t)) x_j(t)$$

n , eğitim kümesindeki toplam örnek sayısını belirtir.

$$w_j(t+1) = w_j(t) + \eta x_j(t) e_j(t)$$

LMS öğrenme kuralı, delta kuralı olarak da bilinir. Eğer hata yüzeyi şekil olarak konveks bir yapıda ise global minimuma ulaşılır. Fakat, tek ve çok katmanlı ağlarda eğer aktivasyon işlevi lineer değilse hata yüzeyi konveks olmamakta ve ağırlık vektörü de lokal minimumlara yakalanmaktadır.

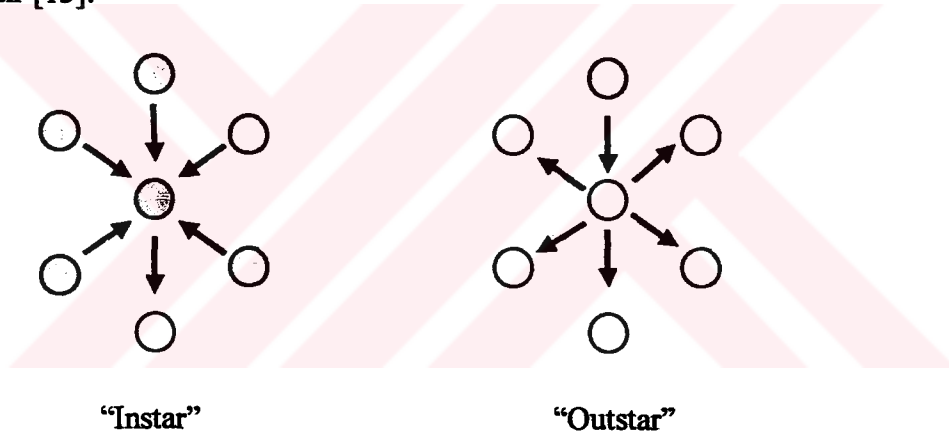
2.7.1.3. Grossberg öğrenme kuralı

Instar ve outstar eğitme kuralı olarak da bilinir. Wasserman tarafından 1989'da geliştirilmiştir.

$$w_i(t+1) = w_i(t) + \eta[x_i(t) - w_i(t)] \quad (2.11)$$

Denklem (2.11)'deki $x_i(t)$; ağ yapısının instar olması durumunda istenen giriş değeri, outstar olması durumunda ise istenen çıkış değeri olacaktır.

Her nörona, diğer nöronlardan sinapsları aracılığıyla çok sayıda işaret gelir. Bu işaretler nöronun giriş değerleri olarak adlandırılır. Nöronun modeli şematik olarak bir yıldıza (star) benzetilebilir. Yapay sinir ağlarında her nöron, diğer nöronlarca uyarıldığından bir "instar"dır. Aynı zamanda her nöron diğer nöronları uyardığından bir "outstar"dır [13].



Şekil 2.14 – Instar ve Outstar modeli

2.7.1.4.BP – (Backpropagation) Geriye yayılım algoritması

BP algoritması, zincir kuralı uyarınca sinir ağındaki her katmanda gradyan azalması tekniğine dayanır. Algoritma, hata işlevinin ağırlık değerlerine göre kısmi türevleri

$$\frac{\partial E(t)}{\partial w(t-1)}$$

üzerine kendini gerçektir.

Şekil 2.15'de sinir ağı kullanılan nonlinear bir hücre modeli görülmektedir. Sinir ağı

çıkışı y , hücre çıkışı y_d olarak tanımlanmıştır. Hata işlevi; $E(t) = \frac{1}{2}[y_d(t) - y(t)]^2$ dir.

Hata işlevinin, ayarlanabilen ağırlık değerlerine göre türevinin minimum olması hedeflenmektedir. Ağırlık değerleri değişimi denklem (2.12)'de gösterilmektedir.

$$e(t) = y_d(t) - y(t) ; \quad \eta = \text{öğrenme katsayısı}$$

$$w(t) = w(t-1) + \eta \left(\frac{\partial E(t)}{\partial w(t-1)} \right) \quad (2.12)$$

$$\frac{\partial E(t)}{\partial w(t-1)} = -e(t) \left(\frac{\partial y(t)}{\partial w(t-1)} \right)$$

Şekil 2.15'deki BP algoritmasının dezavantajları şöyle sıralanabilir;

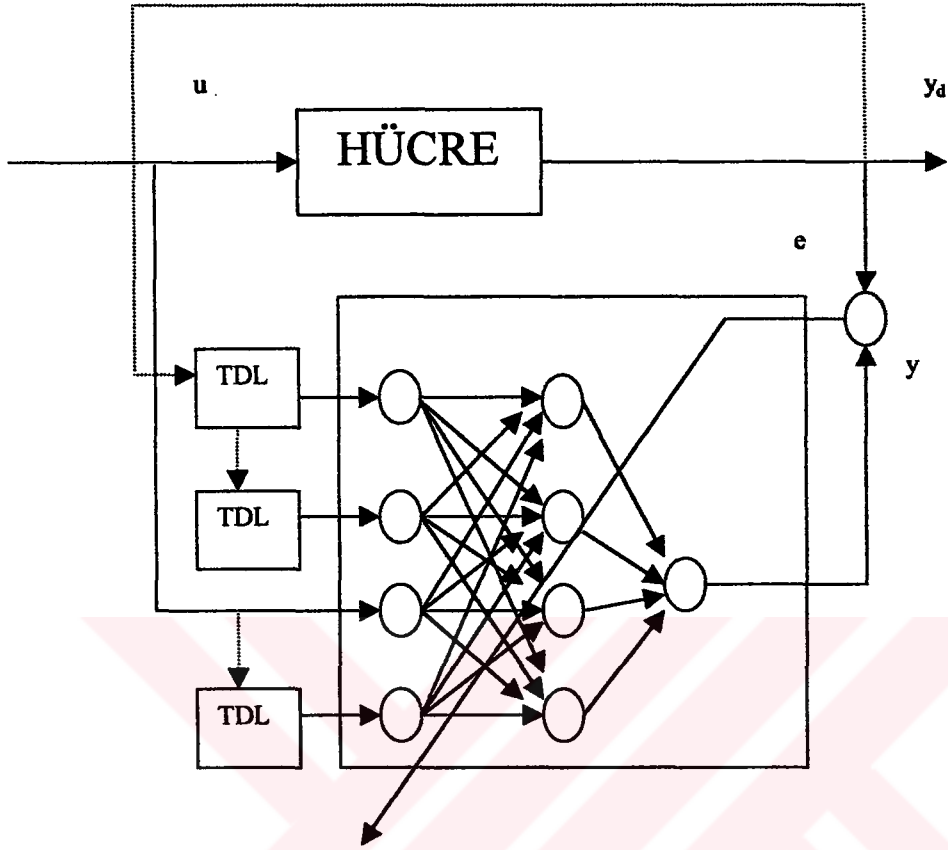
- i. Yakınsama hızı yavaştır.
- ii. Sistemin ilk koşullara duyarlılığı vardır.
- iii. Yerel minimuma takılma olasılığı vardır.
- iv. Büyük öğrenme katsayısı seçimi durumunda sistemde kararsızlık olasıdır.

Yukarıda sözü edilen dezavantajlarına rağmen Geriye Yayılım Algoritması kontrol problemlerinde sıkça kullanılmaktadır.

Genelde Hatanın Geriye Yayılımı Algoritması'nda (BP) öğrenme yavaştır. Bu, tipik olarak hata yüzeyinin karakteristiği sebebiyledir. Hatanın Geriye Yayılımı Algoritması'nı geliştirme amacıyla pekçok öneri ortaya atılmıştır. Bunların çoğu deneyime dayalı düzenlemeler olup, bazıları sistemin hızını arttırmaya yönelik, bazıları yerel minimuma takılıp kalmayı önlemeye yönelik, bazıları da sistemin genelleme yeteneğini arttırmaya yöneliktir [7, 9].

Hatanın Geriye Yayılması Algoritması, algoritmanın gradyen azalma tabanlı olması sebebiyle, ilk koşullara çok duyarlıdır. Eğer ilk atanan ağırlık vektörü, hata yüzeyinin minimumuna çok yakın ve eğimli bir bölgede ise yapay sinir ağının öğrenmesi çok çabuk olacaktır. Eğer, ilk atanan ağırlık vektörü hata yüzeyinin minimumuna uzak ve eğimsiz bir bölgede ise bu durumda sonuca ulaşmak çok zaman alabileceği gibi ulaşamayabilmek de.

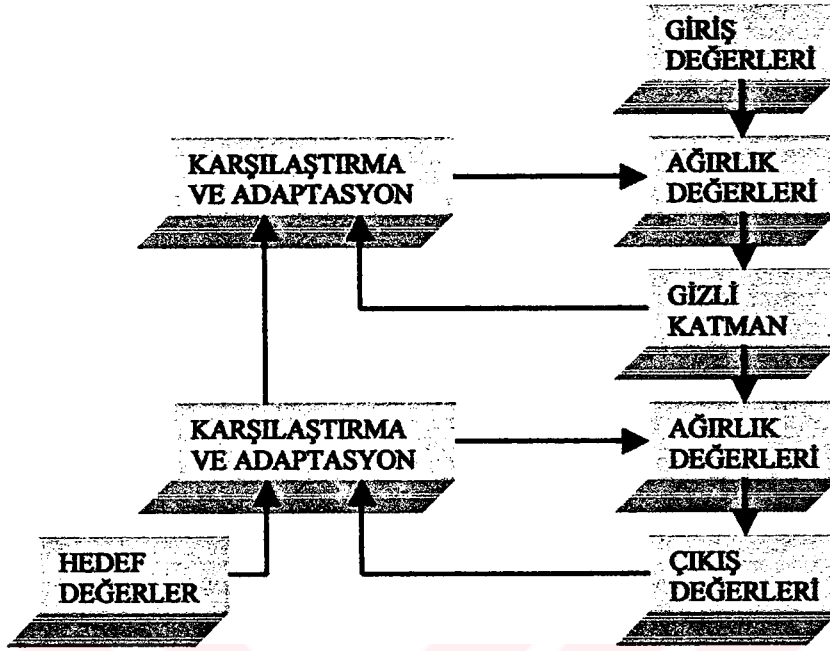
Uygulamada, ağırlık değerlerinin ilk atanmaları sıfıra yakın keyfi değerler şeklinde olur. Genellikle -1.0 ve 1.0 aralığında veya -0.5 ile 0.5 aralığında seçilirler [8]. Ağırlıkların büyük seçilmesi nöronların erken doymaya ulaşmalarına ve kendilerini öğrenme işlemine duyarsız hale getirmelerine neden olur [10].



Şekil 2.15 – Nonlineer hücre modeli

TDL : Tapped delay lines

Hatanın Geriye Yayılması Algoritması'nın blok akış şeması Şekil 2.16'da gösterilmektedir.



Şekil 2.16 - BP Algoritması blok akış şeması

Hatanın Geriye Yayılması Algoritması'nda çözüme ulaşma hızı ile öğrenme katsayısının değeri doğrudan ilgilidir. Eğer öğrenme katsayısı küçük seçilirse, yapay sinir ağının izleyeceği araştırma yolu gradyen yola çok yakın olacaktır. Bu durumda yerel minimuma ulaşmak için atılacak adım sayısı artacağından çözüme ulaşmak uzun zaman alacaktır. Diğer yandan öğrenme katsayısı çok büyük seçilirse, başlangıçta yapay sinir ağı çözüme kolay yaklaşacak fakat çözüm civarında osilasyon yaparak çözümü bulamayacaktır. Genelde minimumdan uzak bölgelerde büyük adım katetmek ve minimum civarında küçük adımlar kullanmak en uygun çözümdür [9]. Öğrenme katsayısının optimal değeri her uygulama için farklıdır. Genelde 0.1 ile 0.9 arasında değerler seçilir [8].

BP algoritmasına Bölüm 3'de geniş olarak yer verilecektir.

2.7.1.5. Momentum katsayılı geriye yayılım algoritması

BPM – Backpropagation with momentum

Rumelhart 1986 yılında BP algoritmasının ağırlık adaptasyon denklemine momentum katsayısını ekleyerek denklem (2.13)'ü oluşturmuştur.

$$w(t) = w(t-1) + \left(-\eta \frac{\partial E(t)}{\partial w(t-1)} \right) + \alpha \Delta w(t-1) \quad (2.13)$$

α , momentum katsayısıdır. Genelde $0 < \alpha < 1$ olarak seçilir. Momentum teriminin eklenmesi, yakınsama hızını arttırarak ağıın yerel minimuma takılması olasılığını azaltır. Denklem (2.14), denklem (2.13) geliştirilerek oluşturulmuştur (Nagata 1990). Denklem (2.14)'deki β katsayısı ağ tasarımcısının keyfi olarak seçeceği bir katsayıdır.

$$w(t) = w(t-1) + \left(-\eta \frac{\partial E(t)}{\partial w(t-1)} \right) + \alpha \Delta w(t-1) + \beta \Delta w(t-2) \quad (2.14)$$

Nagata 1990, β katsayısının ağıın yerel minimuma takılma olasılığını azalttığını savunmuştur.

2.7.1.6. Delta Bar Delta algoritması – DBD

DBD öğrenme kuralı, yerel optimizasyon yöntemine dayalı adaptif öğrenme hızını, ağıdaki yakınsamayı arttırmak için kullanır. DBD kuralında, gradyan azalması doğrultusu ile ağırlık uzayı yönünün çakışmasına gerek duyulmaz. (Jacobs, 1988)

Ardışıl iterasyonlar arasında ağırlık adaptasyonları zıt doğrultularda ise adım boyutu küçültülür, aksi halde adım boyutu büyütülür.

$w_{ij}(t)$ ağırlık değeri için öğrenme katsayısı $\eta_{ij}(t)$ olsun.

$$\eta_{ij}(t+1) = \eta_{ij}(t) + \Delta \eta_{ij}(t) \quad (2.15)$$

$$\Delta \eta_{ij}(t) = \begin{cases} \kappa & \bar{\varphi}_{ij}(t-1)\varphi_{ij}(t) > 0 \\ -\beta \eta_{ij}(t) & \bar{\varphi}_{ij}(t-1)\varphi_{ij}(t) < 0 \\ 0 & \text{aksi halde} \end{cases} \quad (2.16)$$

Denklem (2.16)'da geçen ifadeler;

$$\varphi_{ij}(t) = \frac{\partial E(t)}{\partial w_{ij}(t-1)} \quad ; \quad \bar{\varphi}_{ij}(t) = (1 - \varepsilon)\varphi_{ij}(t-1) + \varepsilon \bar{\varphi}_{ij}(t-1)$$

ε , pozitif sabit bir değerdir. $\kappa, \beta, \varepsilon$ değerleri ağ tasarımcısı tarafından belirlenir.

Jacobs (1988), DBD algoritmasına momentum katsayısı ekleyerek performansı arttırmıştır.

Benzetim sonuçları, DBD algoritmasının küçük parametre değişimlerine dahi çok duyarlı olduğunu göstermektedir. Özellikle κ katsayısının büyük seçilmesi ağda sapmalara yol açarken, küçük seçilmesi de öğrenme sürecinin uzamasına neden olmaktadır.

2.7.1.7. Decoupled momentum algoritması – DM

DM algoritmasında her w_{ij} ağırlık değerinin kendi adaptif öğrenme oranı η_{ij} , ve momentum katsayısı α_{ij} bulunmaktadır (Minai ve Williams, 1992). Öğrenme oranı, hata yüzeyine göre değişkenlik gösterir. Düzlem bölgelerinde hızlı artarken eğimli alanlarda yavaşlar. Momentum katsayısı da öğrenme oranı gibi değişir. Platolarda artarken, minimum yakınlığında eksponansiyel olarak azalır.

$$\eta_{ij}(t+1) = \eta_{ij}(t) + \Delta\eta_{ij}(t)$$

$$\Delta\eta_{ij}(t) = \begin{cases} \kappa_1 e^{-\gamma_1 |\bar{\varphi}_{ij}(t)|} & \bar{\varphi}_{ij}(t-1)\bar{\varphi}_{ij}(t) > 0 \\ -\beta_1 \eta_{ij}(t) & \bar{\varphi}_{ij}(t-1)\bar{\varphi}_{ij}(t) < 0 \\ 0 & \text{aksi halde} \end{cases} \quad (2.17)$$

$$\alpha_{ij}(t) = \min[\alpha_{\max}, \alpha_{ij}(t-1) + \Delta\alpha_{ij}(t-1)] \quad (2.18)$$

$$\Delta\alpha_{ij}(t) = \begin{cases} \kappa_m e^{-\gamma_m |\bar{\varphi}_{ij}(t)|} & \bar{\varphi}_{ij}(t-1)\bar{\varphi}_{ij}(t) > 0 \\ -\beta_m \alpha_{ij}(t) & \bar{\varphi}_{ij}(t-1)\bar{\varphi}_{ij}(t) < 0 \\ 0 & \text{aksi halde} \end{cases} \quad (2.19)$$

Yukarıda geçen ifadelerde;

$$\varphi_{ij}(t) = \frac{\partial E(t)}{\partial w_{ij}(t-1)} \quad ; \quad \bar{\varphi}_{ij}(t) = (1-\varepsilon)\varphi_{ij}(t-1) + \varepsilon \bar{\varphi}_{ij}(t-1)$$

olurken, $\varepsilon, \kappa_1, \beta_1, \gamma_1, \kappa_m, \beta_m, \gamma_m$ parametreleri ağ tasarımcısı tarafından belirlenir.

Öğrenme oranı η_{ij} lineer artış, eksponansiyel azalış gösterir.

Minia ve Williams 1992’de delta-bar-delta (DBD) algoritması ile decoupled momentum (DM) algoritmasını birleştirerek DBD-M algoritmasını kurmuşlardır. DBD-M algoritması, momentum katsayısı değişimleri için denklem (2.18) ve denklem (2.19)’u uygularken, öğrenme oranı değişimi için de denklem (2.16)’yı esas alır.

Minia ve Williams (1992), DBD-M ve DM algoritmalarının yakınsama hızı açısından BPM ve DBD algoritmalarından daha iyi olduklarını savunsalar da bu algoritmaların uygulanabilirliği, tasarımcının sisteme tanıtması gereken parametrelerin çokluğu ve algoritmanın uç bölgelerdeki aşırı hassasiyeti nedeniyle sınırlı kalmaktadır.

2.7.1.8. Rprop algoritması

Jervis ve Fitzgerald 1993’de RProp algoritmasını tasarladılar. RProp algoritması da DBD algoritmasında olduğu gibi her ağırlık değeri için farklı öğrenme katsayısı kullanır. Fakat ağırlık değeri w_{ij} yenilenirken, DBD algoritmasının aksine,

$\frac{\partial E(t)}{\partial w_{ij}(t-1)}$ türevinin işareti kullanılır.

$$\eta_{ij}(t) = \begin{cases} \min[1.2\eta_{ij}(t-1), \eta_{\max}] & \bar{\varphi}_{ij}(t-1)\varphi_{ij}(t) > 0 \\ \max[0.5\eta_{ij}(t-1), \eta_{\min}] & \bar{\varphi}_{ij}(t-1)\varphi_{ij}(t) < 0 \\ \eta_{ij}(t-1) & \text{aksi halde} \end{cases} \quad (2.20)$$

$\eta_{\max}$ ve $\eta_{\min}$ adım aralığını sınırlarlar.

$$\Delta w_{ij}(t) = \begin{cases} -\text{işaret}(\varphi_{ij})\eta_{ij}(t) & \bar{\varphi}_{ij}(t-1)\varphi_{ij}(t) > 0 \\ 0 & \text{aksi halde} \end{cases} \quad (2.21)$$

2.7.1.9. Adaptif geriye yayılım algoritması

ABP – Adaptive Backpropagation

ABP algoritması (Parlos, 1994) ağırlık değerlerini gradyan azalması doğrultusunda, hata ve hata türevinin işlevi olan bir öğrenme oranı ile yeniler. Ağırlık iterasyonları denklem (2.22)'deki gibidir.

$$w(\text{yeni}) = w(\text{eski}) + \rho(E) \frac{\varphi}{\|\varphi\|^2} \quad (2.22)$$
$$\rho(E) = \frac{\partial E}{\partial w}, \quad \rho(E) = \text{hata fonksiyonu}$$

$$\rho(E) = \begin{cases} \eta \\ \eta E \\ \eta \tanh\left(\frac{E}{E_0}\right) \end{cases} \quad (2.23)$$

E_0 ; ağırlık çıkış katmanındaki hata değeridir.

Parlos, ABP algoritmasının BP ve Quickprop algoritmasından daha hızlı olduğunu savunmuştur.

2.7.1.10. Değiştirilmiş adaptif geriye yayılım algoritması

MABP – Modified Adaptive Backpropagation

ABP algoritması biraz değiştirilerek MABP algoritması kurulmuştur. Ağırlık değerleri denklem (2.24)'deki gibi değişir.

$$w(t) = w(t-1) + \eta \frac{\varphi(t)}{\|\varphi(t)\|} \quad (2.24)$$

MABP'nin ABP'den farkı, η öğrenme oranının hata gradyeninin karesi yerine kendisine bölümünden ibarettir. Ayırık zaman aralıklarında MABP, ABP'den daha iyi performans göstermiştir.

2.7.1.11. Adaptif kazançlı geriye yayılım algoritması

BPG – Backpropagation with Adaptive Gains

BPG, Kruschke ve Movellan (1991) tarafından tasarlanmıştır. Algoritma, aktivasyon işlevine g kazanç değerini de eklemektedir.

$$y(t) = f(gS(t)) \quad (2.25)$$

$$w(t) = w(t-1) + \eta \left(-\frac{\partial E(t)}{\partial w(t-1)} \right) = w(t-1) + \eta g f'(Sg) \left(\frac{\partial S(t)}{\partial w(t-1)} \right) \quad (2.26)$$

g kazanç parametresi denklem (2.27) ile adaptasyon kazanır.

$$g(t) = g(t-1) + \eta \left(-\frac{\partial E(t)}{\partial g(t-1)} \right) \quad (2.27)$$

2.7.1.12. Newton'un ikinci dereceden türev alınan öğrenme algoritması

Newton Öğrenme Algoritması'nda hatanın (E), ağırlıklara göre ikinci derece türeviyle oluşan Hessian matrisi kullanılır. Hessian matrisi, ağırlık uzayının farklı doğrultularındaki gradyan değişimini gösterir. Hessian matrisi;

$$H(t) = \frac{\partial^2 E(t)}{\partial w(t-1)^2}$$

olarak tanımlanır. Newton'un algoritmasına göre ağırlık değerlerinin yenilenmesi denklem (2.28)'de gösterildiği gibidir (Cichocki ve Unbehauen, 1993) [2].

$$w(t) = w(t-1) + \eta [H(t)]^{-1} \frac{\partial E(t)}{\partial w(t-1)} \quad (2.28)$$

2.7.1.13. İkinci dereceden türev alınan Gauss-Newton öğrenme algoritması

Gauss-Newton Yöntemi'nde, Hessian matris yaklaşık $n \times n$ boyutunda, elemanları denklem (2.29)'daki gibi olan bir matris üretilir.

$$H(t) = \varphi^T(t)\varphi(t)$$

$$\varphi(t) = \frac{\partial e(t)}{\partial w(t-1)} \quad [\text{Ljung, 1987}] \quad (2.29)$$

2.7.1.14. Levenberg - Marquardt öğrenme algoritması

LM Öğrenme Algoritması'nda Gauss Newton yaklaşımı Hessian matrisine σ terimi eklenerek denklem (2.30) oluşturulur. Denklem (2.30)'daki σ terimi pozitif değerli küçük bir sayıdır, Marquardt parametresi olarak adlandırılır. I terimi ise birim matrisi gösterir.

$$H(t) = \varphi^T(t)\varphi(t) + \sigma I \quad (2.30)$$

Temel olarak Levenberg-Marquardt Algoritması (LMA), maksimum komşuluk fikri üzerine kurulmuş bir minimum ortalama hata karesi hesaplama metodudur. Bu algoritma Gauss-Newton ve Gradyan Azalması algoritmalarının en iyi özelliklerinden oluşur ve bu iki metodun kısıtlamalarını ortadan kaldırır. Genel olarak LMA yöntemi yavaş yakınsama probleminden etkilenmez. LMA'da hedef, minimum hata değerine karşılık düşen ağırlık değerini bulmaktır [17].

2.7.1.15. Quickprop öğrenme algoritması

Fahlman (1988), Quickprop Algoritması'nı, hata işlevi - ağırlık değerleri eğrisinin bir parabole yakınsanabileceği varsayımı üzerine kurmuştur. Quickprop Algoritması'nda Hessian matrisi köşegen matrisdir. Ağırlık adaptasyonu formülü denklem (2.31)'de verilmiştir.

$$\Delta w(t) = \frac{\varphi(t)}{\varphi(t-1) - \varphi(t)} \Delta w(t-1) \quad (2.31)$$

2.7.1.16. Birleşik gradyan azalması algoritması

CGD - Conjugate Gradient Descent

CGD Algoritması'nda, ağırlık uzayında, hata yüzeyi denklem (2.32)'yi sağlar.

$$E(t) = C^T w(t-1) + \frac{1}{2} w^T(t-1) H(t) w(t-1) \quad (2.32)$$

C terimi, sabit bir matris iken, H terimi Hessian matrisini ifade etmektedir. CGD Metodu ile hata yüzeyinde yerel minimuma ulaşmak hedeflendiğinden minimum arama doğrultusuna $v(t)$ imgesi atanarak denklem (2.33) kurulmuştur.

$$v(t) = -\frac{\partial E(t)}{\partial w(t-1)} + \beta(t)v(t-1) \quad (2.33)$$

Denklem (2.33)'de belirtilen $\beta(t)$ terimi Polak – Ribiere kuralına göre denklem (2.34)'deki gibi ifade edilirken, Hestenes – Stiefel kuralına göre ise denklem (2.35)'de gösterildiği gibi tanımlanır.

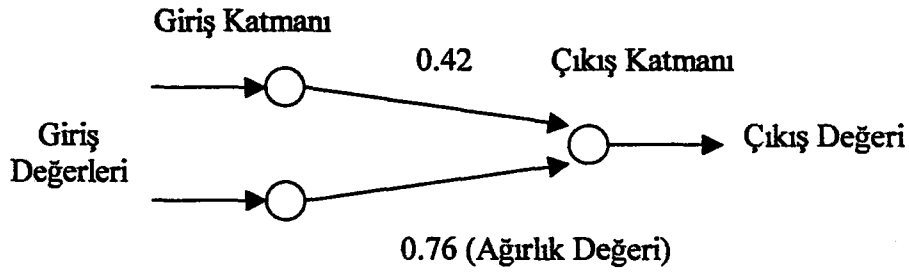
$$\beta(t) = \frac{[\varphi(t) - \varphi(t-1)]^T [\varphi(t)]}{[\varphi(t-1)]^T [\varphi(t-1)]} \quad (2.34)$$

$$\beta(t) = \frac{[\varphi(t-1) - \varphi(t)]^T [\varphi(t)]}{v(t)^T [\varphi(t-1)]} \quad (2.35)$$

CGD Algoritması, adım boyutundan çok, minimum arama doğrultusunu belirleyen bir yöntemdir.

2.7.1.17. Eğitici öğrenme algoritması uygulaması - sayısal örnek

Aşağıda verilen örnekte kullanılan sinir ağı tek katmanlı bir perseptrondur. Uygulamada, giriş değerleri ve hedeflenen çıkış değerleri ağı tanıtarak sinir ağının eğitimi yapılacaktır. İzlenecek aşamalar 2.7.1.1 başlığı altında açıklanmıştır. Çıkış ve hedeflenen çıkış arasındaki farka göre hata hesaplanır ve bu hata değerine bağlı olarak bağlantıların ağırlık değerleri güncellenir.



Şekil 2.17 – Tek katmanlı perseptron kullanılan sayısal örnek

Giriş Değerleri :

a) 0 1

b) 1 1

Çıkış Değeri :

a) 0

b) 1

(a) giriş verileri için;

1. girişten çıkış katmanına: $0 * 0.42 = 0$

2. girişten çıkış katmanına: $1 * 0.76 = 0.76$

Çıkış katmanındaki toplam: $0 + 0.76 = 0.76$

Hata değeri hesabı: $0 - 0.76 = -0.76$

Öğrenme katsayısı: 0.25 seçilmiştir.

1. ağırlığın değişim değeri: $0.25 * 0 * (-0.76) = 0$

2. ağırlığın değişim değeri: $0.25 * 1 * (-0.76) = -0.19$

1. ağırlığın yenilenmesi: $0.42 + 0 = 0.42$

2. ağırlığın yenilenmesi: $0.76 + (-0.19) = 0.57$

(b) giriş verileri için;

1. girişten çıkış katmanına: $1 * 0.42 = 0.42$

2. girişten çıkış katmanına: $1 * 0.57 = 0.57$

Çıkış katmanındaki toplam: $0.42 + 0.57 = 0.99$

Hata değeri hesabı: $1 - 0.99 = 0.01$

1. ağırlığın değişim değeri: $0.25 * 1 * 0.01 = 0.0025$
2. ağırlığın değişim değeri: $0.25 * 1 * 0.01 = 0.0025$
1. ağırlığın yenilenmesi: $0.42 + 0.0025 = 0.4225$
2. ağırlığın yenilenmesi: $0.57 + 0.0025 = 0.5725$

Her giriş değerine ait hata değerlerinin kareleri toplanarak ortalama hata karesi hesaplanır;

$$E = \frac{1}{2} [(-0.76)^2 + (0.01)^2] = 0.28885$$

Buraya kadar olan adımlar tekrar edildikçe E, hata değeri gittikçe küçülecektir. Bu süreç hata değeri kabul edilebilir bir seviyeye düşene kadar devam edecektir [12].

2.7.2. Eğiticişiz öğrenme algoritmaları

Eğiticişiz öğrenmede, öğrenme süreci sonucunun ne olacağı önceden bilinemez. Öğrenme süreci içinde sınırları birleştiren bağlantıların ağırlıkları verilen giriş değerlerine bağlı olarak belirli bir aralık içinde düzenlenir. Burada amaç, birbirlerine benzer birimleri belirli değer aralıkları içinde gruplamaktır. Kohonen'in geliştirdiği Özdenetimli Harita (SOM – Self Organizing Map), biyolojik insan beynini simüle etmeye en uygun eğiticişiz öğrenme algoritmasıdır.

Bilindiği gibi beynin korteks tabakası her biri değişik işlevleri yerine getiren farklı bölgelere ayrılmıştır. Sinir hücreleri gelen bilgilere göre kendilerini gruplar durumunda organize eder. Bu gelen bilgiler, tek bir sinir hücresi tarafından alınmakla kalmaz, bu sinir hücresinin komşuluğundaki diğer sinir hücrelerini de etkiler. Bu organizasyonun sonucunda bir çeşit harita ortaya çıkar. Bu harita içinde benzer işlevleri yerine getiren birimler birbirlerine yakın bir şekilde düzenlenir. Bu kendi kendine organizasyon, yapay sinir ağı tarafından da gerçekleştirilebilir [12]. SOM ağında, verilen giriş örneğine en yakın çıkış düğümünü bulmak için uzaklık ölçüsü kullanılır [6, 13].

SOM algoritması aşağıda verilen adımlarla eğitilir;

- i. X giriş vektörü ağı tanıtılır.
- ii. X_i vektörü ile W_{ij} ağırlık vektörü arasındaki D_{ij} uzaklığı Euclidean uzayında hesaplanır. Buradaki $j = 0, 1, 2, \dots, m-1$ indisi ağıdaki nöron sayısını temsil ederken, $i = 0, 1, 2, \dots, n-1$ indisi de giriş vektörünün boyutunu belirtir.

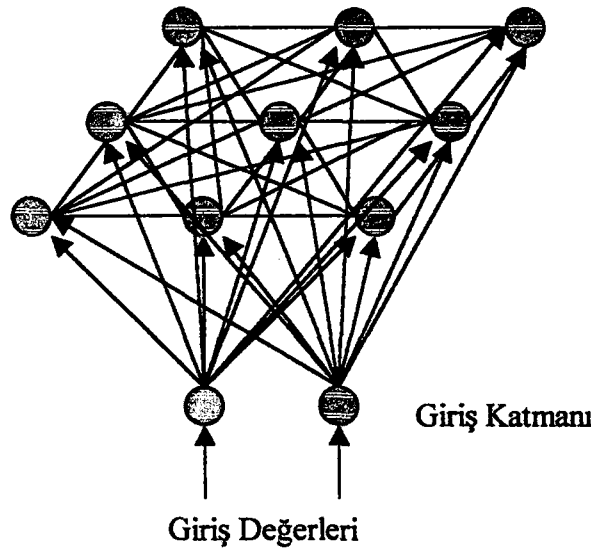
$$D_{ij} = \|(x_i(t) - w_{ij}(t))\| \quad (2.36)$$

- iii. X giriş vektörüne en yakın W ağırlık vektörü bulunarak W_w vektörü seçilir ve D uzaklığı içindeki diğer ağırlık vektörlerinin merkezi olarak belirlenir.
- iv. W_w 'nin D uzaklığı içindeki diğer ağırlık vektörleri denklem (2.37) uyarınca eğitilir.

$$w_{ij}(t+1) = w_{ij}(t) + \eta[x_i(t) - w_{ij}(t)] \quad (2.37)$$

- v. Yakınsama sağlanana dek i-iv adımları her giriş vektörü için tekrar edilir [2].

Kohonen'in Özdenetimli Haritası Şekil 2.18'de gösterilmiştir. Şekilden de anlaşılacağı gibi giriş katmanındaki her bir sinir hücresi diğer tüm sinir hücreleriyle ilişkilidir. Ayrıca harita üzerindeki her bir sinir hücresinin de diğerleriyle ilişkisi vardır [19].



Şekil 2.18 – Özdenetimli Harita

2.8. Yapay Sinir Ağlarının Kullanım Nedenleri

Yapay sinir ağları, gerçek dünyaya ait ilişkileri tanıyabilirler. Sınıflandırma, tahmin ve işlev uydurma gibi görevleri gerçekleştirebilirler. Yapay sinir ağlarının, ani değişen değerlerden örnekler alarak doğrudan doğruya örneklerin birbirleri arasındaki benzer ilişkileri öğrenme yetenekleri vardır. Yapay sinir ağlarının bir olaydaki verilerden hareketle önceki bilgileri bilmeseyse bile sonuç çıkarma yetenekleri de vardır. Çünkü yapay sinir ağları klasik programlama ile çalışmazlar. Aralarındaki fark; ağların bir işi yapmaları için eğitilmelerinin gerekliliğidir. Bu eğitme işlemi, çevrimlerin oluşturulması, koşulların saptanması ve bunların sınanması ile bir derleyici üzerinde koşturma işlemlerini içermez. Bütün bunların yerine eğitme prosedürleri seçim, analiz ve verilerin bir araya getirilmesi ile başlar.

Sinir ağı gerekli ilişkileri oluştururken, ağı geliştiren kişinin, klasik programlarda olduğu gibi programı adım adım izlemesi, mümkün değildir.

Yapay sinir ağlarında veri toplamak için bir ön sorgulama veya açıklama gerekmez. Sinir ağları genelleştirilebilir. Genelleştirme özelliği sayesinde giriş verilerindeki gürültü ve bozucu etkiler ağıın işleyişini etkilemez. Yapay sinir ağları paralel işlemleri aynı anda çok hızlı yürütebilirler. Endüstriyel kontrol işlemleri, yapay sinir ağları için, oldukça uygun uygulama alanlarıdır [6].

2.9. Yapay Sinir Ağlarının Dezavantajları

Yapay sinir ağlarında amaç, ham verilerden toplanarak elde edilen değerleri analiz edip değişkenleri seçerek, veriyi ön işlemlerden geçirip ağıın etkili şekilde öğrenmesini sağlamaktır. Yapay sinir ağları, eğitilecek bir işlev bulunamaması durumunda çözümde hata yapabilmektedir. Bunu engellemek için sisteme yeterli sayıda giriş verisi tanıtılmalıdır.

Yapay sinir ağlarının sonuçlarını değerlendirmek zor olabilir. Sonuç almak, yüzlerce giriş örneğinin hesaplanmasına bağlı olabilir.

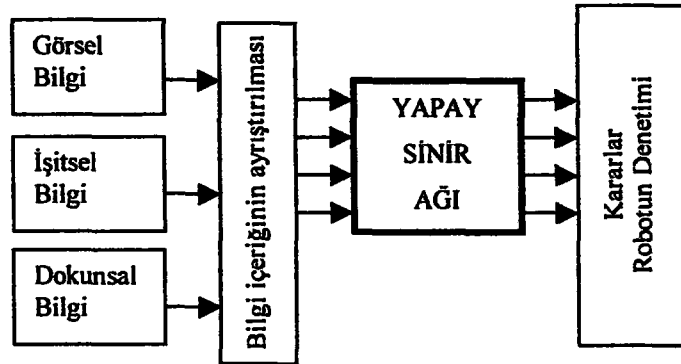
Bir yapay sinir ağıının kalitesi ve kapasitesi uygulamadaki hızıyla orantılıdır. Yapay sinir ağlarında eğitme işlemi yavaş ve pahalı bir süreç olabilir. Maliyeti arttıran sebeplerden ilki eğitme verilerinin toplanması ve değerlendirilmesidir.

2.10. Yapay Sinir Ağlarının Potansiyel Uygulama Alanları

Yapay sinir ağları, gerek yapısal açıdan çeşitlilikleri, gerekse öğrenme algoritmalarının çeşitliliği açısından karmaşık problemlerin çözümünde tasarımcıya oldukça değişik olanaklar sunmaktadır. Yapay zeka alanındaki gelişmeler, günlük hayatın bir çok alanında kullanıcı isteklerini algılayabilen ürünler ortaya çıkarmıştır.

Amerikan Ulusal Havacılık Dairesi'nin bir açıklamasında, bazı savaş uçaklarında yapay sinir ağlarının simülatörlerden ve gerçek zamanlı uçuşlardan elde edilen verilerle eğitmek suretiyle otopilot olarak kullanılabildiğine değinilmektedir. Günlük hayatın içerisinde de işlemsel akıl içeren sistemler, gerek otomatik odaklama sistemine sahip kamera modülleri olarak, gerekse sürücüsüz otomobiller olarak gündeme gelebilmektedir. Otomatik vitesli otomobillerde, vites kutusu işlevini yerine getiren mekanik düzeneğe kumanda eden modüle, sürücünün otomobil kullanma tarzının öğretilebileceği ve bu yolla sürücü açısından kullanım kolaylığının arttırılabileceği verilebilecek bir başka örnektir.

Günümüz teknolojinin insan hizmetine sunduğu bir ürün tipi de ev içinde kullanılan hizmetçi robotlardır. Bu robotlar algılayıcıları ile içinde bulundukları ortamdan bilgi alıp değerlendirerek, belirli bir görevi gerekli koşulların gerçekleşmesi durumunda yerine getirebilmektedirler. Veriler, yapay sinir ağları kullanılarak değerlendirilir. Şekil 2.19'da bilgi akışı ve ağ yapısının robot sistemindeki yeri gösterilmiştir [11].



Şekil 2.19 - Bilgi Akışı

Yapay sinir ağıları, askeriyede hedef tanıma, iletişimde yankı sınırlama, güvenlikte optik karakter algılaması, finans sektöründe mali işlemler gibi değişik alanlarda uygulama kaynağı bulur.

1992 yılında London Business School ve University Collage London tarafından kurulan Financial NeuroForecasting Centre para piyasaları ve global marketler üzerine yoğunlaşarak finansal istatistikler çıkartmaktadır.

Fujitsu Ltd., Kawasaki ve Nippon Steel Corp. proses kontrolünde yapay sinir ağılarını kullanmaktadırlar [20].



3. HATANIN GERİYE YAYILIMI ALGORİTMASI

3.1. Giriş

Yapay sinir ağlarının parametrelerinin güncellenmesi için literatürde en çok kullanılan yöntem hatanın geriye yayılması algoritmasıdır. Ses tanıma problemlerinden nonlinear sistem tanıma ve denetimi problemlerine kadar yapay sinir ağları ile çözüm üretilen pek çok alanda kullanılan bir yöntemdir.

BP eğitilmesi sırasında ağ, her giriş örneğini, çıkış düğümlerinde sonuç üretmek üzere saklı katmanlardaki düğümlere geçirir. Sonra da çıkış katmanındaki hataları bulabilmek için amaçlanan hedef sonuçtan gerçek sonucu elde eder. Bundan sonra ağ, çıkış hatalarının türevini çıkış katmanından geriye doğru saklı katmanlara geçirir. BP algoritmasına ismini veren de işte bu hataların geriye doğru yayılması özelliğidir [6].

Geriye yayılım algoritması sadece ileri beslemeli ağlar için kullanılabilir. Herhangi bir katmandaki nöron bir üst katmandaki nöronların hepsine bağlı olduğu gibi bir alt katmandaki nöronların da hepsine bağlıdır. Üst katman nöronlarına işaret gönderirken alt katman nöronlarından işaret alır. Bu yapı tam bağlılık olarak isimlendirilir. Bir nöron, bulunduğu tabaka içindeki hiçbir elemana doğrudan bağlı olamaz [27].

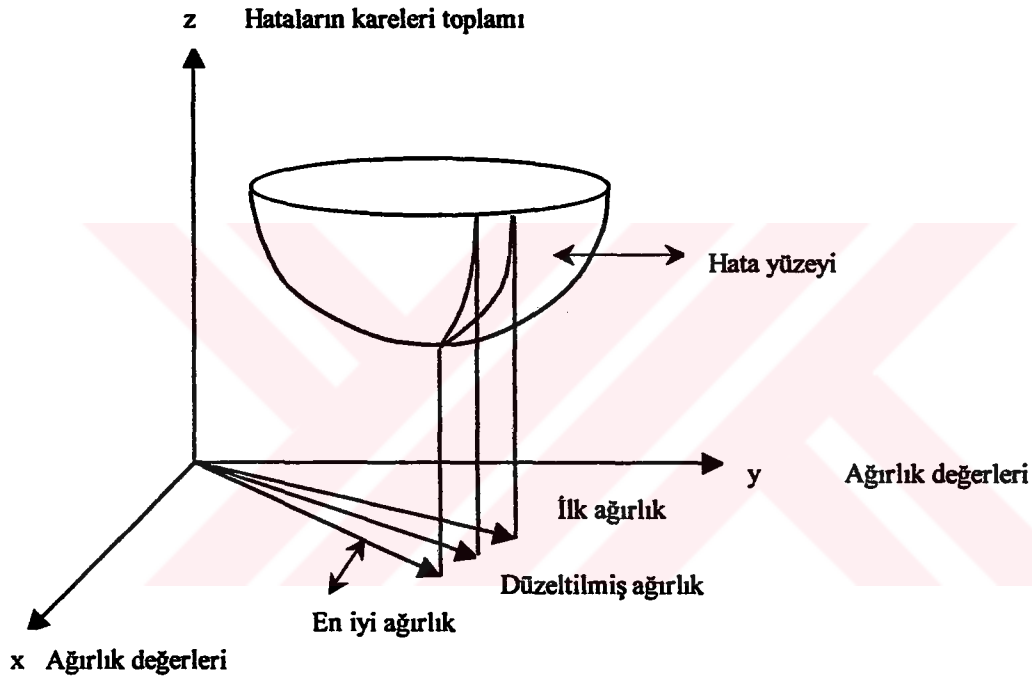
Her çıkış ve saklı düğümlerin hata değerleri bulunduktan sonra, düğümler kendi hatalarını azaltmak için kendi ağırlıklarını ayarlar. Ağırlık değiştirme denklemlerinin ağdaki ortalama hata karesini minimize edecek şekilde düzenlendiği daha önce belirtilmişti.

3.2. Hata Minimizasyonu

Hata minimizasyonunun geometrik bir yorumunu yapmak mümkündür. Bunu görebilmek için ağırlıkların mümkün olan tüm değerleri, hataların kareleri toplamına karşı gelecek şekilde (xyz) koordinat sisteminde çizilir. Bu çizim sonunda hata

yüzeyi küresel bir tasa benzer. Tasın en alt kısmı hataların kareleri toplamının en küçüklerine karşı gelmektedir. Eğitim sırasında amaç, ağırlıklar kümesinin en iyisini bulmak demek olan tasa benzer şeklin en alt kısmını bulmaktır.

BP algoritması, o andaki ağırlıklar yerine, yüzey hatasının eğimini hesaplayarak amacına ulaşır. Daha sonra da bu ağırlıkları tasın alt kısmına doğru artımsal olarak değiştirir. İşte bu artımsal olarak tasın üst kısmından alt kısmına doğru ilerleme işlemine Gradyan Azalması (Gradient Descent) denir. Hata minimizasyonunun geometrik yorumu Şekil 3.1’de gösterilmiştir [18].

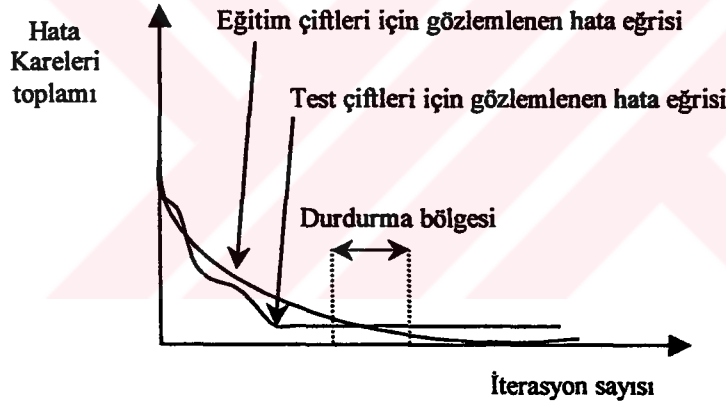


Şekil 3.1 - Hata minimizasyonunun geometrik yorumu

3.3. Yapay Sinir Ağlarında Öğrenme ve Ezberleme Problemi

Yapay sinir ağlarında öğrenme ve ezberleme sınırının belirlenmesinde iki tipte uygulamadan söz edilir. Birinci tipte uygulama sonlu sayıda durumdan oluşur ve bu durumların her biri eğitim çiftleri tarafından temsil edilir. Lojik işlevler, XOR problemleri bu duruma örnektir. Girişlerin alabileceği tüm değerler eğitim kümesi tarafından içeriliyorsa, toplam karesel hata değeri sıfıra ne kadar yaklaşırsa ağ başarımı o kadar yüksek olacaktır. İkinci tipte uygulamalarda ise sonsuz sayıda eleman bulunması gerekir. Pratikte sonsuz sayıda eğitim çifti kullanılamayacağına göre, sonlu sayıda çift ile eşleştirmenin gerçekleştirilebilmesi, öğrenme ve ezberleme

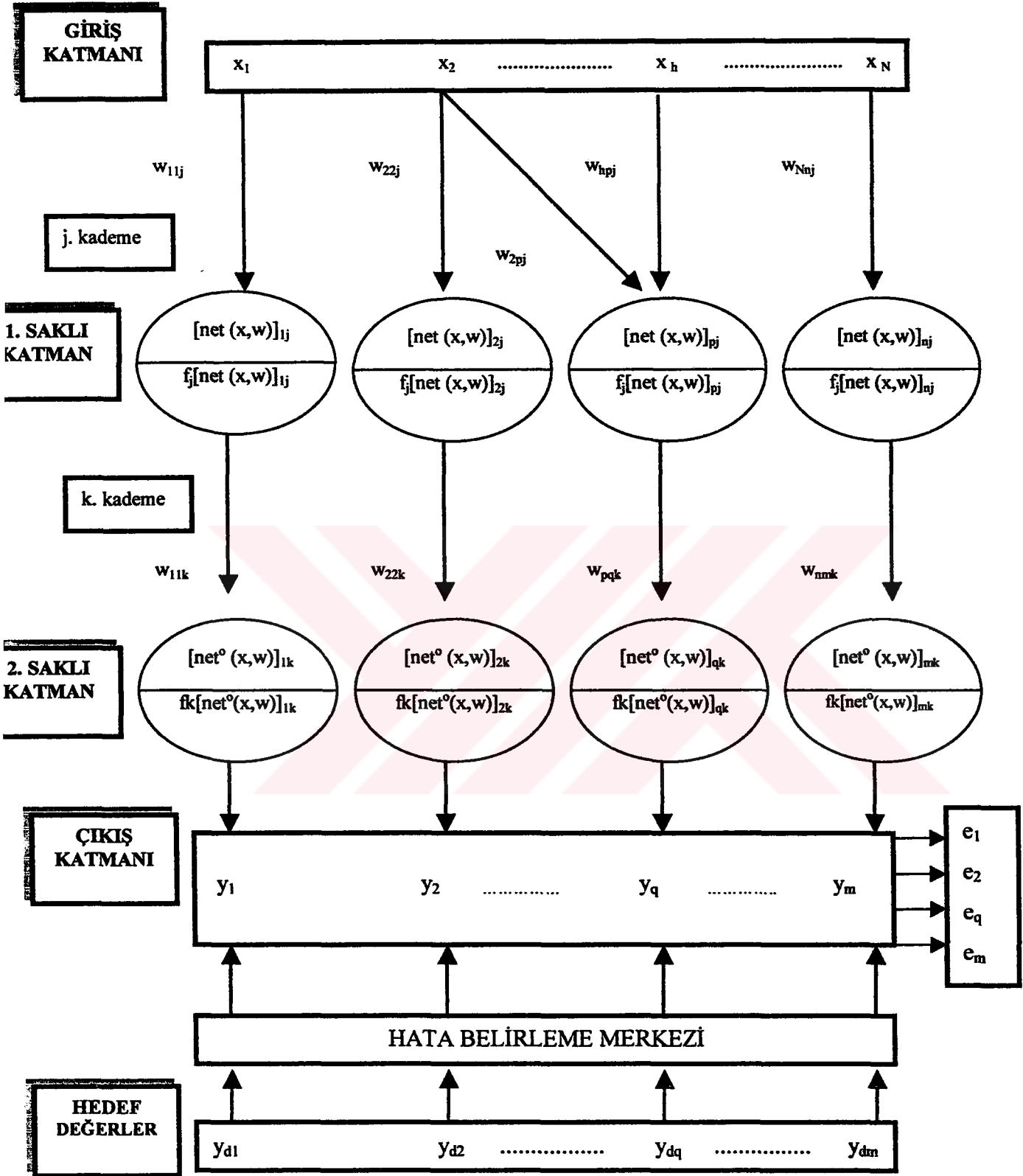
arasındaki sınırın iyi çizilmesini gerektirir. Eğer sinir ağı, verilen noktalarda sıfıra çok yakın bir ortalama karesel hata elde edecek biçimde eğitilmişse, eğitim çiftleri kümesinde bulunmayan ancak aynı sürekli yüzey denklemlerinden elde edilen bir başka noktada büyük hata içeren bir cevap üretmesi olası bir durumdur. Bu nedenle eğitim çiftleri kümesinin yanında test çiftleri kümesi de oluşturulmalı ve bu iki küme mümkün olduğunca az sayıda özdeş eleman bulundurmamalıdır. Eğitim esnasında eğitim çiftlerine göre parametre güncellemesi yapılırken, test çiftleri kümesi için de gözlemlenen ortalama hata karesi kontrol edilmelidir. Her iki büyüklüğün de azalma gösterdiği süre boyunca genelleme mantığına uygun bir öğrenme gerçekleşmektedir. Bir süre sonra eğitim çiftleri için hata azalmaya devam ederken test çiftleri için hata artmaya başlayacaktır. Bu anda eğitime son verilmelidir; çünkü kullanılan eğitim çiftleri için artık ezberleme süreci başlamıştır. Bu durum Şekil 3.2’de gösterilmektedir [8, 11, 21].



Şekil 3.2 – Eğitimin durdurulması için uygun durdurma bölgesi

3.4. BP Algoritmasında Ağırlıkların Yenilenmesi

Geriye yayılım ağında önce giriş vektörleri ağı uygulanır ve bu giriş değerlerine karşılık gelen çıkış değerleri belirlenir. Gerçekleşen çıkışlarla hedeflenen çıkış değerleri karşılaştırılarak hata bulunur. Hatanın azaltılması için ağırlıkların pozitif veya negatif yönde değiştirilmesi gerekliliği belirlenir. Her ağırlık değeri için değişiklik miktarı belirlenerek düzenleme yapılır. Algoritma bütün eğitim verileri için, hata limit değeri sınırına düşene kadar tekrar edilir. Genel bir BP sinir ağı yapısı Şekil 3.3’de gösterilmiştir.



Şekil 3.3 – BP sinir ağı yapısı

3.4.1. Aktivasyon işlevinin türevleri

BP algoritmasında kullanılan zincir kuralı, aktivasyon işlevinin türevini almayı gerektirir. Lojik, tanjant hiperbolik ve lineer işlevler için türev denklemleri aşağıda verilmiştir.

Aşağıdaki denklemlerde kolaylık sağlanması amacıyla $net(x,w) = I$ olarak gösterilmiştir.

Lineer aktivasyon işlevinin türevi;

$$\begin{aligned}\psi(I) &= I \\ \dot{\psi}(I) &= 1\end{aligned}\tag{3.1}$$

Lojik aktivasyon işlevinin türevi;

$$\begin{aligned}\psi(I) &= \frac{1}{1 + e^{-\beta I}} \\ \dot{\psi}(I) &= \beta \psi(I)(1 - \psi(I))\end{aligned}\tag{3.2}$$

Tanjant hiperbolik aktivasyon işlevinin türevi;

$$\begin{aligned}\psi(I) &= \frac{e^{\beta I} - e^{-\beta I}}{e^{\beta I} + e^{-\beta I}} \\ \dot{\psi}(I) &= \beta (1 - \psi(I)^2)\end{aligned}\tag{3.3}$$

β sabiti eğim parametresidir ve genellikle 1 değerinde seçilir.

3.4.2. Çıkış Tabakasındaki Ağırlıkların Yenilenmesi

Herhangi bir çıkış birimi için oluşan hatayı;

$$e_q(t) = y_{dq}(t) - y_q(t)\tag{3.4}$$

şeklinde tanımlayalım. Burada, $y_{dq}(t)$ istenen doğru çıkışa, $y_q(t)$ ise gerçekleşen çıkış değerine karşılık gelmektedir.

Genelleştirilmiş delta kuralına göre minimize edilmesi gereken hata, her çıkış birimi için oluşan hataların kareleri toplamıdır.

$$E_q = \frac{1}{2} \sum_{k=1}^m e_q^2 \quad (3.5)$$

Ağırlıkların değişim yönünü bulmak için E_q 'nin w_{hpj} ağırlıklarına göre negatif ∇E_q gradyanı hesaplanır. Böylece ağırlık değerlerini hatayı azaltacak yönde değiştirebiliriz. E_q 'yu ağırlık uzayında bir yüzey olarak görmek kullanışlı olmaktadır. Ağırlık değişimleri, hata değerinde hızlı azalışın gerçekleştiği negatif gradyan yönünde yapılmaktadır. Her katmanında lojistik sigmoid aktivasyon işlevi kullanılan iki katmanlı bir yapay sinir ağının, ∇E_q gradyanı hesaplanacak olursa denklem (3.6), denklem (3.7), denklem (3.8), denklem (3.9), denklem (3.10) ve denklem (3.11) elde edilecektir.

Hedeflenen çıkış değerinin lojistik sigmoid işlevine göre [0 1] aralığında değişeceği bilinmektedir.

$$\Delta w_{pqk} = -\eta_{pq} \frac{\partial E_q}{\partial w_{pqk}} \quad (3.6)$$

$$\Delta w_{pqk} = -\eta_{pq} \frac{\partial E_q}{\partial \psi_{qk}} \frac{\partial \psi_{qk}}{\partial I_{qk}} \frac{\partial I_{qk}}{\partial w_{pqk}} \quad (3.7)$$

$$y_q = fk[net^o(w, x)]_{qk} = \psi_{qk} \quad (3.8)$$

$$f_j[net^o(x, w)]_{pj} = \psi_{pj}$$

$$\Delta w_{pqk} = -\eta_{pq} (-2) [y_{dq} - \psi_{qk}] \psi_{qk} [1 - \psi_{qk}] \psi_{pj} \quad (3.9)$$

$$\delta_{pqk} = (-2) [y_{dq} - \psi_{qk}] \psi_{qk} [1 - \psi_{qk}] \quad (3.10)$$

olarak tanımlanırsa,

$$w_{pqk}(t+1) = w_{pqk}(t) - \eta_{pq} \delta_{pqk} \psi_{pj} \quad (3.11)$$

bulunur.

Ağırlık değişimlerinde sonuca ulaşmayı hızlandırmak için momentum terimi ile, önceki ağırlık değişimi oranının son eşitliğe eklenmesi mümkündür.

$$w_{pqk}(t+1) = w_{pqk}(t) - \eta_{pq} \delta_{pqk} \psi_{pj} + \alpha \Delta w_{pqk}(t-1) \quad (3.12)$$

3.4.3. Ara Tabaka Ağırlıklarının Yenilenmesi

Ara tabaka ağırlıklarının belirlenmesinde, ara tabaka çıkış değerleri hakkında bir ön bilgi bulunmamaktadır. Bu nedenle saklı katman çıkışları belirgin bir değere hedeflenmemektedir. Sezgisel olarak anlaşılabileceği gibi E_q toplam hatası, ara tabakalardaki çıkış değerleri ile ilişkilidir. Bunun denklemsel gösterimi;

$$\Delta w_{hpj} = -\eta_{hp} \sum_{k=1}^m \frac{\partial E_q}{\partial w_{hpj}} \quad (3.13)$$

$$\Delta w_{hpj} = -\eta_{hp} \sum_{k=1}^m \frac{\partial E_q}{\partial \psi_{qk}} \frac{\partial \psi_{qk}}{\partial I_{qk}} \frac{\partial I_{qk}}{\partial \psi_{pj}} \frac{\partial \psi_{pj}}{\partial I_{pj}} \frac{\partial I_{pj}}{\partial w_{hpj}} \quad (3.14)$$

$$\frac{\partial E_q}{\partial \psi_{qk}} = (-2)[y_{dq} - \psi_{qk}]$$

$$\frac{\partial \psi_{qk}}{\partial I_{qk}} = \beta \psi_{qk} [1 - \psi_{qk}]$$

denklem (3.2)'de belirtilmişti.

$$\frac{\partial I_{qk}}{\partial \psi_{pj}} = w_{pqk}$$

$$\frac{\partial \psi_{pj}}{\partial I_{pj}} = \beta \psi_{pj} [1 - \psi_{pj}]$$

$$\frac{\partial I_{pj}}{\partial w_{hpj}} = x_h$$

değerleri denklem (3.14)'de yerine konursa;

$$\frac{\partial E_q}{\partial w_{hpj}} = \sum_{k=1}^m (-2) [y_{dq} - \psi_{qk}] \cdot \beta \psi_{qk} [1 - \psi_{qk}] \cdot w_{pqk} \cdot \beta \psi_{pj} [1 - \psi_{pj}] \cdot x_h \quad (3.15)$$

denklem (3.15) ve denklem (3.16) elde edilir.

$$\frac{\partial E_q}{\partial w_{hpj}} = \sum_{k=1}^m \delta_{pqk} \cdot w_{pqk} \cdot \beta \psi_{pj} [1 - \psi_{pj}] x_h \quad (3.16)$$

Ara tabakadaki hata terimi denklem (3.17)'deki gibi tanımlanır.

$$\delta_{hpj} = \delta_{pqk} w_{pqk} \frac{\partial \psi_{pj}}{\partial I_{pj}} \quad (3.17)$$

Denklem (3.18), denklem (3.16) ve denklem (3.17)'den bulunabilir.

$$w_{hpj}(t+1) = w_{hpj}(t) - \eta_{hp} \cdot x_h \cdot \delta_{hpj} \quad (3.18)$$

3.5. BP Algoritmasında Saklı Katman Nöronlarının Sayısının Belirlenmesi

BP algoritmasında, saklı katman nöron sayısının belirli bir kurala göre saptanması söz konusu değildir. Deneme yanılma yöntemi kullanılarak optimal nöron sayısına ulaşılmaya çalışılır.

Saklı katmandaki nöron sayısı, yakınsama açısından başarı sınırı sayılmaktadır. Giriş ve çıkış verilerinin artması durumunda, saklı katmanda kullanılması gereken nöron sayısının da artacağı düşünülebilir. Ancak saklı katman nöron sayısını arttırmanın, yakınsamada iyileşme göstermemesi de mümkündür. Ağ yapısı dikkate alınarak, saklı katmandaki nöronlara yapılacak ilavelerin, ağın giriş ve çıkış nöronlarına bağlı olarak hesap yükünü arttıracığı söylenebilir.

Lippmann, tek saklı katmanlı sinir ağlarında saklı katman nöron sayısının $M(N+1)$ değerine eşit olması gerektiğini savunmuştur. M değeri çıkış nöron sayısını belirtirken, N değeri de giriş nöron sayısını belirtir. Uygulamada, Lippmann'ın belirttiği $M(N+1)$ miktarının en büyük değer olarak kabul edilmesi ve bu değerden daha az sayıda nöron kullanılması ile performansın arttırıldığı görülmüştür.

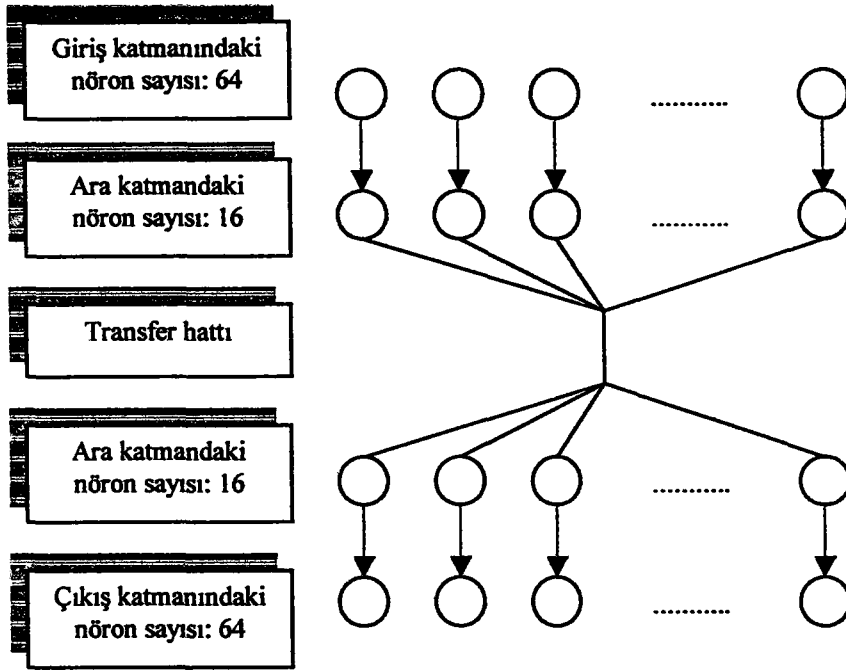
3.6. Geriye Yayılım Algoritmasının Uygulama Alanları

Geriye Yayılım Algoritması'nın pek çok alanda uygulamaları bulunmaktadır. Bunlardan bazıları; [22]

- Aminoasit dizisinden protein yapısının anlaşılması
- Seslerin sınıflandırılması [23]
- Tavla oyununda doğru hamlenin belirlenmesi [24]
- Karakter algılama [29]
- Veri sıkıştırma
- Yazı karakterlerinin sesli karakterlere dönüştürülmesi
- Elektriksel delinme gerilimi hesaplaması [25]
- Yalıtkan tanılamasında kullanımı [28]
- Cinsiyet ayrıştırma
- Üniversal motor dinamiğinin tanılanması [30]
- Finansal bilgilerin teknik analizinde kullanımı [12]
- İletim hatlarının taşıma kapasitesinin bulunması [31]
- Güdümlü roketlerin uçuş rotasının belirlenmesi [32]
- EGK vurularının sınıflandırılması [33]
- Kontrol problemlerinde kullanımı [34]
- Elektrik tesislerinde mesafe rölesi ve mesafe korumasında kullanımı
- İşlev yaklaşıklama [37]

3.6.1. Geriye yayılım algoritması ile veri sıkıştırma

Veri sıkıştırma, genellikle bilginin bir noktadan diğer bir noktaya bir bilgi kanalı vasıtasıyla taşınması sırasında, taşınacak olan veri miktarının azaltılması işlemidir. Genellikle düzensiz ve düzgün tanımlanmamış yapılarda karşılaşılan video görüntü verilerinin sıkıştırılması işlemi algoritmik olarak zor olduğundan bu tür problemlerde yapay sinir ağı yaklaşımı gerçekleştirilebilir. Çünkü bir geriye yayılım ağı, “n” boyutlu uzaydan “m” boyutlu uzaya dönüşüm gerçekleştirebilecek şekilde rahatlıkla eğitilebilir. Sıkıştırma işleminde $n > m$ durumu söz konusudur. Herhangi bir video görüntüsü, şekil elemanları matrisi gibi düşünülerek n boyutlu uzayda bir vektöre karşılık getirilmektedir.



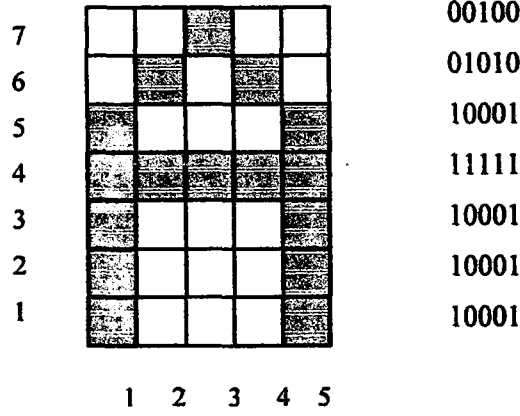
Şekil 3.4 – Yapay sinir ağları ile veri sıkıştırma

Şekil 3.4’de 64 nöronu olan giriş vektörünün yapay sinir ağı ile sıkıştırılarak iletim hattı ile taşınması ve daha sonra alıcı ucun çıkış katmanında ilk katmanın giriş vektörünün tekrar elde edilmesi gösterilmektedir. Alıcı tarafta da 16 giriş vektörü 64 çıkış vektörü olan ikinci bir sinir ağı bulunmaktadır. Giriş vektörü 64 byte olarak düşünülürse, iletim 16 byte üzerinden gerçekleşecektir. Sinir ağı, giriş bilgisinin kayba uğramadan yeniden elde edilebileceği biçimde eğitilir. Bu tasarıma göre taşınacak veri $\frac{1}{4}$ oranında azaltılmış olur. Sonuçta, ağ “n” boyutlu uzaydan “n/4” boyutlu uzaya dönüşüm gerçekleştirecek ve daha sonra ters dönüşüm ile tekrar “n” boyutlu uzaya geçecektir.

3.6.2. Geriye yayılım algoritması ile karakter algılama

Alfabadeki harflerin yapay sinir ağları kullanılarak algılanması, harf karakterinin 5x7 lik ikili sayılardan oluşan bir matris ile temsil edilmesi sonucunda gerçekleştirilmiştir. Karakteri çevreleyen ve 5x7 lik birimlere ayrılmış alanda içi dolu bulunan her birim 1 değerini, içi boş bulunan her birim ise 0 değerini almaktadır. Böylece şekil, ikili sayı sisteminde kodlanabilir. Sinir ağı yapısı, 35 giriş değerine karşılık ikili düzende 4 çıkış değeri ile kurulur. Alfabadeki ilk 16 harfin algılanması gerçekleştirilecektir. Böylece giriş vektörü 35x16 boyutunda olurken çıkış vektörü

4x16 boyutunda olacaktır. A karakterinin giriş vektörü $x_a = [0\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 0\ 1]$, çıkış vektörü $y_a = [0\ 0\ 0\ 0]$ olarak tanıtılır.



Şekil 3.5 – A harfinin ikili sayı düzeninde kodlanması

Bu şekilde $y_b = [0\ 0\ 0\ 1]$ çıkışı alfabenin B harfine, $y_c = [0\ 0\ 1\ 0]$ çıkışı alfabenin C harfine karşılık gelecektir.

Karakter görüntüsünde bozukluk olması durumunda bile ağ genelleştirme yeteneği sayesinde karakteri algılayarak doğru çıkışı üretebilecektir [29].

3.6.3. Geriye yayılım algoritması ile yalıtkan tanınması

Yüksek gerilim güç transformatörlerinin işletme karakteristikleri, kullanılan yalıtım maddesinin özelliğine bağlıdır. Kağıt ve benzeri selüloz bazlı ürünler, düşük maliyeti ile iyi elektriksel ve mekanik özelliklerinden dolayı yağlı tip trafolarla genelde tercih edilen yalıtım maddeleridir. Transformatörün dielektrik özellikleri, nem oranının yükselmesi ile azalma eğilimi gösterir. %2 su buharı bulunduran bir ortama oranla oksijenden arındırılmış ortamdaki ömür süresi 20 kat fazladır.

Yalıtkan tanılama işleminin amacı yüksek gerilimde kullanılan elemanların zarar görmesine neden olacak dielektrik bozulmanın belirlenmesini sağlamaktır. Yalıtıktaki nem oranının artması ile artan proton yoğunluğu iletkenliği artırırken delinme sürecini hızlandıracaktır. Geriye yayılım algoritması ile farklı nem oranlarında yalıtım delinme süreçleri sinir ağına tanıtılarak optimal çalışma ömrü hesaplanır [28].

3.6.4. Üniversal motor dinamiğinin tanılanması

Üniversal motorlar, seri sargılı, armatür kontrollü, özel amaçlı motorlardır. Bu motorlar, hem doğru akım hem de alternatif akım uyarı modlarında aynı verim ve çalışma özelliklerini gösterirler. Sürekli güç tipleri 1mHP'den 1HP'ye kadar olabilir. Bu motorlar uygulama alanına göre 10000 devir/dakika'lık hızlara kadar çıkabilirler. Üniversal doğru akım motorunun dinamik denklemleri aşağıdaki gibidir.

$$e_a(t) = R_a i_a(t) + L_a \frac{di_a(t)}{dt} + e_b(t) \quad (3.19)$$

$$e_b(t) = K_b i_a(t) \omega(t) \quad (3.20)$$

$$T_g = K_t [i_a(t)]^2 \quad (3.21)$$

Denklem (3.19)'da belirtilen $e_a(t)$ uygulanan gerilimi, L_a armatür endüktansını, R_a armatür direncini, denklem (3.20)'deki K_b geri elektromotor kuvvet katsayısını, denklem (3.21)'de belirtilen $T_g(t)$ üretilen torku ve K_t tork katsayısını göstermektedir. Mekanik tork denklemi, denklem (3.22)'deki gibidir.

$$T_g = J \dot{\omega} + D\omega \quad (3.22)$$

Motorun dinamik denklemleri düzenlenip geriye doğru fark yöntemi ile ayrıştırıldığında denklem (3.23) elde edilir.

$$\left(\frac{1}{T} + D\right) \omega^3(k) + \left[2c\left(\frac{J}{T} + D\right) - \frac{1}{T}\right] \omega^2(k-1) \omega^2(k) + \left[c^2\left(\frac{J}{T} + D\right) - 2c\frac{J}{T}\right] \omega(k-1) \omega(k) - \left[c^2\left(\frac{J}{T} + D\right) + p^2\right] \omega(k-2) = 0$$

$$c = \left(R_a + \frac{L_a}{T}\right) \frac{1}{K_b} \quad (3.23)$$
$$p = e_a(k) \frac{\sqrt{K_t}}{K_b} + \frac{L_a}{K_b T} \sqrt{\left(\frac{J}{T} + D\right) \omega(k-1) - \frac{J}{T} \omega(k-2)}$$

Denklem (3.23), her bir k anında uygulanan $e_a(k)$ gerilimi ve önceki $w(k-1)$ ile $w(k-2)$ hızlarının işlevi olarak $w(k)$ hızını verir. Böylece motor hızı denklem (3.24)'deki gibi ifade edilir.

$$w(k) = \Omega [e_a(k), w(k-1), w(k-2)] \quad (3.24)$$

Yapay sinir ağına tanıtılacak eğitsel küme giriş kalıpları $[e_a(k), w(k-1), w(k-2)]^T$ ve buna karşılık gelen çıkış kalıpları $[w(k)]$ veri kümeleridir. Yapay sinir ağı eğitim sinyali olarak çok frekanslı, düz spektrumlu Schroeder fazlı harmonik sinyal (SPHS) kullanılmıştır. Gürültülü ortamda eğitilmiş tanıyıcı performansı da kabul edilebilir düzeyde olmuştur [30].

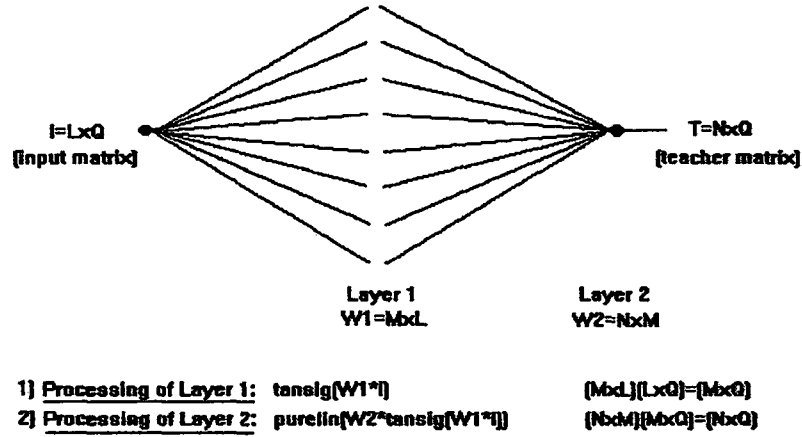
3.6.5. Elektrik tesislerinde mesafe rölesi ve mesafe korumasında geriye yayılım algoritmasının kullanımı

Elektrik tesislerini kısa devreye karşı korumak için mesafe rölesi kullanılır. Şebekede bir arıza oluştuğunda, arıza yerinin besleme noktasından ayrılması süresince geçen zaman yani açma süresi, arıza yerinin rölenin bulunduğu yere uzaklığına bağlıdır. Mesafe ölçüsü olarak, korunacak olan hattın empedansından faydalanılır. Arıza yeri röleden ne kadar uzaksa, ölçülen direnç o kadar büyüktür ve rölenin açma zamanı o kadar uzundur. Mesafe röleleri, yerleştirildikleri yerin yakınında oluşan hatalarda kısa açma süresi ile, uzağındaki hatalarda da uzun açma süresi ile çalışırlar [35].

Mesafe korumasında geriye yayılım algoritmasının kullanımı mesafe rölesinin algıladığı arıza yerlerinin akım ve gerilim dağılımları uyarınca bulunmasını sağlar. Faz gerilimleri ve faz akımları giriş değerleri olarak sinir ağına tanıtılır. Sinir ağı, arıza durumunda 0, arıza olmaması durumunda 1 çıktısı verecek şekilde eğitilir. Sinir ağının arıza olduğu şeklindeki cevabı durumunda mesafe rölesi açma emrini gerçekleştirir.

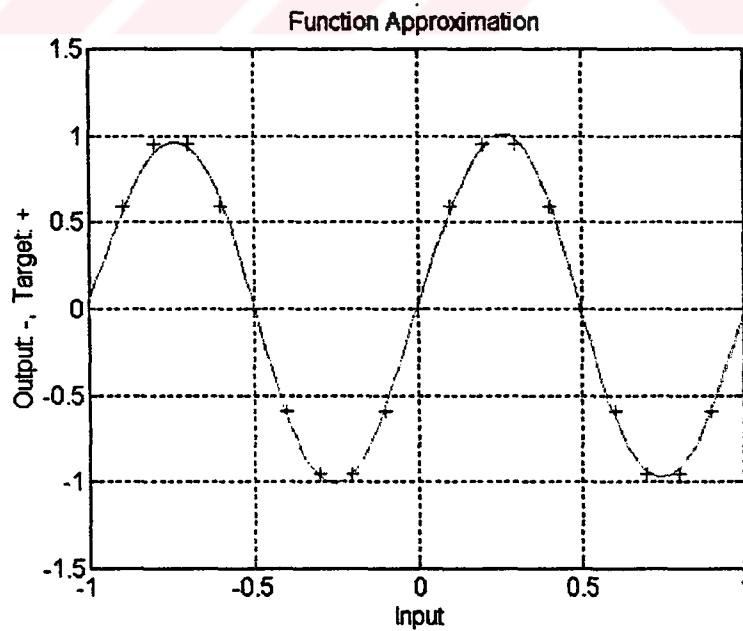
3.6.6. İşlev Yaklaşıklaama

Matlab Neural Network Toolbox kullanarak iki katmanlı bir geriye yayılım algoritması ile işlev yaklaşımı örneği Şekil 3.6'da gösterilmiştir.



Şekil 3.6 – Giriş katmanı, saklı katman ve çıkış katmanından oluşan ağ yapısı

Çıkış katmanındaki $N \times Q$ boyutlu matris, yine $N \times Q$ boyutundaki hedeflenen çıkış matrisi ile karşılaştırılarak hata terimi belirlenir. Giriş verileri olarak ağa 21 değer tanıtılmış ve hedeflenen kosinüs işlev yaklaşımı Şekil 3.7'de elde edilmiştir.



Şekil 3.7 – İşlev yaklaşımı

3.6.7. İletim hatlarında taşıma kapasitesinin bulunması

Yüksek gerilim iletim hatlarında taşıma kapasitesi, iletken ısı dayanımı ile ilişkilidir. Yüksek ısı dayanımlı iletken kullanımı ile hava hatlarında taşıma kapasitesi artırılır. Yapay sinir ağına tanıtılacak veriler, iletken sıcaklığının her saat başında deneysel olarak ölçülmesi ile elde edilmiştir. Yapay sinir ağını eğitmek için elde edilen 744 değerden 500'ü kullanılmış, geriye kalan 244 veri de performans değerlendirmesi için test amaçlı kullanılmıştır. Yapay sinir ağına $x_1, x_2, \dots, x_k$ verileri tanıtılmış ve x_{k+1} değeri elde edilmeye çalışılmıştır. Eğitim işlemi giriş verileri sayısı $k = 3, 15, 30$ iken gerçekleştirilmiş ve hata değerleri sırasıyla 0.113, 0.107, 0.110 olarak bulunmuştur. Bu hata değerleri doğrultusunda en iyi performans 15 nöronlu giriş katmanı ile sağlandığından, giriş verileri sayısı 15'e yakın tutulmuş ve Tablo 3.1'de gösterilen hata dağılımı gözlenmiştir [31].

Tablo 3.1 – İletim hatlarında taşıma kapasitesinin bulunmasında geriye yayılım algoritması ile gözlenen hata dağılımı değerleri

	A	B	C	D	E
%	56	30	10	2.5	1.5

A = Hata oranı %5'e eşit veya daha küçük

B = Hata oranı %5'den büyük, %10'dan küçük veya eşit

C = Hata oranı %10'dan büyük, %15'den küçük veya eşit

D = Hata oranı %15'den büyük, %20'den küçük veya eşit

E = Hata oranı %20'den büyük, %25'den küçük veya eşit

3.6.8. Elektrokardiyogram verilerinin tanınması

Kalp ile ilgili hastalıkların tanısında EGK kayıtlarının hızlı yorumlanması önem kazanır. Ancak kalbin durumu hakkında yeterli ve anlamlı bilginin elde edilmesi, uzun kayıtların analizini gerektirir. Analizde harcanan süreyi kısaltmak, sistemde saklanan veri miktarını azaltmak ve bilgisayar destekli teşhise imkan sağlamak, EGK vurularının çeşitli hastalıklar hakkında ön bilgi içeren kategorilere otomatik olarak ayrılması ile mümkün olur.

Çalışmada, EGK vurularını sınıflama işlemi üç aşamada gerçekleştirilmiştir. Normalizasyon işlemi, öznitelik çıkartma işlemi ve sınıflayıcı olarak yapay sinir ağı kullanımı.

EGK vurularını sınıflamada çok katmanlı sinir ağı, geriye yayılım algoritması kullanılarak incelenmiş, hibrit yapılı genetik algoritmalar ile eğitilen kısıtlı Coulomb enerji ağı (GetRCE) ile karşılaştırılmıştır. Sonuçta, geriye yayılım algoritmasının GetRCE ağından yavaş kalmasına rağmen toplam sınıflama başarımı %87,6 gibi yüksek bir oran çıkmıştır [33].



4. YAPAY SİNİR AĞLARININ ELEKTRİKSEL DELİNME DAYANIMI DEĞERLERİNİN BELİRLENMESİNDE KULLANILMASI

4.1. Giriş

Yapay sinir ağlarında, sonuç katmanında tahmin edilmesi istenen çıktılara yönelik ilişkisi olduğu düşünülen her türlü veri girdi olarak kullanılabilir. Bu değerler, birbirlerinden bağımsız değerler olarak kullanılabileceği gibi, birbirlerini destekler bir yapı içinde de kullanılabilirler. Ancak, yapay sinir ağlarında giriş değerleri olarak kullanılacak verileri belirlerken bu verilerin birbirlerini tekrarlar nitelikte olmamasına özellikle dikkat edilmelidir.

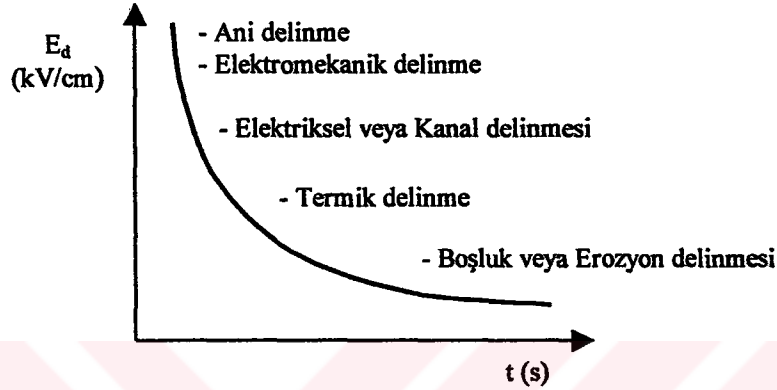
Uygulamada yapay sinir ağına tanıtılacak veriler, deneysel yöntemlerle ölçülmüş delinme dayanımı değerleridir. Deneylerde, yalıtkan malzeme olarak polyster kullanılmıştır. Belirli bazı sıcaklık değerlerinde ve folye kalınlıklarında yapılan deneyler sonucunda elde edilen delinme dayanımı değerleri ağa tanıtılarak farklı sıcaklık değerlerine ve folye kalınlıklarına karşılık gelen delinme dayanımı değerlerini bulmak amaçlanmaktadır.

Tahmin edileceği gibi her ortam koşulunda deneysel veri elde etmek mümkün olmayacağından ekstrapolasyon ve interpolasyon yöntemlerini kullanarak yeni değerler elde etmek, kullanılan klasik yöntemlerdendir. Bu aşamada, yapay sinir ağlarının kullanımı, tutarlı sonuçları ile farklı bir bakış açısı getirmektedir.

4.2. Katı Yalıtkanlarda Delinme Türleri

Katı yalıtkanlarda delinme gerilimi, gerilimin uygulanma süresine bağlıdır. Delinme geriliminin değeri, nano ve mikrosaniyeler mertebesindeki kısa ve orta süreli gerilim uygulamalarında farklılıklar gösterir. Uzun süreli gerilim uygulamalarında delinmeye yeten gerilim, darbe gerilimindeki delinme geriliminin ancak bir kesri kadardır. Eğer yalıtkanın dielektrik kayıpları büyükse, bu fark da büyük olur. Uzun süreli gerilim

uygulamaları durumunda, katı yalıtkanın içinde bulunan boşluklarda devamlı olarak korona kayıpları oluşur. Böylece yalıtkan madde aşınarak delinir. Çok kısa süreli gerilim uygulamalarında ise ani delinme veya elektromekanik delinme gerçekleşir. Şekil 4.1’de katı yalıtkanlarda delinme dayanımının gerilim uygulama süresine göre değişimi gösterilmektedir.



Şekil 4.1 - Katı yalıtkanlarda delinme dayanımının gerilim uygulama süresine göre değişimi

4.3. Polyester Film Tabakalarında Elektriksel Delinme Dayanımı

Delinme dayanımı, yalıtımın ısı, elektriksel ve mekanik yaşlanmasını tanıtan ve yalıtkan malzemeyi veya sistemi karakterize eden temel bir özelliktir. Yalıtkan maddelerin kalitesinin bir ölçütüdür. Bu nedenle, elektriksel yalıtkan maddelerin kısa süreli başarımını değerlendirmek amacıyla, delinme dayanımı ölçmeleri yapılır.

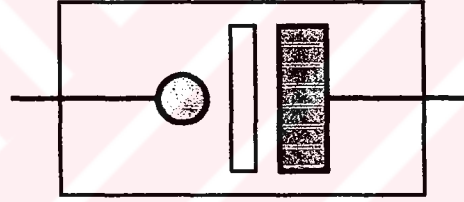
Polyester, iyi dielektrik özelliklerine ve yüksek yüzeysel dayanıma sahip olup kimyasal etkilere karşı dayanıklı bir malzemedir. Dikarboksilik asitlerin ve dihidrik alkollerin yoğunlaştırılması ile elde edilen tüm termoset plastiklerin özelliklerini taşır.

Polyester, doymuş ve doymamış tür olarak iki sınıfa ayrılır. Doymamış polyester, cam tabakalarında veya cam liflerinde takviye malzemesi olarak kullanılır. Bunun dışında, elektriksel ve mekanik dayanımı yeterli küçük elektriksel elemanlar yapmak için yaygın olarak kullanılır. Doymuş polyesterler, fiber ve film yapımında kullanılır. Polyester fiber, elektriksel amaçlı kağıt, halı ve giysi yapımında kullanılmaktadır. Polyester film; motor, kondansatör ve transformatörlerdeki kablo ve yalıtılmış tellerde kullanılır. Polyester film, çoğunlukla kağıt yapımında kullanılmaktadır. Endüstriyel frekanslarda, polyesterin dielektrik kaybı çok düşüktür. Polyesterin dielektrik kayıpları, sıcaklık arttıkça azalır. Yüksek yumuşama sıcaklığı nedeniyle,

kağıt yalıtımın çalışma sınırının üstündeki sıcaklıklarda kullanılmasını mümkün kılar. Polyesterin hava koşullarına karşı direnci yüksektir [36].

4.4. Giriş Değerleri Olarak Sinir Ağına Tanıtılacak Deneysel Verilerin Elde Edilmesi

Deneyde, polyester malzemeden, kalmınlıkları 0.012, 0.023, 0.036, 0.050, 0.100 ve 0.200 mm ve kenar uzunlukları 100 mm olan kare şeklinde numuneler hazırlanmıştır. Deneyde kullanılan elektrot sistemi, küre - düzlem elektrot sistemidir. Küre - düzlem elektrot sisteminde, 20 mm çapında bir küre elektrot ile 50 mm genişliğinde disk şeklinde bir düzlem elektrot kullanılmıştır. Elektrot sistemi, Şekil 4.2’de gösterilmiştir. Elektrotları ve numuneyi çevreleyen ortam, yani daldırma ortamı havadır [36].



Şekil 4.2. – Deneyde kullanılan elektrot sistemi

Farklı kalmınlıklardaki polyester folye numuneleri için hava ortamında delinme dayanımının sıcaklıkla değişimi Tablo 4.1’de gösterilmiştir.

Tablo 4.1. – Farklı kalınlıklarda polyester folyeler için delinme dayanımının sıcaklıkla değişimi (E_d kV/mm)

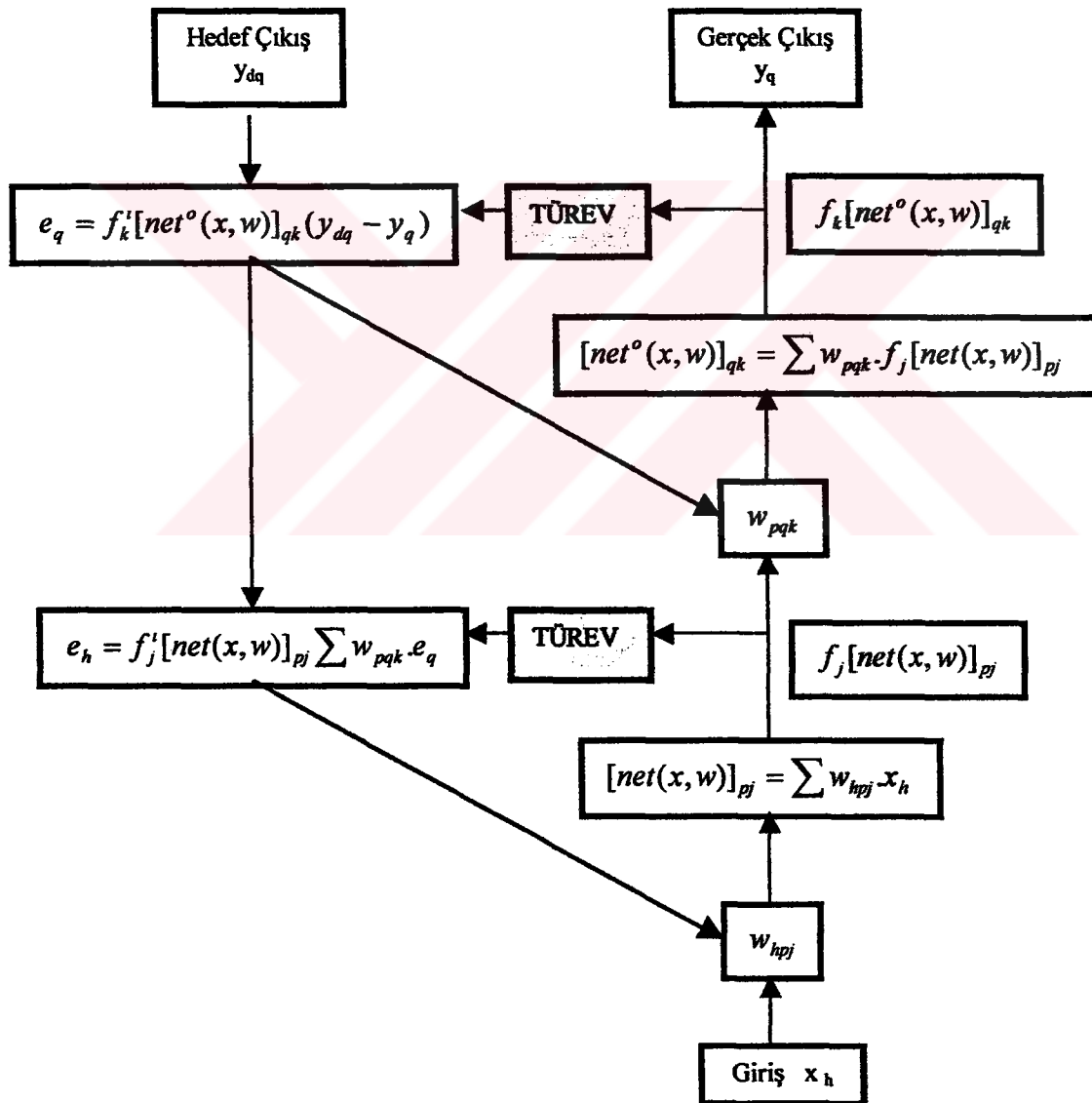
$t^{\circ}\text{C}$	-40°C	-20°C	0°C	20°C	40°C	60°C	80°C	100°C	120°C
12 μm	345	362	370	360	340	332	310	292	270
23 μm	310	318	325	320	305	294	280	260	241
36 μm	245	255	270	260	240	230	212	190	173
50 μm	220	230	242	230	210	202	189	171	154
100 μm	175	182	195	180	170	158	145	130	116
200 μm	160	165	170	160	155	145	135	122	109

Tablo 4.1’de elde edilen sayısal değerlerin geriye yayılım algoritması kullanılarak sinir ağına tanıtılması ve uygulama sonuçları Bölüm 5’de ayrıntılı olarak açıklanmıştır.

5. BİLGİSAYAR UYGULAMASI VE SONUÇLARI

5.1. Uygulama

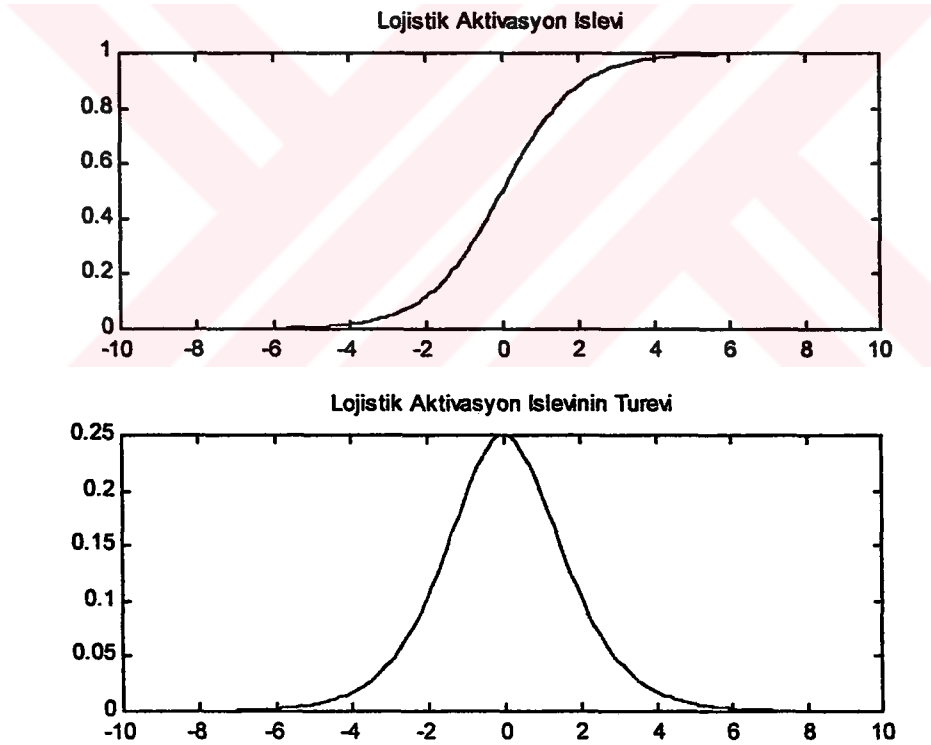
Uygulamada kullanılan geriye yayılım algoritmasının akış diyagramı şekil 5.1’de gösterilmiştir.



Şekil 5.1 – BP algoritması akış şeması

Bu alıřmadaki bilgisayar uygulaması Matlab Neural Network Toolbox kullanılarak yapılmıřtır. Hazırlanan bilgisayar programında, lojistik sigmoid aktivasyon iřlevi kullanılmıřtır (řekil 5.2). Lojistik sigmoid aktivasyon iřlevi trevinin maksimum noktası, giriř verilerinin sıfıra eřitlendiėi noktadır [29].

```
x = [-10:1:10]
y = 1./(1+exp(-x));
dy = y.*(1-y);
subplot (2,1,1)
plot (x,y)
title ('Lojistik Aktivasyon İřlevi')
subplot (2,1,2)
plot (x,dy)
title ('Lojistik Aktivasyon İřlevinin Trevi')
```



řekil 5.2. – Lojistik sigmoid aktivasyon iřlevi ve trevi

Kurulan Matlab programı üzerinde saklı katman düğüm sayısı, öğrenme oranı ve hata işlevi değerleri ile değişik parametrelerle çok sayıda denemeler ve hesaplamalar yapılmıştır.

Saklı katman düğüm sayısı ve ağı katman sayısı uygulamada optimal değerde olmalıdır. Belirli değerin üstündeki düğüm ve tabaka sayıları, ağı kırılğan hale sokmaktadır.

Ağ performansı, öğrenme oranının doğru seçilmesi ile ilişkilidir. Eğer öğrenme oranı çok büyük seçilirse, sonuca yakınsama sağlanamaz.

Hata değerinin seçimi, hedeflenen doğrulukla ilişkilidir. Hata değeri küçüldükçe daha doğru çözümlere ulaşılmaktadır. Hata değerinin küçük seçilmesi, eğitim süresini ve iterasyon sayısını artırır.

Eğer öğrenme işlemi yapay sinir ağı tarafından gerçekleştirilemiyorsa, ya eğitim süresi gerektiğinden kısa tutulmuş ya da saklı katmandaki nöron sayısı optimum seviyede seçilmemiştir.

Aktivasyon işlevinin türevinin alınabilir olması gerekliliği unutulmamalıdır. Yapay sinir ağlarının, öğrenme işlemini gerçekleyebilmesi için çıkış verilerinin aynı çıkışa denk düşen birden fazla giriş değeri ile eşleştirilmemesi gerekmektedir.

5.1.1. Hata limit değerinin değiştirilmesi ile gözlenen değişiklikler

Uygulamada, tek saklı katmanlı sinir ağında saklı katman nöron sayısı 3 alınarak hata limit değerindeki değişikliklere göre öğrenme süreci gözlenmiştir.

Şekil 5.3'den Şekil 5.22'ye kadar olan şekiller 12 μ m kalınlıktaki polyester folyenin -40°C, -20°C, 0°C, 20°C, 40°C, 60°C, 80°C, 100°C, 120°C sıcaklık değerlerinde ölçülen delinme dayanımı değerlerinin sinir ağına tanıtılmasıyla yapılan uygulamalara ait grafiklerdir.

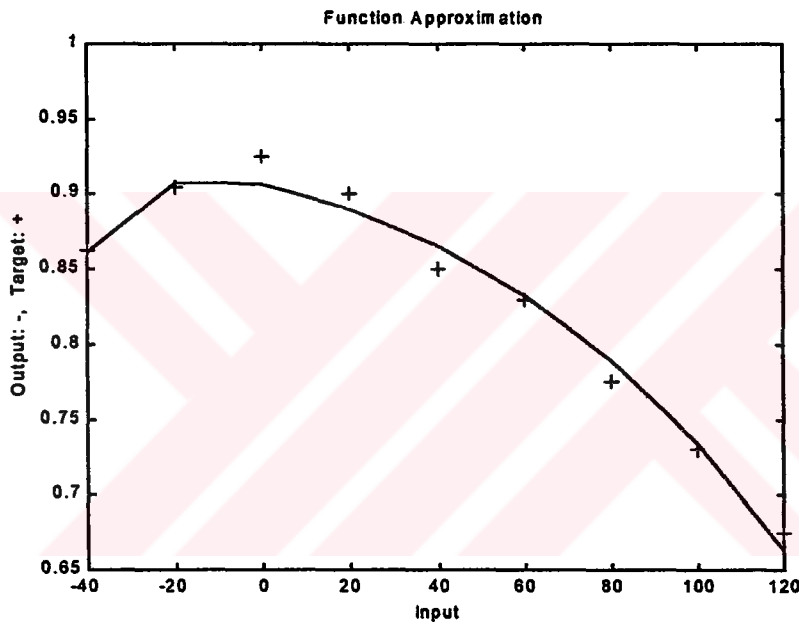
KOMUT DİZGİSİ

clf reset	%clear the screen
H=3;	%number of neurons in hidden layer (saklı katmandaki nöron sayısı)
D=40;	%display period (periyot)
R=1000;	%number of training iterations (eğitme döngüsü sayısı)
E=0.001;	%desired error (hedeflenen hata oranı)
lr=0.001;	%learning rate (öğrenme katsayısı)

```

m=0.8; %momentum
I=1:.01:1;
P=[-40 -20 0 20 40 60 80 100 120];
T=[0.8625 0.905 0.925 0.9 0.85 0.83 0.775 0.73 0.675];
W1 = randn(3,9);
B1 = randn(3,1);
W2 = randn(9,3);
B2 = randn(9,1);
[W1,B1,W2,B2] = initff(P,H,'logsig',T,'logsig');
[W1,B1,W2,B2,iter,errors]=trainbp(W1,B1,'logsig',W2,B2,'purelin',P,T,[D R E lr m]);
pause
plotfa(P,T,I,purelin(W2*tansig(W1*I,B1),B2));
pause
cls
P=[ -30 -10 5 25 35 50 70 90 110]
A=simuff(W1,B1,'logsig',W2,B2,'logsig',P,T,[D R E lr m])
plot
end

```

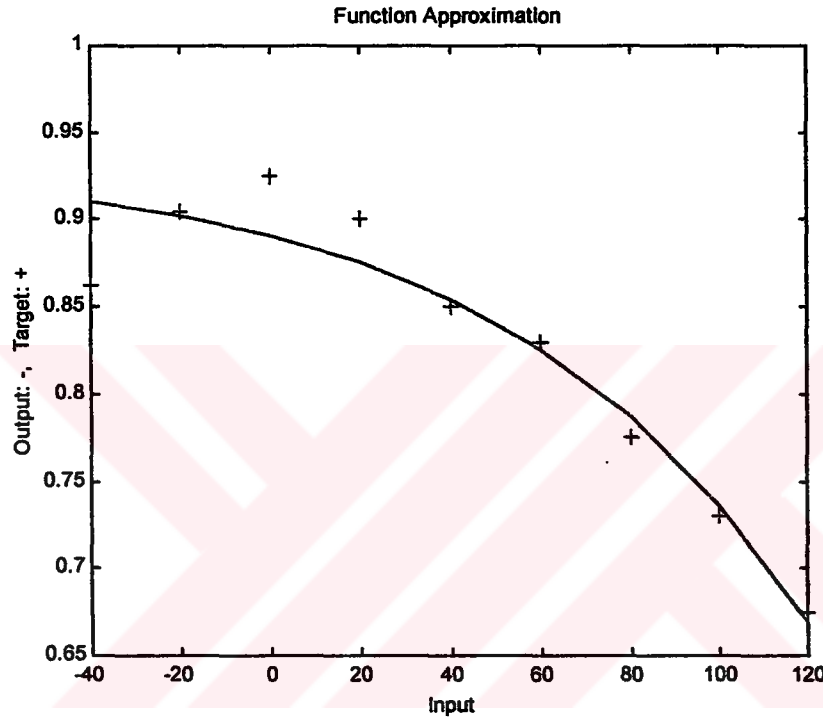


Şekil 5.3 – 3 nöronlu 1 saklı katmanlı sinir ağında işlev yaklaşımı şekli (E=0.001)

TRAINBP: 0/1000 epochs, SSE = 167.648.
 TRAINBP: 40/1000 epochs, SSE = 5.87962.
 TRAINBP: 80/1000 epochs, SSE = 1.85964.
 TRAINBP: 120/1000 epochs, SSE = 0.656232.
 TRAINBP: 160/1000 epochs, SSE = 0.0203227.
 TRAINBP: 200/1000 epochs, SSE = 0.0203819.
 TRAINBP: 240/1000 epochs, SSE = 0.0107531.
 TRAINBP: 280/1000 epochs, SSE = 0.00694912.
 TRAINBP: 320/1000 epochs, SSE = 0.00459729.
 TRAINBP: 360/1000 epochs, SSE = 0.00308007.
 TRAINBP: 400/1000 epochs, SSE = 0.00209666.
 TRAINBP: 440/1000 epochs, SSE = 0.00150575.
 TRAINBP: 480/1000 epochs, SSE = 0.00123189.
 TRAINBP: 520/1000 epochs, SSE = 0.00114893.
 TRAINBP: 560/1000 epochs, SSE = 0.00112276.
 TRAINBP: 600/1000 epochs, SSE = 0.00110807.
 TRAINBP: 640/1000 epochs, SSE = 0.00109821.
 TRAINBP: 680/1000 epochs, SSE = 0.00109144.
 TRAINBP: 720/1000 epochs, SSE = 0.00108677.

TRAINBP: 760/1000 epochs, SSE = 0.0010835.
 TRAINBP: 800/1000 epochs, SSE = 0.00108122.
 TRAINBP: 840/1000 epochs, SSE = 0.0010796.
 TRAINBP: 880/1000 epochs, SSE = 0.00107845.
 TRAINBP: 920/1000 epochs, SSE = 0.00107763.
 TRAINBP: 960/1000 epochs, SSE = 0.00107704.
 TRAINBP: 1000/1000 epochs, SSE = 0.00107661.

Saklı katmandaki nöron sayısı 3 iken yapılan bir başka denemede ise aşağıdaki sonuç elde edilmiştir.



Şekil 5.4 - 3 nöronlu 1 saklı katmanlı sinir ağında işlev yaklaşımı şekli (E=0.004)

TRAINBP: 0/1000 epochs, SSE = 12.2574.
 TRAINBP: 400/1000 epochs, SSE = 0.0123829.
 TRAINBP: 1000/1000 epochs, SSE = 0.00435406.

Hata limit değeri arttırıldığında ağın öğrenme duyarlılığının azaldığı ve sonuçların tutarlılıktan uzaklaştığı gözlenmektedir.

KOMUT DİZGİSİ

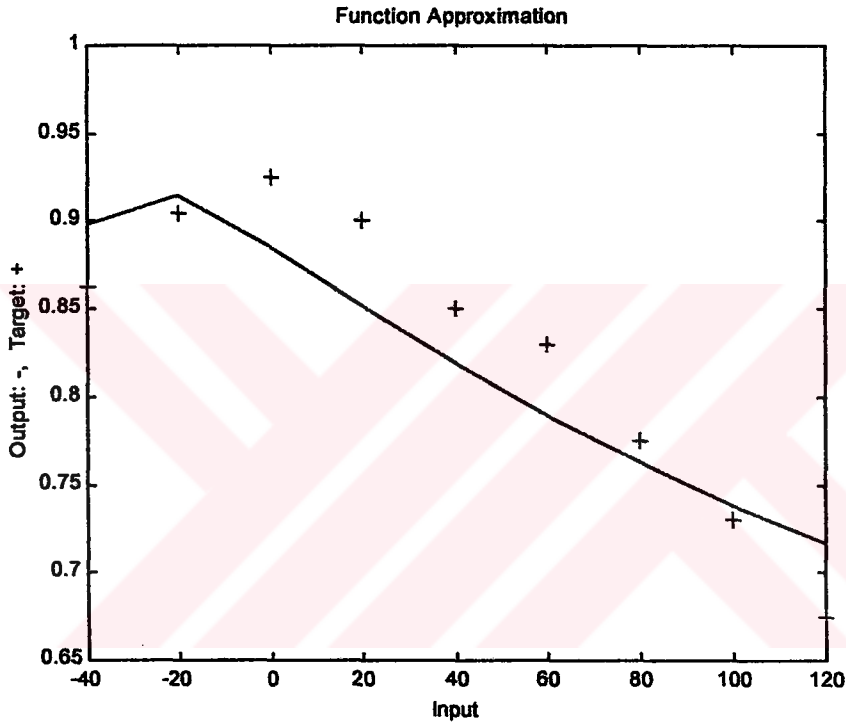
```

clf reset           %clear the screen
H=3;               %number of neurons in hidden layer
D=60;              %display period
R=1000;            %number of training iterations
E=0.01;            %desired error
lr=0.001;          %learning rate
m=0.8;             %momentum
I=-1:.01:1;
P=[-40 -20 0 20 40 60 80 100 120];
T=[0.8625 0.905 0.925 0.9 0.85 0.83 0.775 0.73 0.675];
  
```

```

W1 = randn(3,9);
B1 = randn(3,1);
W2 = randn(9,3);
B2 = randn(9,1);
[W1,B1,W2,B2] = initff(P,H,'logsig',T,'logsig');
[W1,B1,W2,B2,iter,errors]=trainbp(W1,B1,'logsig',W2,B2,'purelin',P,T,[D R E lr m]);
pause
plotfa(P,T,I,purelin(W2*tansig(W1*I,B1),B2));
pause
cls
P=[ -30 -10 5 25 35 50 70 90 110]
A=simuff(W1,B1,'logsig',W2,B2,'logsig',P,T,[D R E lr m])
plot
end

```



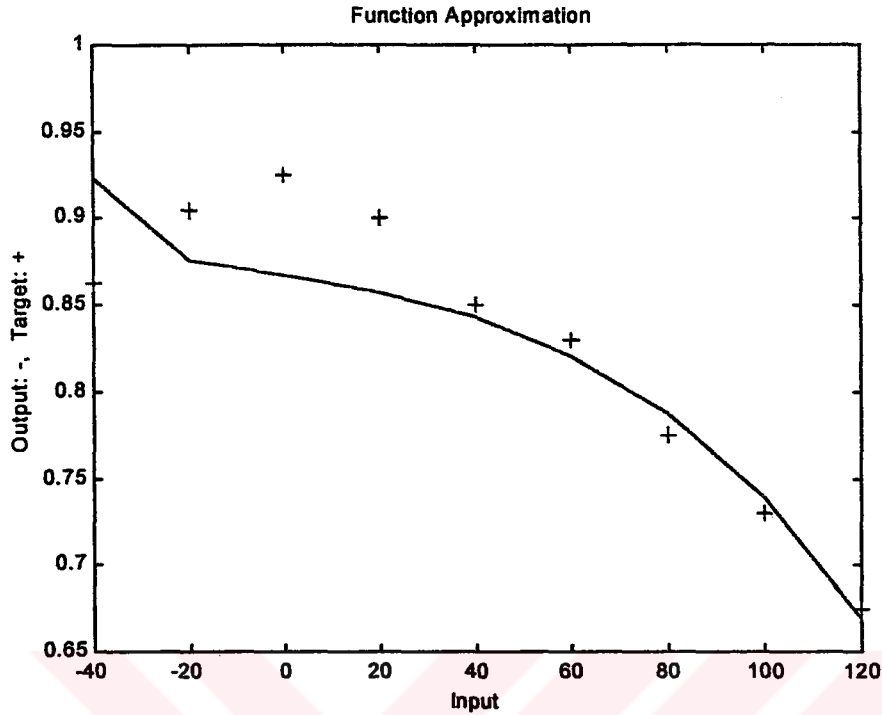
Şekil 5.5 - 3 nöronlu 1 saklı katmanlı sinir ağında işlev yaklaşımı şekli (E=0.01)

```

TRAINBP: 0/1000 epochs, SSE = 13.8829.
TRAINBP: 60/1000 epochs, SSE = 0.0788921.
TRAINBP: 120/1000 epochs, SSE = 0.0431077.
TRAINBP: 180/1000 epochs, SSE = 0.0414405.
TRAINBP: 240/1000 epochs, SSE = 0.0405524.
TRAINBP: 300/1000 epochs, SSE = 0.0395692.
TRAINBP: 360/1000 epochs, SSE = 0.0384505.
TRAINBP: 420/1000 epochs, SSE = 0.03717.
TRAINBP: 480/1000 epochs, SSE = 0.0356972.
TRAINBP: 540/1000 epochs, SSE = 0.0339988.
TRAINBP: 600/1000 epochs, SSE = 0.0320488.
TRAINBP: 660/1000 epochs, SSE = 0.0298496.
TRAINBP: 720/1000 epochs, SSE = 0.0274289.
TRAINBP: 780/1000 epochs, SSE = 0.0244027.
TRAINBP: 840/1000 epochs, SSE = 0.0155408.
TRAINBP: 866/1000 epochs, SSE = 0.00981096.

```

Aynı parametrelerle yapılan bir başka denemede de aşağıdaki sonuçlar gözlenmiştir.



Şekil 5.6 - 3 nöronlu 1 saklı katmanlı sinir ağında işlev yaklaşımı şekli ($E=0.01$)

TRAINBP: 0/1000 epochs, SSE = 26.2993.
 TRAINBP: 100/1000 epochs, SSE = 0.0878368.
 TRAINBP: 200/1000 epochs, SSE = 0.0699023.
 TRAINBP: 300/1000 epochs, SSE = 0.0687911.
 TRAINBP: 400/1000 epochs, SSE = 0.0677951.
 TRAINBP: 500/1000 epochs, SSE = 0.0659788.
 TRAINBP: 600/1000 epochs, SSE = 0.0240314.
 TRAINBP: 680/1000 epochs, SSE = 0.0102512.
 TRAINBP: 685/1000 epochs, SSE = 0.00998924.

Şekil 5.6'da her ne kadar hata limit değerine ulaşılmış olsa da öğrenme sonucu yeterince iyi düzeyde değildir. Bunun telafi edilebilmesi için ağ döngü sayısı arttırılacak ve hata limit değeri küçültülecektir. Elde edilen sonuçlar aşağıda verilmiştir.

KOMUT DİZGİSİ

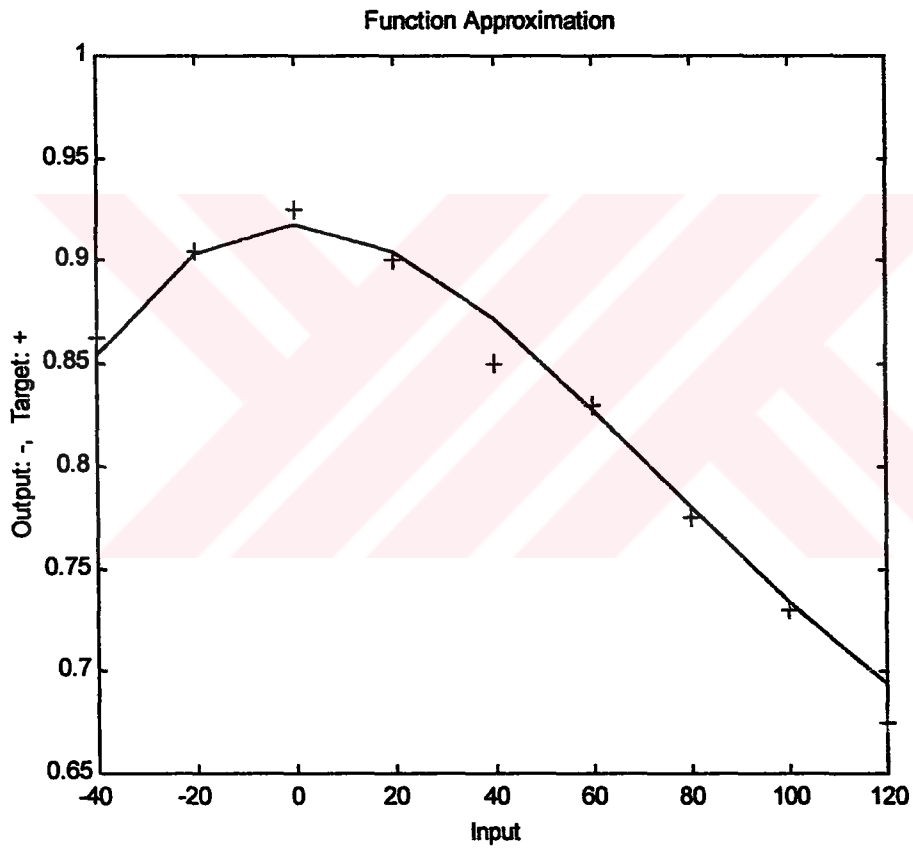
```

clf reset                %clear the screen
H=3;                    %number of neurons in hidden layer
D=20;                   %display period
R=150000;               %number of training iterations
E=0.001;                %desired error
lr=0.001;               %learning rate
m=0.8;                  %momentum
I=-1:0.01:1;
P=[-40 -20 0 20 40 60 80 100 120];
T=[0.8625 0.905 0.925 0.9 0.85 0.83 0.775 0.73 0.675];
W1 = randn(3,9);
  
```

```

B1 = randn(3,1);
W2 = randn(9,3);
B2 = randn(9,1);
[W1,B1,W2,B2] = initff(P,H,'logsig',T,'logsig');
[W1,B1,W2,B2,iter,errors]=trainbp(W1,B1,'logsig',W2,B2,'purelin',P,T,[D R E lr m]);
pause
plotfa(P,T,I,purelin(W2*tansig(W1*I,B1),B2));
pause
cls
P=[ -30 -10 5 25 35 50 70 90 110]
A=simuff(W1,B1,'logsig',W2,B2,'logsig',P,T,[D R E lr m])
plot
end

```



Şekil 5.7 –3 nöronlu 1 saklı katmanlı sinir ağında işlev yaklaşımı şekli
(E=0.001, R=150000)

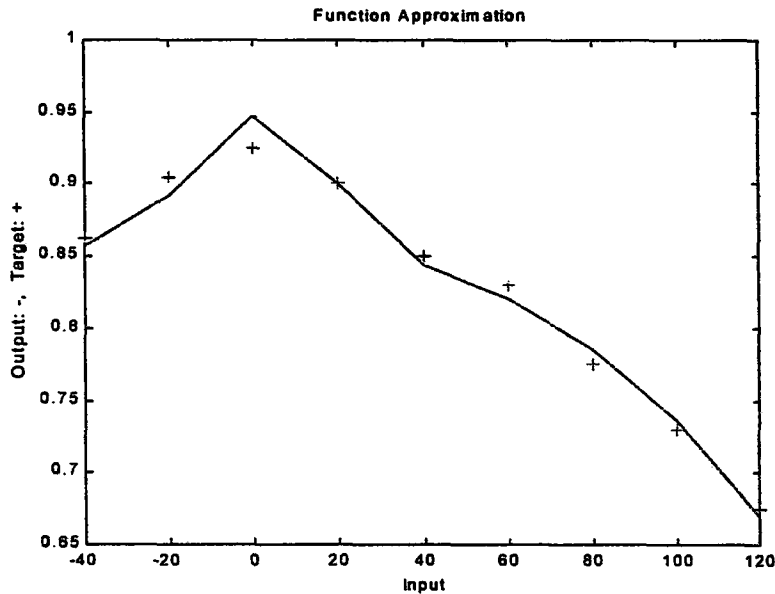
TRAINBP: 150000/150000 epochs, SSE = 0.00101.

5.1.2. Saklı katmandaki nöron sayısı artırılarak yapılan denemeler.

Uygulamada saklı katman nöron sayısı 3, 5, 7 ve 10 alınarak işlev yaklaşımı grafikleri çıkartılmış ve optimum saklı katman nöron sayısının 10 olması gerektiğine karar verilmiştir. Uygulama sonuçları aşağıda gösterilmiştir.

KOMUT DİZGİSİ

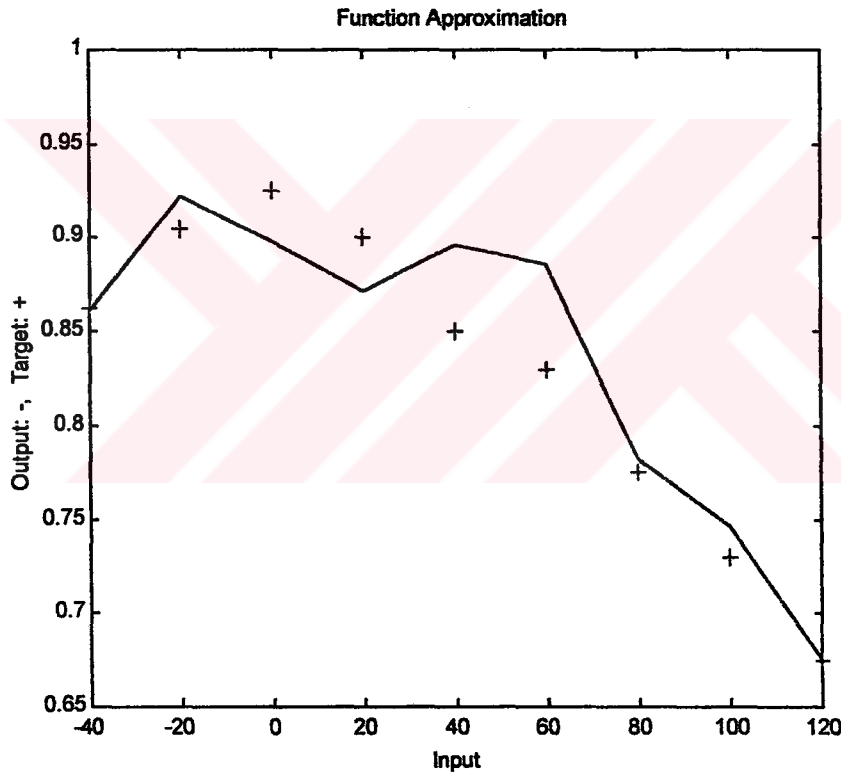
```
clf reset           %clear the screen
H=5;               %number of neurons in hidden layer
D=1000;            %display period
R=150000;          %number of training iterations
E=0.001;           %desired error
lr=0.001;          %learning rate
m=0.8;             %momentum
I=-1:0.01:1;
P=[-40 -20 0 20 40 60 80 100 120];
T=[0.8625 0.905 0.925 0.9 0.85 0.83 0.775 0.73 0.675];
W1 = randn(5,9);
B1 = randn(5,1);
W2 = randn(9,5);
B2 = randn(9,1);
[W1,B1,W2,B2] = initff(P,H,'logsig',T,'logsig');
[W1,B1,W2,B2,iter,errors]=trainbp(W1,B1,'logsig',W2,B2,'purelin',P,T,[D R E lr m]);
pause
plotfa(P,T,I,purelin(W2*tansig(W1*I,B1),B2));
pause
cls
P=[ -30 -10 5 25 35 50 70 90 110]
A=simuff[W1,B1,'logsig',W2,B2,'logsig',P,T,[D R E lr m]]
plot
end
```



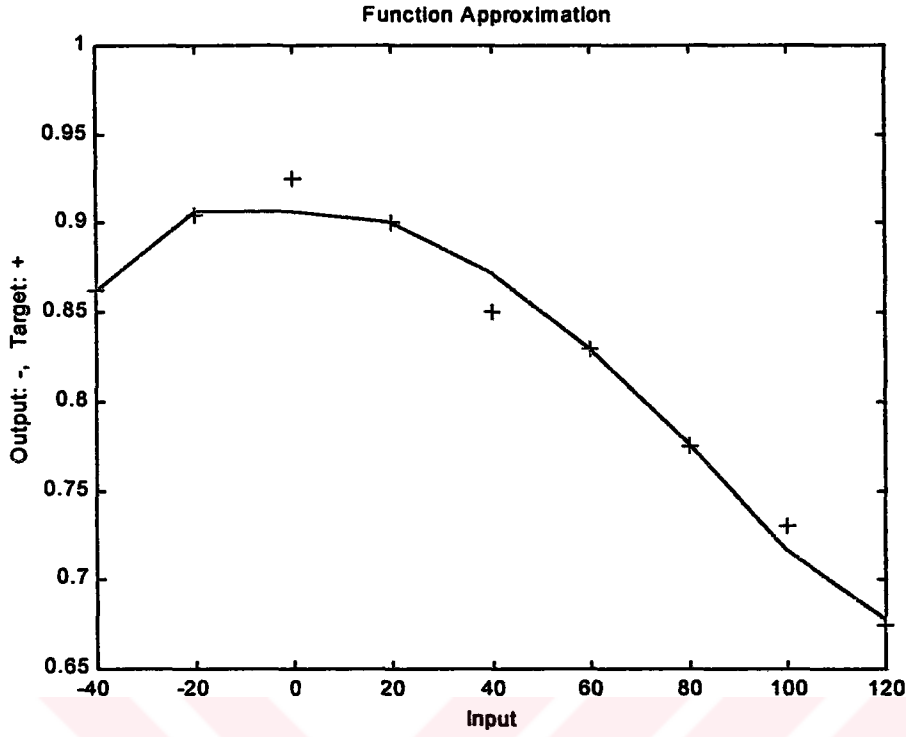
Şekil 5.8 – 5 nöronlu 1 saklı katmanlı sinir ağında işlev yaklaşımı şekli
(E=0.001, R=150000)

TRAINBP: 0/150000 epochs, SSE = 58.8115.
 TRAINBP: 1000/150000 epochs, SSE = 3.06239.
 TRAINBP: 2000/150000 epochs, SSE = 1.28732.
 TRAINBP: 3000/150000 epochs, SSE = 0.555991.
 TRAINBP: 4000/150000 epochs, SSE = 0.240499.
 TRAINBP: 5000/150000 epochs, SSE = 0.104121.
 TRAINBP: 6000/150000 epochs, SSE = 0.0451957.
 TRAINBP: 7000/150000 epochs, SSE = 0.0197694.
 TRAINBP: 8000/150000 epochs, SSE = 0.00882857.
 TRAINBP: 9000/150000 epochs, SSE = 0.00414404.
 TRAINBP: 10000/150000 epochs, SSE = 0.00215092.
 TRAINBP: 11000/150000 epochs, SSE = 0.00130691.
 TRAINBP: 11794/150000 epochs, SSE = 0.000999856.

Saklı katmandaki nöron sayısı 7 ve 10 iken elde edilen işlev yaklaşımı ise sırasıyla Şekil 5.9'da ve Şekil 5.10'da gösterilmiştir.



Şekil 5.9 – 7 nöronlu 1 saklı katmanlı sinir ağında işlev yaklaşımı şekli (E=0.001, R=150000)



Şekil 5.10 – 10 nöronlu 1 saklı katmanlı sinir ağında işlev yaklaşımı şekli
($E=0.001$, $R=150000$)

TRAINBP: 0/150000 epochs, SSE = 134.284.
 TRAINBP: 1000/150000 epochs, SSE = 0.0590433.
 TRAINBP: 1364/150000 epochs, SSE = 0.000999724.

5.1.3. Momentum değerinin arttırılması ile elde edilen sonuçlar

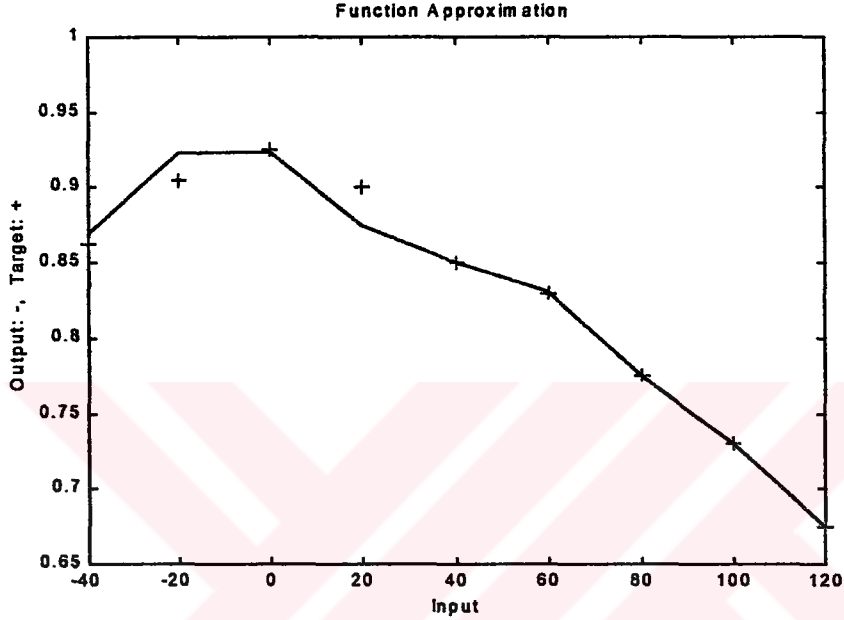
Momentum değerinin arttırılması ile daha kısa sürede hata limitine ulaşıldığı gözlenmiştir. Denemeler tek saklı katmanda 10 nöron olması durumunda momentum değeri 0.2, 0.4, 0.6, 0.8, 0.9 ve 0.99 değerleri için yapılacaktır. Şekil 5.11’de momentum değeri 0.2 seçilmiştir. Eğitim sonuçları aşağıda gösterilmiştir

```
clf reset           %clear the screen
H=10;              %number of neurons in hidden layer
D=1000;            %display period
R=150000;          %number of training iterations
E=0.001;           %desired error
lr=0.001;          %learning rate
m=0.2;             %momentum
I=-1:.01:1;
P=[-40 -20 0 20 40 60 80 100 120];
T=[0.8625 0.905 0.925 0.9 0.85 0.83 0.775 0.73 0.675];
W1 = randn(10,10);
B1 = randn(10,1);
W2 = randn(10,10);
```

```

B2 = randn(10,1);
[W1,B1,W2,B2] = initff(P,H,'logsig',T,'logsig');
[W1,B1,W2,B2,iter,errors]=trainbp(W1,B1,'logsig',W2,B2,'purelin',P,T,[D R E l r m]);
pause
plotfa(P,T,I,purelin(W2*tansig(W1*I,B1),B2));
pause
cls
P=[ -30 -10 5 25 35 50 70 90 110]
A=simuff[W1,B1,'logsig',W2,B2,'logsig',P,T,[D R E l r m]]

```

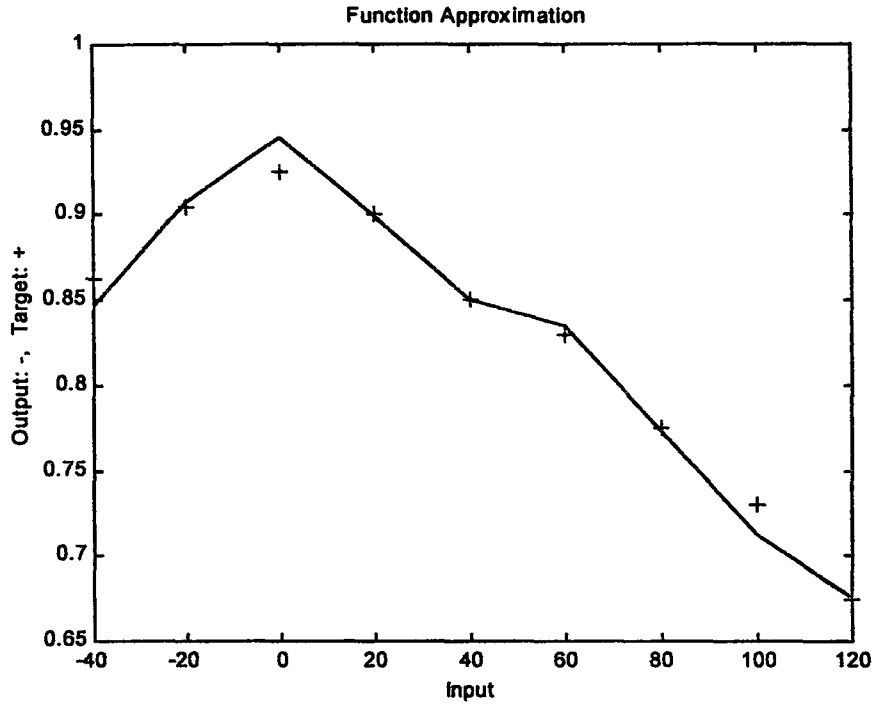


Şekil 5.11 – Saklı katman nöron sayısı 10, $\eta=0.001$ ve $m=0.2$ iken elde edilen eğitim sonuçları

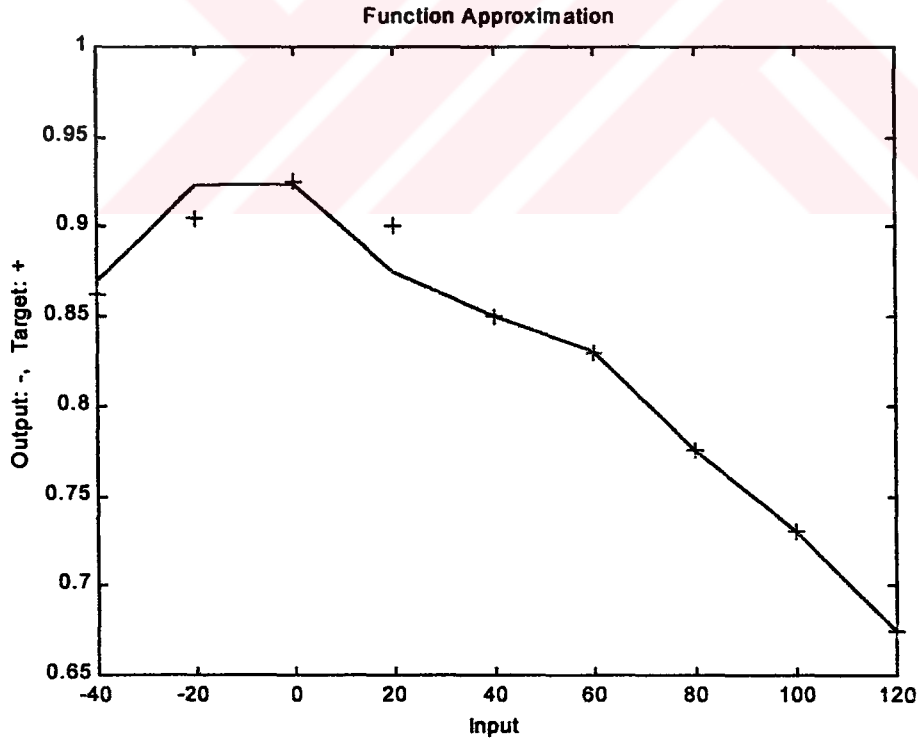
TRAINBP: 0/150000 epochs, SSE = 176.818.
 TRAINBP: 1000/150000 epochs, SSE = 0.00194258.
 TRAINBP: 1804/150000 epochs, SSE = 0.00099936.

Şekil 5.12’de momentum değeri 0.4 iken elde edilen eğitim sonuçları gösterilmektedir. Bu denemede 1705. döngüde hedeflenen hata limit değerine ulaşılmıştır.

TRAINBP: 0/150000 epochs, SSE = 145.901.
 TRAINBP: 1000/150000 epochs, SSE = 0.00479912.
 TRAINBP: 1705/150000 epochs, SSE = 0.000998833.



Şekil 5.12 – Saklı katman nöron sayısı 10, $\eta=0.001$ ve $m=0.4$ iken elde edilen eğitim sonuçları



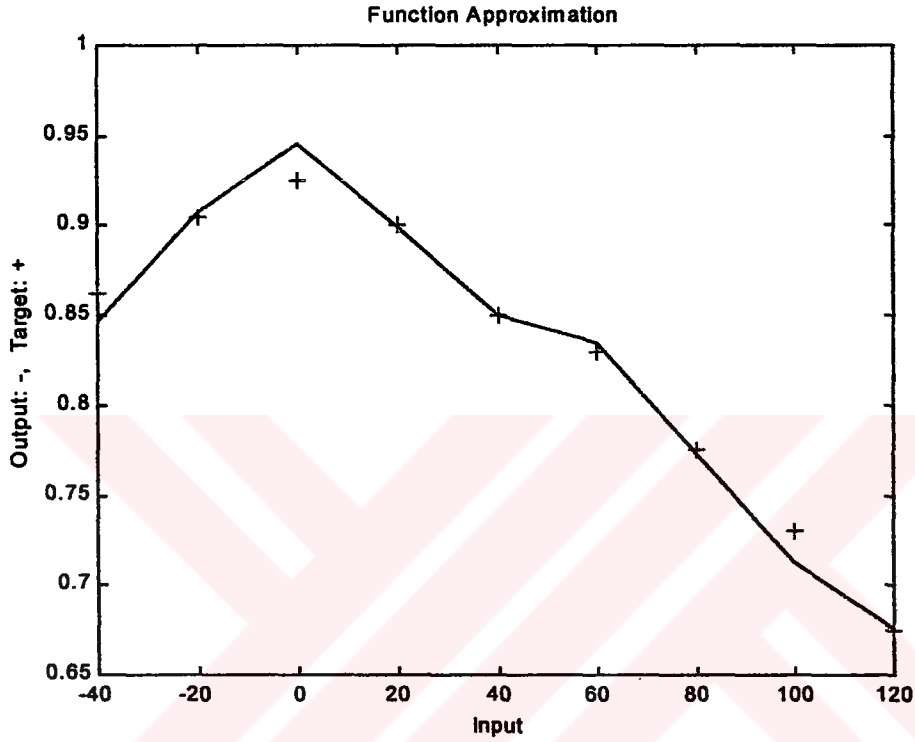
Şekil 5.13 – Saklı katman nöron sayısı 10, $\eta=0.001$ ve $m=0.6$ iken elde edilen eğitim sonuçları

TRAINBP: 0/150000 epochs, SSE = 176.818.

TRAINBP: 1000/150000 epochs, SSE = 0.00194258.

TRAINBP: 1404/150000 epochs, SSE = 0.00099936.

Şekil 5.13’de momentum değeri 0.6 iken elde edilen eğitim sonuçları gösterilmektedir. Bu denemede 1404. döngüde hedeflenen hata limit değerine ulaşılmıştır. Şekil 5.10’da ise momentum katsayısı 0.8 seçilmiş ve hata limit değerine 1364. döngüde ulaşılmıştır. Dikkat edilirse momentum değeri arttıkça hata limitine ulaşma süresinin de kısaldığı görülebilir.

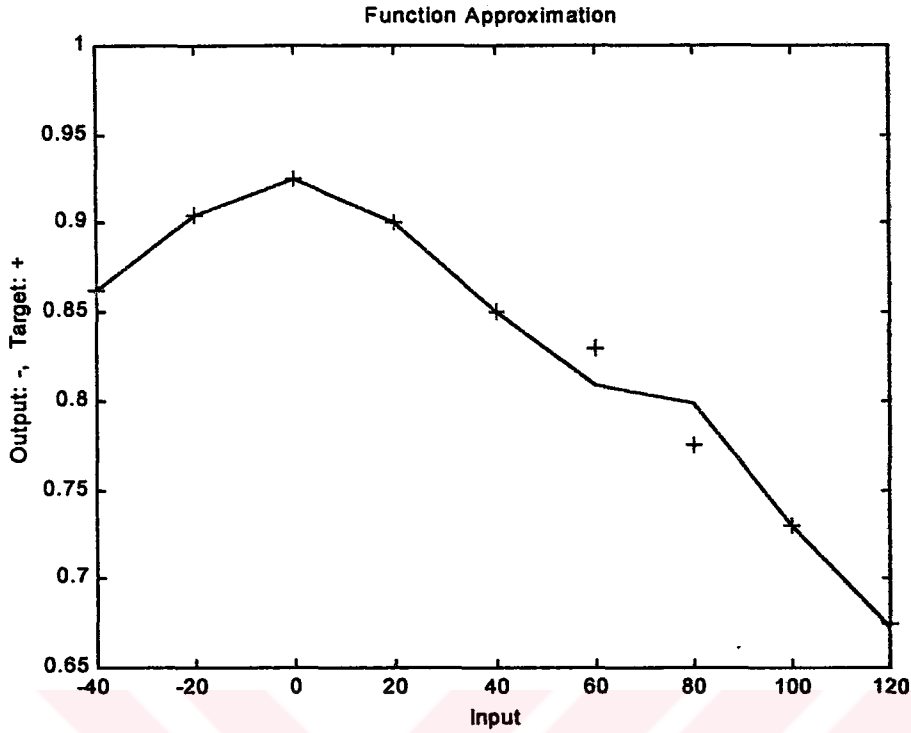


Şekil 5.14 – Saklı katman nöron sayısı 10, $\eta=0.001$ ve $m=0.9$ iken elde edilen eğitim sonuçları

TRAINBP: 0/150000 epochs, SSE = 145.901.
 TRAINBP: 1000/150000 epochs, SSE = 0.00479912.
 TRAINBP: 1705/150000 epochs, SSE = 0.000998833.

Şekil 5.14’de momentum katsayısı 0.9 seçilmiş ve hata limit değerine 1705. döngüde ulaşılmıştır. Şekil 5.15’de de momentum katsayısı 0.99 seçilmiş ve hata limitine 2535. döngüde ulaşılmıştır.

TRAINBP: 0/150000 epochs, SSE = 112.163.
 TRAINBP: 1000/150000 epochs, SSE = 0.00375604.
 TRAINBP: 2000/150000 epochs, SSE = 0.00122515.
 TRAINBP: 2535/150000 epochs, SSE = 0.000999779.



Şekil 5.15 – Saklı katman nöron sayısı 10, $\eta=0.001$ ve $m=0.99$ iken elde edilen eğitim sonuçları

Elde edilen sonuçlara göre hata limitine ulaşma süresinin, momentum katsayısının 0.8 değerine kadar azaldığı bu değerden sonra ise tekrar artma eğilimi gösterdiği gözlenmiştir. Bu uygulama için momentum katsayısının en ideal değeri 0.8 olarak belirlenmiştir.

5.1.4. Öğrenme katsayısının arttırılması ile elde edilen sonuçlar

Öğrenme katsayısının 0.1 seçilmesi durumunda;

TRAINBP: 0/1000 epochs, SSE = 96.8187.
TRAINBP: 200/1000 epochs, SSE = 1.74316e+169.
TRAINBP: 400/1000 epochs, SSE = Inf.
TRAINBP: 600/1000 epochs, SSE = NaN.
TRAINBP: 800/1000 epochs, SSE = NaN.
TRAINBP: 1000/1000 epochs, SSE = NaN.

Öğrenme katsayısının 0.2 seçilmesi durumunda;

TRAINBP: 0/1000 epochs, SSE = 74.4898.
TRAINBP: 200/1000 epochs, SSE = Inf.
TRAINBP: 400/1000 epochs, SSE = NaN.
TRAINBP: 600/1000 epochs, SSE = NaN.
TRAINBP: 800/1000 epochs, SSE = NaN.
TRAINBP: 1000/1000 epochs, SSE = NaN.

Öğrenme katsayısının 0.3 seçilmesi durumunda;

TRAINBP: 0/1000 epochs, SSE = 66.5052.
TRAINBP: 200/1000 epochs, SSE = Inf.
TRAINBP: 400/1000 epochs, SSE = NaN.
TRAINBP: 600/1000 epochs, SSE = NaN.
TRAINBP: 800/1000 epochs, SSE = NaN.
TRAINBP: 1000/1000 epochs, SSE = NaN.

Öğrenme katsayısının 0.4 seçilmesi durumunda;

TRAINBP: 0/1000 epochs, SSE = 145.901.
TRAINBP: 200/1000 epochs, SSE = Inf.
TRAINBP: 400/1000 epochs, SSE = NaN.
TRAINBP: 600/1000 epochs, SSE = NaN.
TRAINBP: 800/1000 epochs, SSE = NaN.
TRAINBP: 1000/1000 epochs, SSE = NaN.

Öğrenme katsayısının 0.5 seçilmesi durumunda;

TRAINBP: 0/1000 epochs, SSE = 158.363.
TRAINBP: 200/1000 epochs, SSE = Inf.
TRAINBP: 400/1000 epochs, SSE = NaN.
TRAINBP: 600/1000 epochs, SSE = NaN.
TRAINBP: 800/1000 epochs, SSE = NaN.
TRAINBP: 1000/1000 epochs, SSE = NaN.

Öğrenme katsayısının 0.6 seçilmesi durumunda;

TRAINBP: 0/1000 epochs, SSE = 112.163.
TRAINBP: 200/1000 epochs, SSE = Inf.
TRAINBP: 400/1000 epochs, SSE = NaN.
TRAINBP: 600/1000 epochs, SSE = NaN.
TRAINBP: 800/1000 epochs, SSE = NaN.
TRAINBP: 1000/1000 epochs, SSE = NaN.

Öğrenme katsayısının 0.7 seçilmesi durumunda;

TRAINBP: 0/1000 epochs, SSE = 136.627.
TRAINBP: 200/1000 epochs, SSE = NaN.
TRAINBP: 400/1000 epochs, SSE = NaN.
TRAINBP: 600/1000 epochs, SSE = NaN.
TRAINBP: 800/1000 epochs, SSE = NaN.
TRAINBP: 1000/1000 epochs, SSE = NaN.

Öğrenme katsayısının 0.8 seçilmesi durumunda;

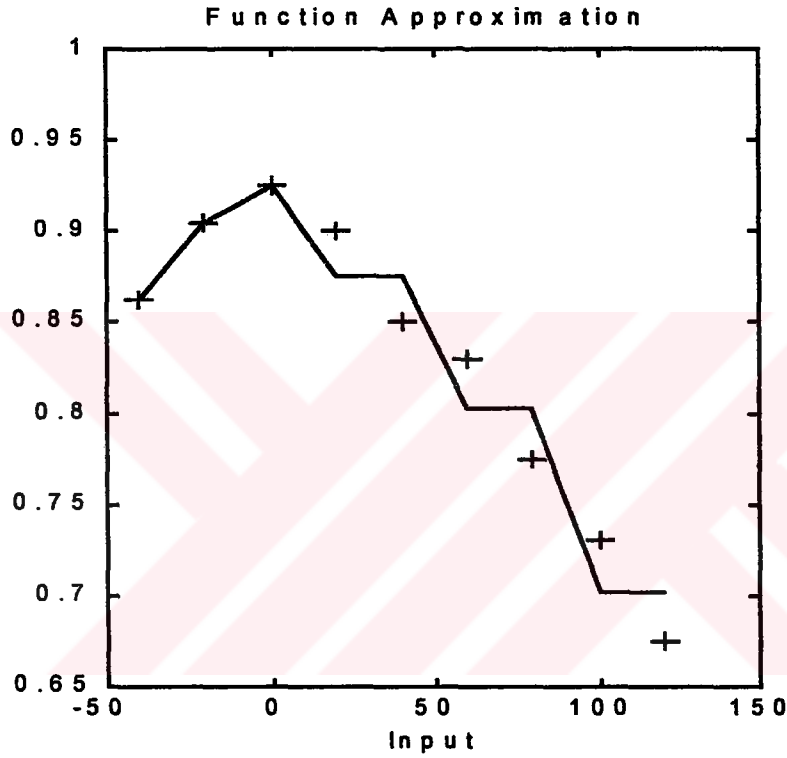
TRAINBP: 0/1000 epochs, SSE = 57.2609.
TRAINBP: 200/1000 epochs, SSE = NaN.
TRAINBP: 400/1000 epochs, SSE = NaN.
TRAINBP: 600/1000 epochs, SSE = NaN.
TRAINBP: 800/1000 epochs, SSE = NaN.
TRAINBP: 1000/1000 epochs, SSE = NaN.

Öğrenme katsayısının 0.9 seçilmesi durumunda;

TRAINBP: 0/1000 epochs, SSE = 91.9696.
TRAINBP: 200/1000 epochs, SSE = NaN.
TRAINBP: 400/1000 epochs, SSE = NaN.

TRAINBP: 600/1000 epochs, SSE = NaN.
TRAINBP: 800/1000 epochs, SSE = NaN.
TRAINBP: 1000/1000 epochs, SSE = NaN.

Uygulama sonuçlarından da görüldüğü gibi öğrenme katsayısı 0.1 ve 0.9 değer aralığında istenen sonucu verememiştir. Öğrenme katsayısı değerleri 0.1 in altında seçilecek olursa şu sonuçlar elde edilecektir; Şekil 5.16'da öğrenme katsayısının 0.05 olarak seçilmesi durumunda gerçekleştirilen işlev yaklaşılması görülmektedir.

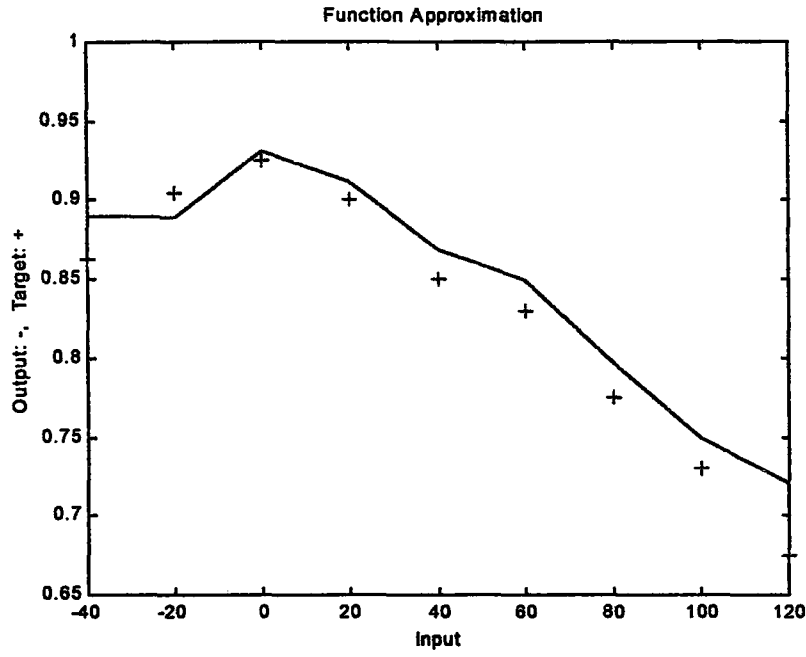


Şekil 5.16 - 10 nöronlu tek saklı katmanlı sinir ağında $\eta = 0.05$ seçilmesi halinde elde edilen sonuçlar

Eğitim sırasında ağı, yerel minimuma takılmış ve hata limitine ulaşamamıştır.

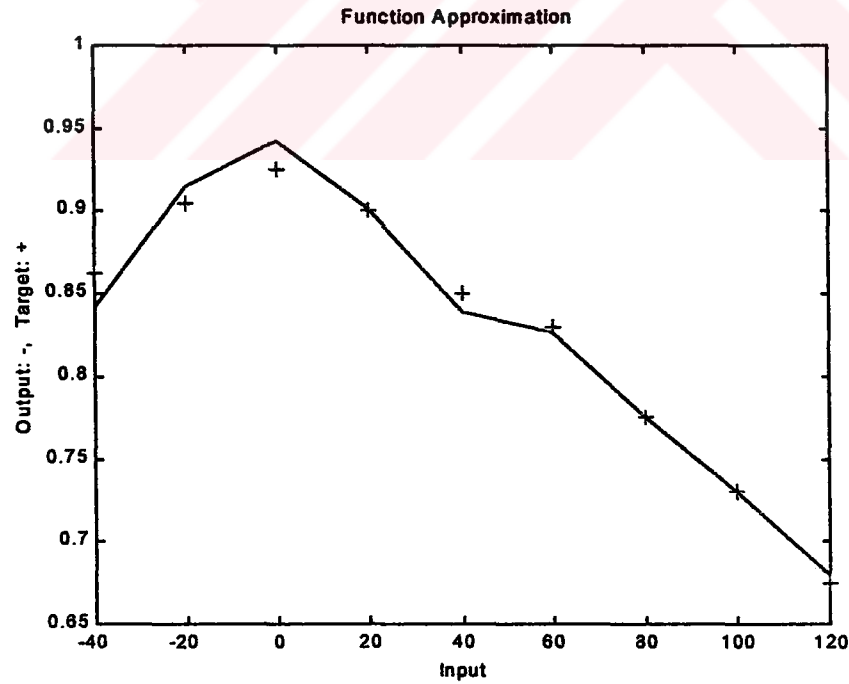
TRAINBP: 149200/150000 epochs, SSE = 0.004275.
TRAINBP: 149400/150000 epochs, SSE = 0.004275.
TRAINBP: 149600/150000 epochs, SSE = 0.004275.
TRAINBP: 149800/150000 epochs, SSE = 0.004275.
TRAINBP: 150000/150000 epochs, SSE = 0.004275.

Öğrenme katsayısının 0.03 seçilmesi durumunda Şekil 5.17 elde edilir.



Şekil 5.17 – Saklı katman nöron sayısı 10, $\eta=0.03$ ve $m=0.8$ iken elde edilen eğitim sonuçları

Öğrenme katsayısının 0.015 seçilmesi durumunda Şekil 5.18 elde edilir.

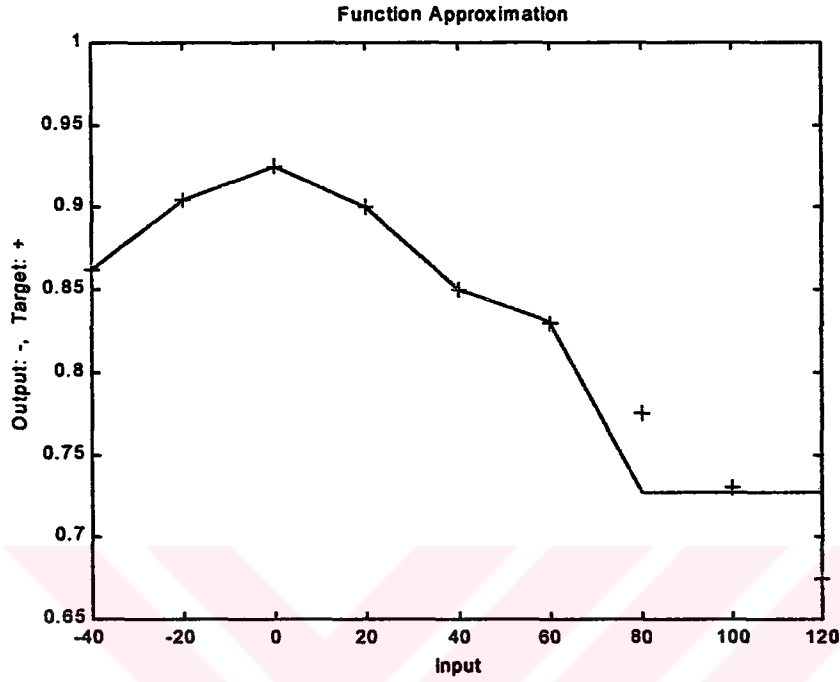


Şekil 5.18 – 10 nöronlu tek saklı katmanlı sinir ağında $\eta = 0.015$ seçilmesi durumu

TRAINBP: 0/45500 epochs, SSE = 54.3531.
 TRAINBP: 60/45500 epochs, SSE = 0.384245.
 TRAINBP: 120/45500 epochs, SSE = 0.0439612.
 TRAINBP: 180/45500 epochs, SSE = 0.0080257.

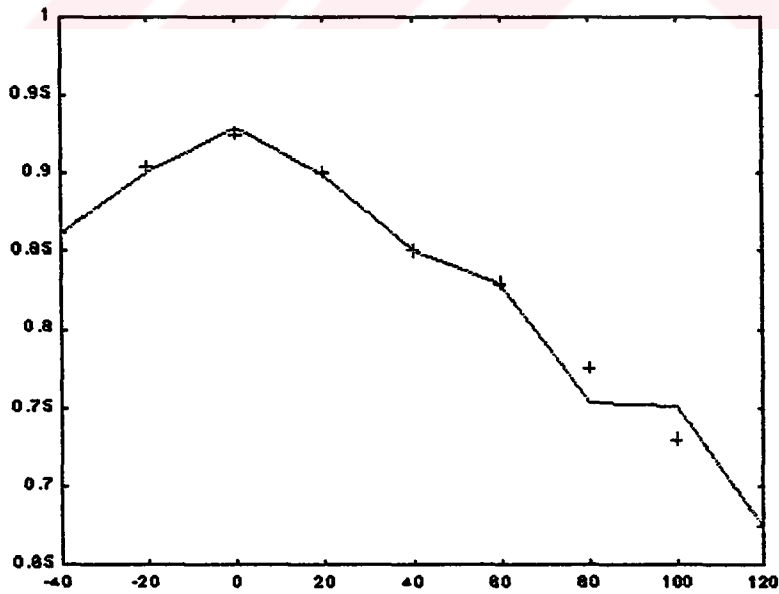
TRAINBP: 240/45500 epochs, SSE = 0.00185427.
TRAINBP: 270/45500 epochs, SSE = 0.000981514.

Öğrenme katsayısının 0.009 seçilmesi durumunda Şekil 5.19 elde edilir.



Şekil 5.19 - 10 nöronlu tek saklı katmanlı sinir ağında $\eta = 0.009$ seçilmesi durumu

Öğrenme katsayısının 0.005 seçilmesi durumunda Şekil 5.20 elde edilir.



Şekil 5.20 - 10 nöronlu tek saklı katmanlı sinir ağında $\eta = 0.005$ seçilmesi durumu

TRAINBP: 127140/150000 epochs, SSE = 0.001.

Bir diğer denemede ise aşağıda belirtilen sonuçlar gözlenmiştir.

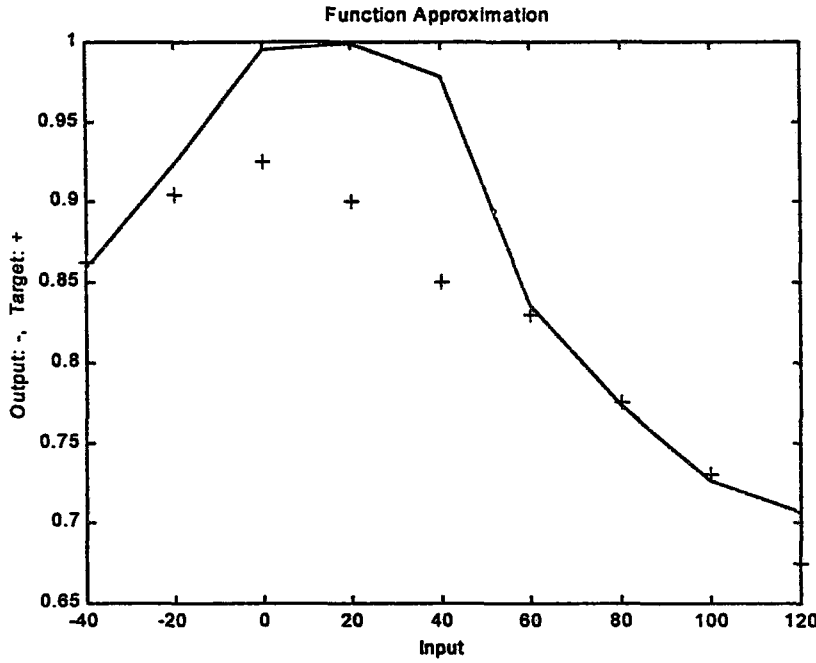
KOMUT DİZGİSİ

```
clf reset                %clear the screen
H=26;                   %saklı katman nöron sayısı
R=1000;                 %eğitme döngüsü sayısı
E=0.03;                 %hedef hata değeri
lr=0.97;                %öğrenme oranı
m=0.95;                 %momentum
P=[-40 -20 0 20 40 60 80 100 120]; %giriş verileri
T=[0.8625 0.905 0.925 0.9 0.85 0.83 0.775 0.73 0.675]; %çıkış verileri
W1 = randn(26,10);
B1 = randn(26,1);
W2 = randn(10,26);
B2 = randn(10,1);
[W1,B1,W2,B2] = initff(P,H,'logsig',T,'logsig');
[W1,B1,W2,B2,iter,errors] =
    trainbpx(W1,B1,'logsig',W2,B2,'logsig',P,T,[D R E lr m]);
pause
cls
P=[ -30 -10 5 25 35 50 70 90 110]; %hedef çıkış değerleri
A=simuff(W1,B1,'logsig',W2,B2,'logsig',P,T,[D R E lr m])
plot
end
```

İşlem sonuçları aşağıdaki gibi olacaktır;

```
TRAINBPX: 0/1000 epochs, lr = 0.97, SSE = 0.237504.
TRAINBPX: 100/1000 epochs, lr = 0.00637849, SSE = 0.0329896.
TRAINBPX: 200/1000 epochs, lr = 4.40461e-005, SSE = 0.0325172.
TRAINBPX: 300/1000 epochs, lr = 2.60776e-007, SSE = 0.0325127.
TRAINBPX: 400/1000 epochs, lr = 1.54393e-009, SSE = 0.0325127.
TRAINBPX: 500/1000 epochs, lr = 9.1409e-012, SSE = 0.0325127.
TRAINBPX: 600/1000 epochs, lr = 5.4119e-014, SSE = 0.0325127.
TRAINBPX: 700/1000 epochs, lr = 2.13765e-015, SSE = 0.0325127.
TRAINBPX: 800/1000 epochs, lr = 2.13765e-015, SSE = 0.0325127.
TRAINBPX: 900/1000 epochs, lr = 2.13765e-015, SSE = 0.0325127.
TRAINBPX: 1000/1000 epochs, lr = 2.13765e-015, SSE = 0.0325127.
```

Bu denemede de ağ, eğitim sırasında yerel minimuma takılmış ve hata limitine ulaşamamıştır. Şekil 5.21'de işlev yaklaşıklığı grafiği görülmektedir. Şekil 5.21'den de anlaşılacağı gibi istenen düzeyde iyi sonuç elde edilememiştir.



Şekil 5.21 - 26 nöronlu tek saklı katmanlı sinir ağında $\eta = 0.97$ seçilmesi durumu

Özetle en ideal sonucun öğrenme katsayısının 0.001 seçildiği durumda elde edildiği söylenebilir.

5.1.5. İdeal parametre değerleri belirlendikten sonra yapılan uygulamalar

Tablo 4.1’de verilen farklı sıcaklıklardaki delinme dayanımları polyster malzemenin 6 farklı kalınlık değeri için ölçülmüştür. Bu 6 değişik kalınlık değerinde elde edilen delinme dayanımlarının sinir ağına tanıtılması ile eğitim boyunca hata karesindeki değişme gözlenmiş ve ilgili grafik Şekil 5.22’de gösterilmiştir.

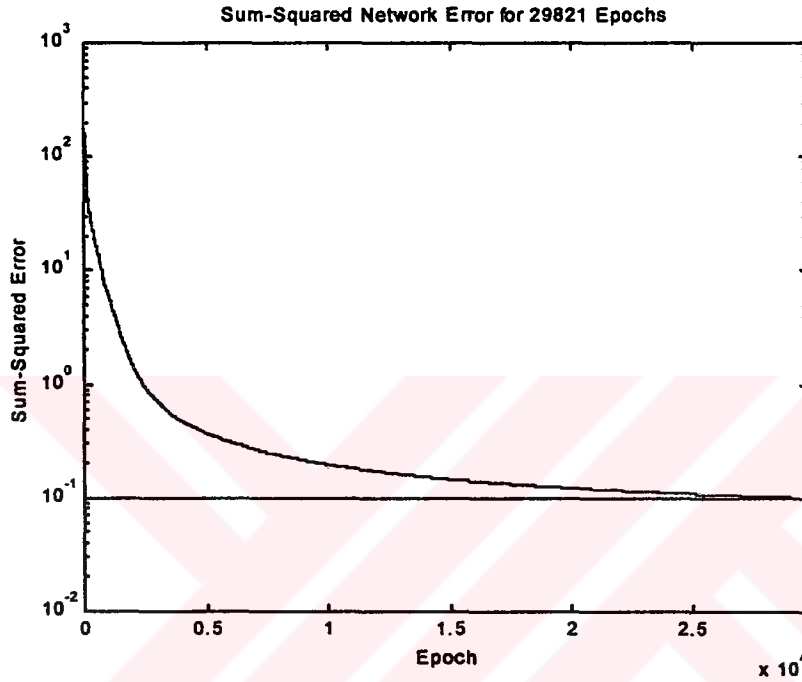
KOMUT DİZGİSİ

```
H=10; %number of neurons in hidden layer
D=1000; %display period
R=30000; %number of training iterations
E=0.1; %desired error
lr=0.001; %learning rate
m=0.8; %momentum
I=-1:.01:1;
P=[-40 -20 0 20 40 60 80 100 120];
T=[0.8625 0.905 0.925 0.9 0.85 0.83 0.775 0.73 0.675;0.775 0.795 0.8125 0.8 0.7625 0.735 0.7 0.65
0.6025;0.6125 0.6375 0.675 0.65 0.6 0.575 0.53 0.475 0.4325;0.55 0.575 0.605 0.575 0.525 0.505
0.4725 0.4275 0.385;0.4375 0.455 0.4875 0.45 0.425 0.395 0.3625 0.325 0.29;0.4 0.4125 0.425 0.4
0.3875 0.3625 0.3375 0.305 0.2725];
W1 = randn(10,10);
B1 = randn(10,1);
W2 = randn(10,10);
B2 = randn(10,1);
[W1,B1,W2,B2] = initff(P,H,'logsig',T,'logsig');
```

```

[W1,B1,W2,B2,iter,errors]=trainbp(W1,B1,'logsig',W2,B2,'purelin',P,T,[D R E lr m]);
pause
plotfa(P,T,I,purelin(W2*tansig(W1*I,B1),B2));
pause
cls
P=[ -30 -10 5 25 35 50 70 90 110]
A=simuff[W1,B1,'logsig',W2,B2,'logsig',P,T,[D R E lr m]]
plot
end

```

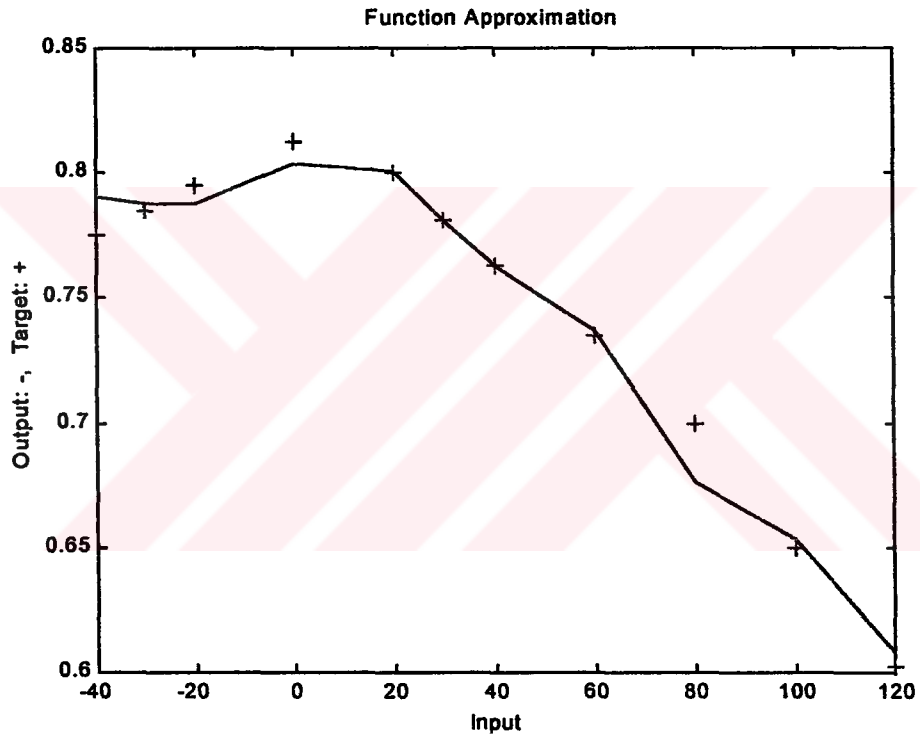


Şekil 5.22 – Eğitim süresince hata karesinin değişimi

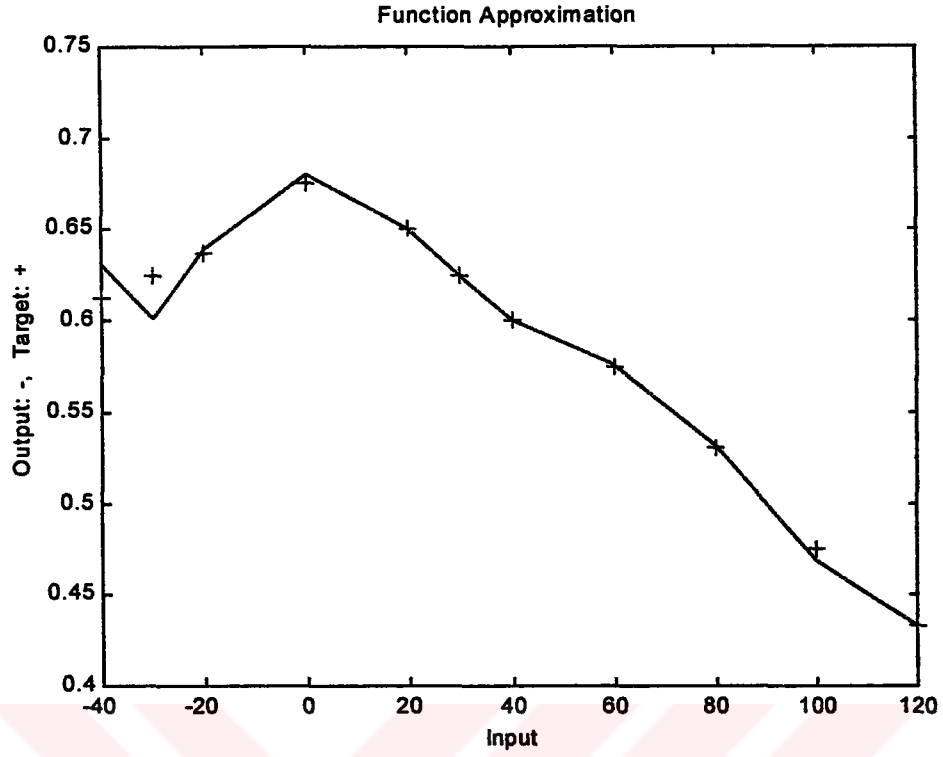
TRAINBP: 0/30000 epochs, SSE = 703.108.
 TRAINBP: 1000/30000 epochs, SSE = 5.87183.
 TRAINBP: 2000/30000 epochs, SSE = 1.44137.
 TRAINBP: 3000/30000 epochs, SSE = 0.694548.
 TRAINBP: 4000/30000 epochs, SSE = 0.47799.
 TRAINBP: 5000/30000 epochs, SSE = 0.373869.
 TRAINBP: 6000/30000 epochs, SSE = 0.311068.
 TRAINBP: 7000/30000 epochs, SSE = 0.268764.
 TRAINBP: 8000/30000 epochs, SSE = 0.238303.
 TRAINBP: 9000/30000 epochs, SSE = 0.215336.
 TRAINBP: 10000/30000 epochs, SSE = 0.197421.
 TRAINBP: 11000/30000 epochs, SSE = 0.183072.
 TRAINBP: 12000/30000 epochs, SSE = 0.171334.
 TRAINBP: 13000/30000 epochs, SSE = 0.161563.
 TRAINBP: 14000/30000 epochs, SSE = 0.153311.
 TRAINBP: 15000/30000 epochs, SSE = 0.146255.
 TRAINBP: 16000/30000 epochs, SSE = 0.140156.
 TRAINBP: 17000/30000 epochs, SSE = 0.134835.
 TRAINBP: 18000/30000 epochs, SSE = 0.130155.
 TRAINBP: 19000/30000 epochs, SSE = 0.126009.
 TRAINBP: 20000/30000 epochs, SSE = 0.122312.
 TRAINBP: 21000/30000 epochs, SSE = 0.118996.
 TRAINBP: 22000/30000 epochs, SSE = 0.116006.

TRAINBP: 23000/30000 epochs, SSE = 0.113297.
 TRAINBP: 24000/30000 epochs, SSE = 0.110833.
 TRAINBP: 25000/30000 epochs, SSE = 0.108582.
 TRAINBP: 26000/30000 epochs, SSE = 0.106517.
 TRAINBP: 27000/30000 epochs, SSE = 0.104618.
 TRAINBP: 28000/30000 epochs, SSE = 0.102865.
 TRAINBP: 29000/30000 epochs, SSE = 0.101243.
 TRAINBP: 29821/30000 epochs, SSE = 0.0999986.

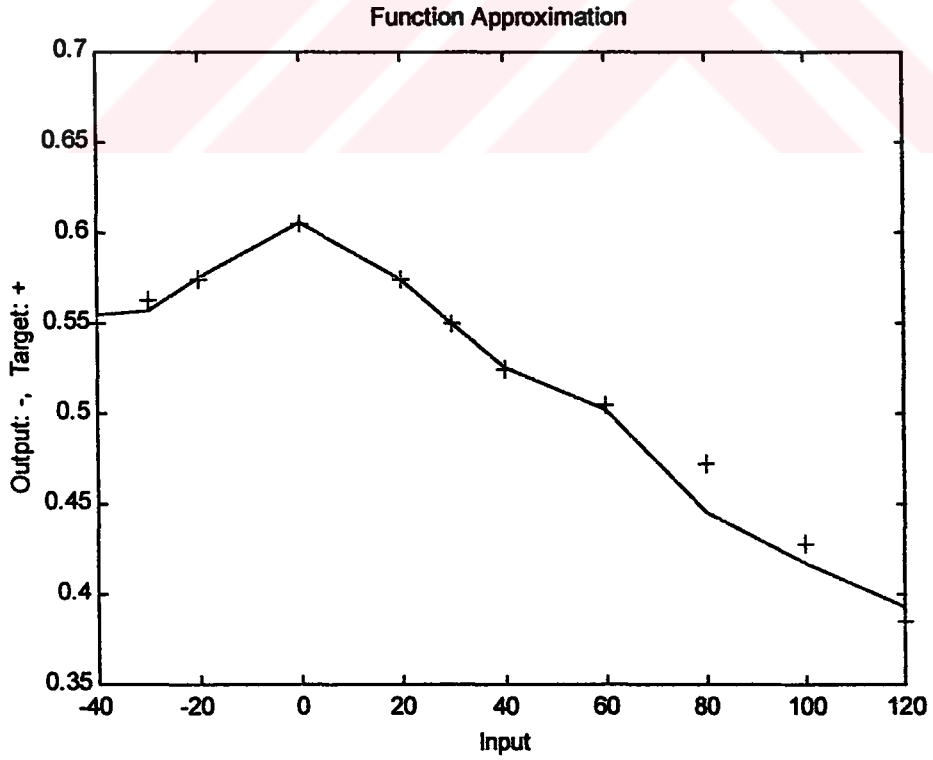
Tablo 4.1'deki ölçümler $12\ \mu\text{m}$, $23\ \mu\text{m}$, $36\ \mu\text{m}$, $50\ \mu\text{m}$, $100\ \mu\text{m}$, ve $200\ \mu\text{m}$ polyester folye kalınlıklarında yapılmıştır. Her bir kalınlıkta yapılan ölçme sonuçlarının sinir ağına tanıtılması ile elde edilen grafikler aşağıdaki şekillerde gösterilmiştir.



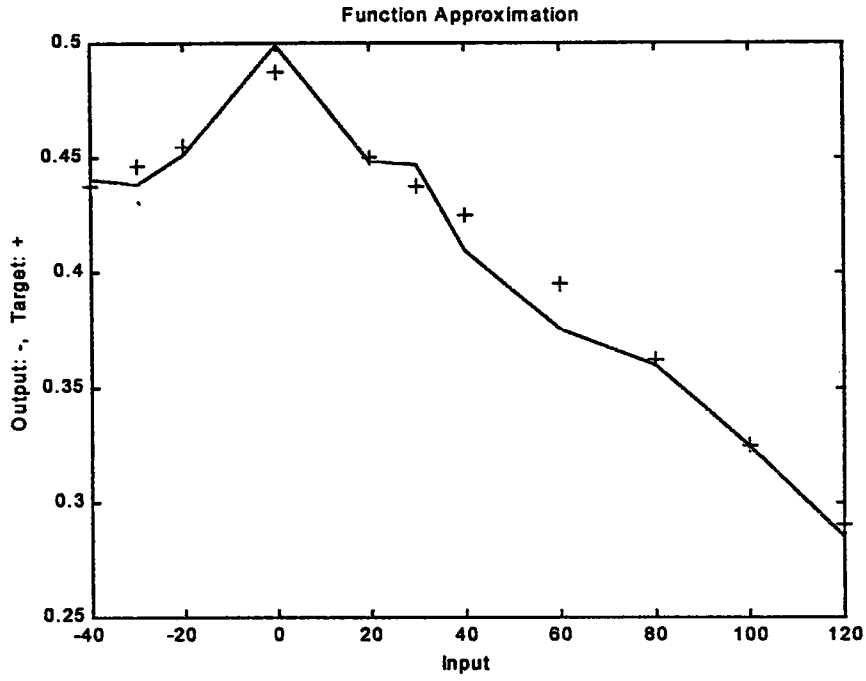
Şekil 5.23 - $23\ \mu\text{m}$ PT folye kalınlığında elde edilen deney sonuçlarının ağa tanıtılması ile elde edilen grafikte $\eta=0.001$, $m=0.8$, $E=0.001$, $H=10$ alınmıştır.



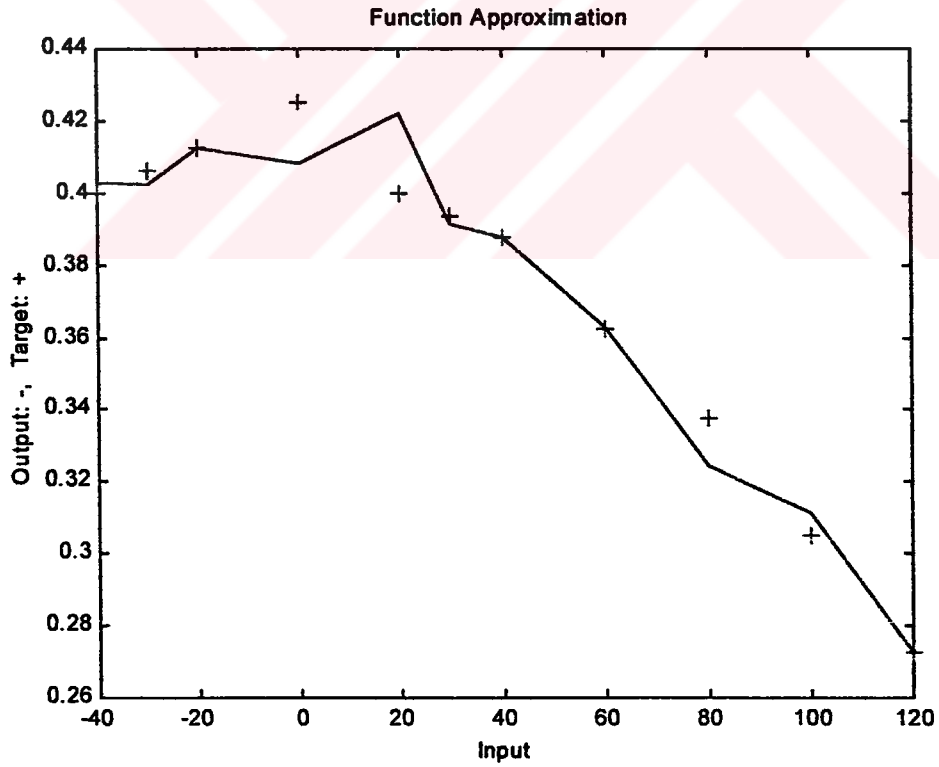
Şekil 5.24 - 36 μ m PT folye kalınlığında elde edilen deney sonuçlarının ağa tanıtılması ile elde edilen grafikte $\eta=0.001$, $m=0.8$, $E=0.001$, $H=10$ alınmıştır.



Şekil 5.25 - 50 μ m PT folye kalınlığında elde edilen deney sonuçlarının ağa tanıtılması ile elde edilen grafikte $\eta=0.001$, $m=0.8$, $E=0.001$, $H=10$ alınmıştır.

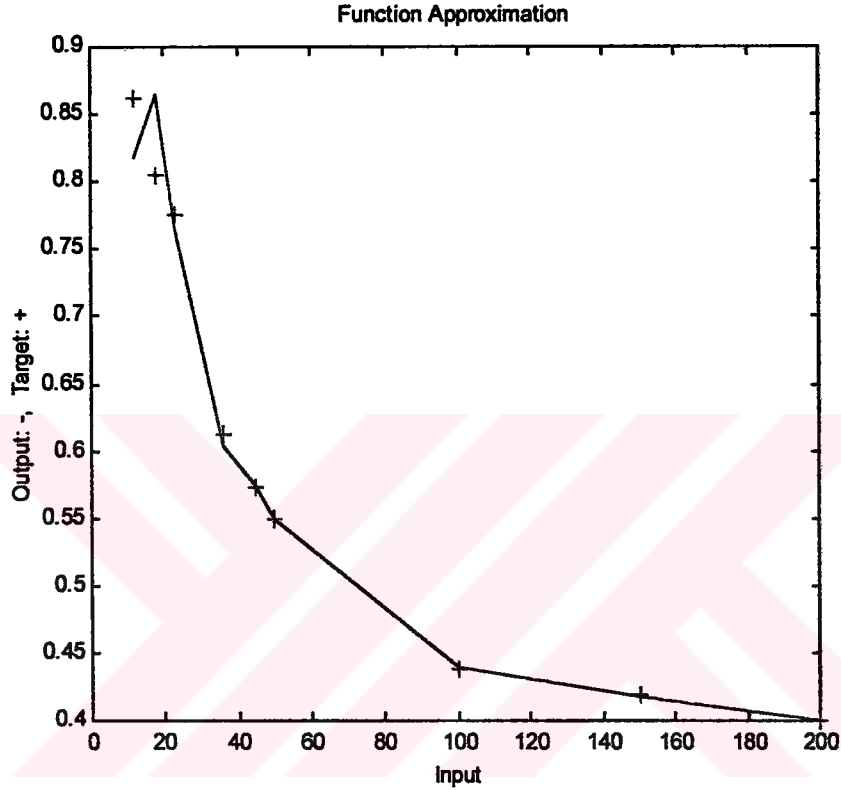


Şekil 5.26 - 100 μ m PT folye kalınlığında elde edilen deney sonuçlarının ağa tanıtılması ile elde edilen grafikte $\eta=0.001$, $m=0.8$, $E=0.001$, $H=10$ alınmıştır.



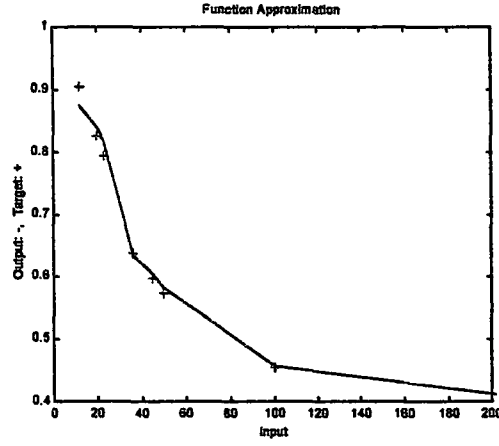
Şekil 5.27 - 200 μ m PT folye kalınlığında elde edilen deney sonuçlarının ağa tanıtılması ile elde edilen grafikte $\eta=0.001$, $m=0.8$, $E=0.001$, $H=10$ alınmıştır.

Şimdi de -40°C sabit sıcaklıkta polyester folye kalınlıklarının $12\ \mu\text{m}$, $23\ \mu\text{m}$, $36\ \mu\text{m}$, $50\ \mu\text{m}$, $100\ \mu\text{m}$ ve $200\ \mu\text{m}$ olarak değiştiği durumda elde edilen delinme dayanımı değerleri ağıta tanıtılacak ve farklı folye kalınlıklarında delinme dayanımı değerlerine ulaşılmaya çalışılacaktır. Şekil 5.28 ilgili grafiği göstermektedir.

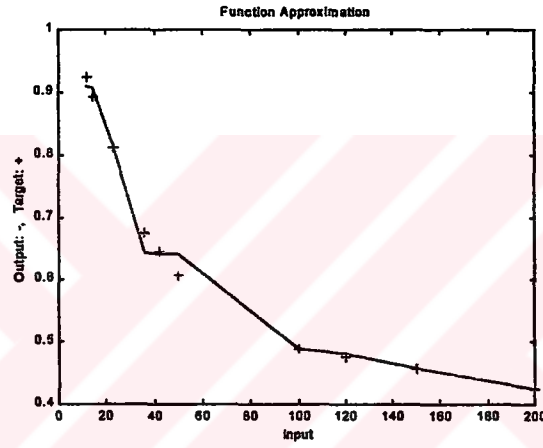


Şekil 5.28 – (-40°C) sıcaklıkta ve değişen folye kalınlıklarında elde edilen delinme dayanımı değerleri

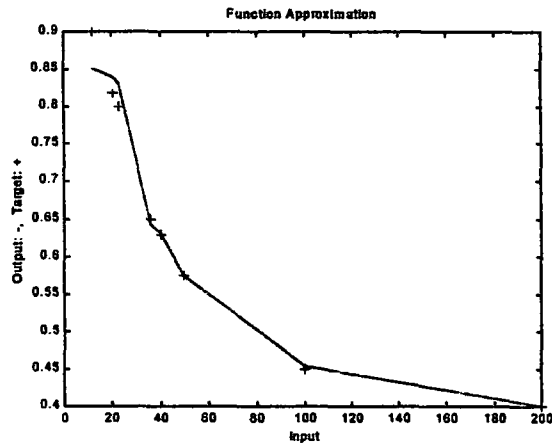
Şekil 5.29, Şekil 5.30, Şekil 5.31, Şekil 5.32, Şekil 5.33, Şekil 5.34, Şekil 5.35 ve Şekil 5.36'da Tablo 4.1'deki delinme dayanımı değerleri, sıcaklık sabit tutularak, ağıta tanıtılmış ve işlev yaklaşılama grafikleri elde edilmiştir.



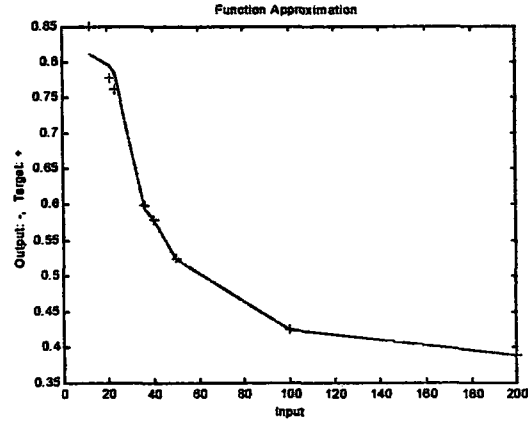
Şekil 5.29 – (-20°C) sıcaklıkta ve değişen folye kalınlıklarında elde edilen delinme dayanımı değerleri



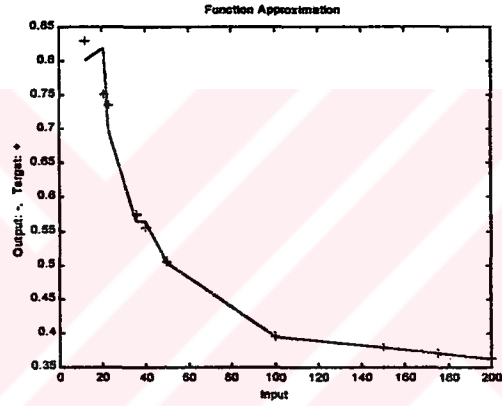
Şekil 5.30 – (0°C) sıcaklıkta ve değişen folye kalınlıklarında elde edilen delinme dayanımı değerleri



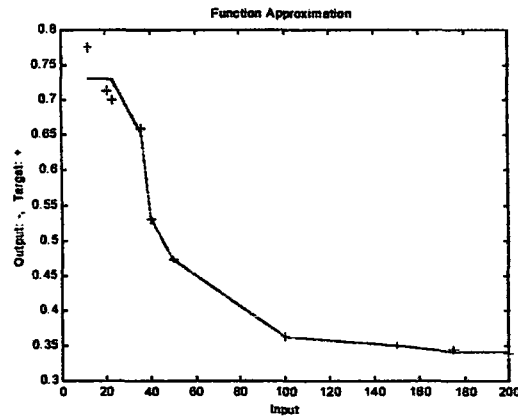
Şekil 5.31 – (20°C) sıcaklıkta ve değişen folye kalınlıklarında elde edilen delinme dayanımı değerleri



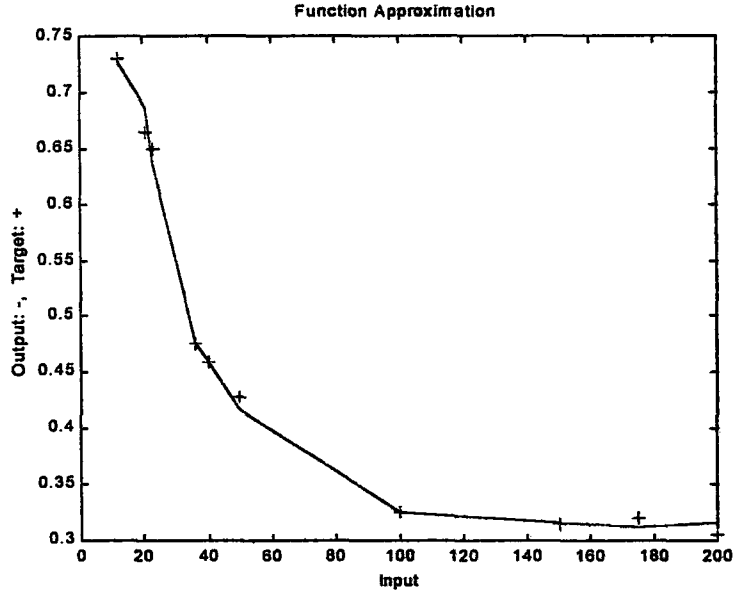
Şekil 5.32 – (40°C) sıcaklıkta ve değişen folye kalınlıklarında elde edilen delinme dayanımı değerleri



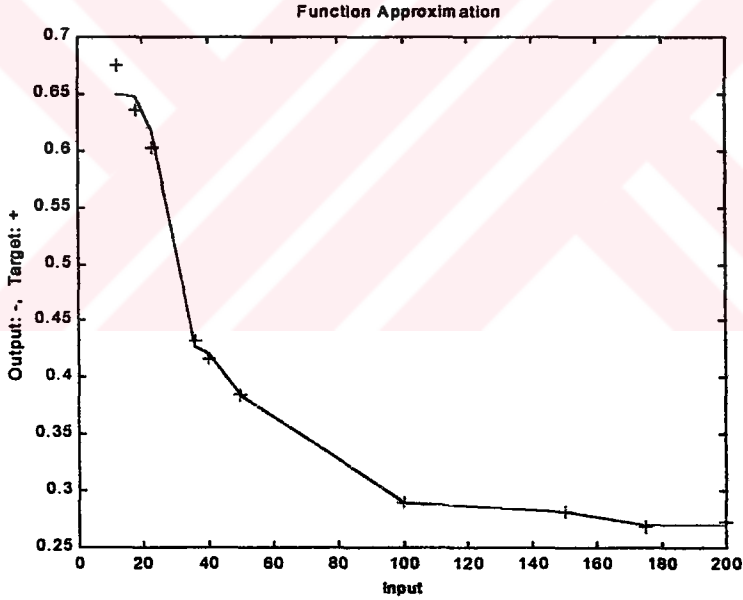
Şekil 5.33 – (60°C) sıcaklıkta ve değişen folye kalınlıklarında elde edilen delinme dayanımı değerleri



Şekil 5.34 – (80°C) sıcaklıkta ve değişen folye kalınlıklarında elde edilen delinme dayanımı değerleri



Şekil 5.35 – (100°C) sıcaklıkta ve değişen folye kalınlıklarında elde edilen delinme dayanımı değerleri



Şekil 5.36 – (120°C) sıcaklıkta ve değişen folye kalınlıklarında elde edilen delinme dayanımı değerleri

Elde edilen tüm verilerin birleştirilerek tablollanması Tablo 5.1 ve Tablo 5.2’de yapılmıştır. Tablo 5.1’de hedef çıktılar olan -30°C, -10°C, 5°C, 25°C, 35°C, 50°C, 70°C, 90°C, 110°C sıcaklık değerleri için sabit polyester folye kalınlıklarında delinme dayanımı değerleri elde edilmiştir. Tablo 5.2’de ise hedef çıkış değerleri,

sabit sıcaklık değerlerinde, 20 μ m, 30 μ m, 40 μ m, 70 μ m, 90 μ m, 170 μ m folye kalınlıklarında hesaplanan delinme dayanımı değerleridir.

Yukarıdaki tüm grafiklerde output yani çıkış değerlerinin delinme dayanımı değerleri olduğu bilinmektedir. Bu değerlerin Tablo 4.1’de verilen deney sonuçları olduğu daha önce söylenmişti. Bu verilerin sinir ağına tanıtımı yapılırken delinme dayanımı değerleri 400 sayısı ile bölünmüş böylece 0 ile 1 arasında bir sayıya küçültülerek ağın öğrenmesi için gereken biçime dönüştürülmüştür. Tablo 5.1 ve Tablo 5.2 hazırlanırken de 0 ile 1 aralığındaki çıkış değerleri 400 sayısı ile çarpılarak delinme dayanımı değerleri onluk sayısal değerlere dönüştürülmüştür.

Tablo 5.1. – Farklı sıcaklık değerlerinde yapay sinir ağı ile elde edilen delinme dayanımı değerleri (E_d kV/mm)

$t^{\circ}\text{C}$	-30 $^{\circ}\text{C}$	-10 $^{\circ}\text{C}$	5 $^{\circ}\text{C}$	25 $^{\circ}\text{C}$	35 $^{\circ}\text{C}$	50 $^{\circ}\text{C}$	70 $^{\circ}\text{C}$	90 $^{\circ}\text{C}$	110 $^{\circ}\text{C}$
12 μm	352	364	362	354	320	336	316	304	280
23 μm	315	319	321	316	309	300	286	266	254
36 μm	242	263	268	255	245	237	221	200	183
50 μm	223	237	238	225	215	207	190	175	165
100 μm	175	190	195	179	172	158	147	140	120
200 μm	161	164	164	162	156	150	138	128	117

Tablo 5.2. – Farklı polyester folye kalınlıklarında yapay sinir ağı ile elde edilen delinme dayanımı değerleri (E_d kV/mm)

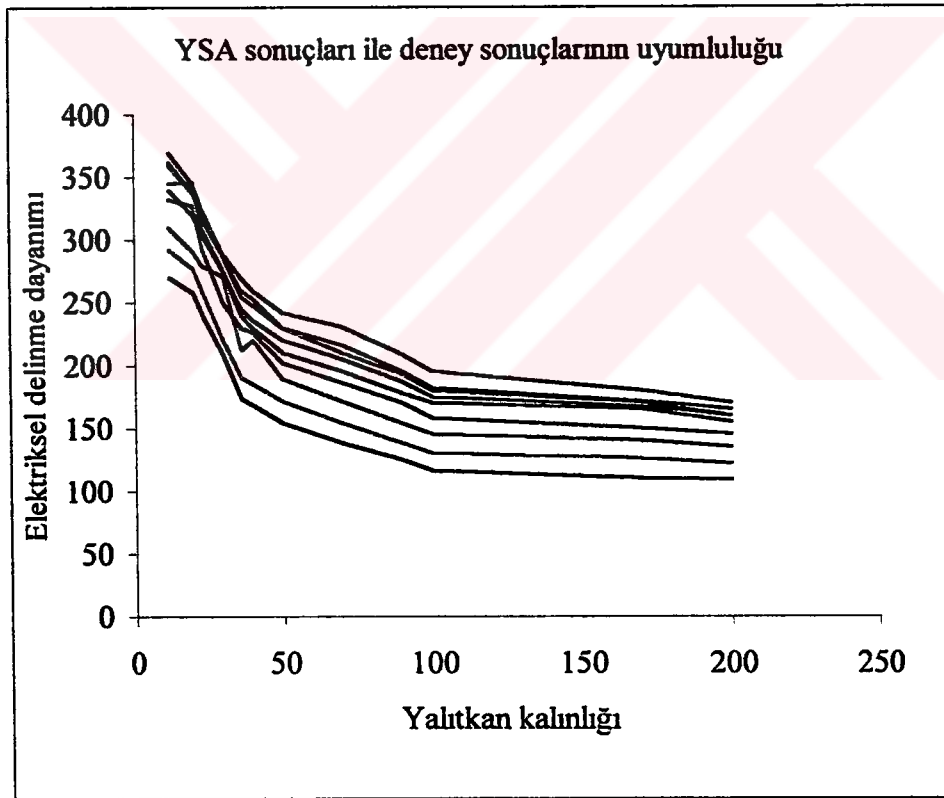
$t^{\circ}\text{C}$	-40 $^{\circ}\text{C}$	-20 $^{\circ}\text{C}$	0 $^{\circ}\text{C}$	20 $^{\circ}\text{C}$	40 $^{\circ}\text{C}$	60 $^{\circ}\text{C}$	80 $^{\circ}\text{C}$	100 $^{\circ}\text{C}$	120 $^{\circ}\text{C}$
20 μm	346	339	345	338	320	327	291	277	258
30 μm	276	288	289	288	275	250	271	220	208
40 μm	236	248	260	253	230	226	220	185	168
70 μm	205	216	231	210	195	186	171	154	138
90 μm	187	195	209	193	178	170	154	138	125
170 μm	167	171	180	170	165	150	140	126	110

Çıkan sonuçlar incelendiğinde, yapay sinir ağlarının bazı eğitim döngülerinde iyi öğrenme yapabildiği bazılarında ise nispeten duyarlılıktan uzaklaştığı saptanmıştır. Bu uygulamada sinir ağı, giriş değerlerine bağlı kalarak ara değer atamaları yapmış, giriş veri uzayı dışına çıkmamıştır.

Tablo 5.2’de verilen, yapay sinir ağı ile elde edilen delinme dayanımı değerleri ile Tablo 4.1’deki delinme dayanımı değerlerinin birbirlerine uyumunu incelemek amacıyla ortak bir tablo oluşturulur. Şekil 5.37, Tablo 5.3’de oluşturulan bu ortak tablonun sıcaklık ve folye kalınlığı değerlerinin değişim grafiğini göstermektedir.

Tablo 5.3. – Farklı polyester folye kalınlıklarında ve sıcaklık değerlerinde deney ve sinir ağı uygulaması ile elde edilen delinme dayanımı değerleri (E_d kV/mm)

$t^{\circ}\text{C}$	-40°C	-20°C	0°C	20°C	40°C	60°C	80°C	100°C	120°C
12 μm	345	362	370	360	340	332	310	292	270
20 μm	346	339	345	338	320	327	291	277	258
23 μm	310	318	325	320	305	294	280	260	241
30 μm	276	288	289	288	275	250	271	220	208
36 μm	245	255	270	260	240	230	212	190	173
40 μm	236	248	260	253	230	226	220	185	168
50 μm	220	230	242	230	210	202	189	171	154
70 μm	205	216	231	210	195	186	171	154	138
90 μm	187	195	209	193	178	170	154	138	125
100 μm	175	182	195	180	170	158	145	130	116
170 μm	167	171	180	170	165	150	140	126	110
200 μm	160	165	170	160	155	145	135	122	109



Şekil 5.37 – Artan sıcaklık değerleri ve yalıtkan kalınlığına göre elektriksel delinme dayanımı eğrileri

Şekil 5.37’den de görüldüğü gibi yapay sinir ağı uygulaması sonuçları, deney verileri ile uyumluluk göstermekte, bulunması istenen hedef çıktılar makul değerlerde gözlenmektedir.

6. SONUÇLAR VE ÖNERİLER

Yalıtkan maddelerde elektriksel delinme dayanımının yapay sinir ağları ile belirlenmesi konulu tez çalışması kapsamında, deneysel olarak farklı sıcaklık ve yalıtkan madde kalınlıklarında elde edilen delinme dayanımı değerleri, geriye yayılım algoritması kullanılarak yapay sinir ağına tanıtılmış ve deneme yanılma yolu ile en uygun çıkışı verecek ağ mimarisi tasarlanmaya çalışılmıştır.

Bilindiği gibi, verilen bir problem için ne tür bir mimari seçileceği kurallara bağlı olmayan, tamamen tasarımcının seçimine bağlı bir konudur. Tasarımcı çeşitli mimariler üzerinde denemeler yaparak içlerinden en uygun olanını seçer.

Bu çalışma kapsamında, hata limit değeri, saklı katman nöron sayısı, momentum katsayısı, öğrenme oranı parametreleri değiştirilerek denemeler yapılmış ve optimum sonuç, $E = 0.001$, $H = 10$, $m = 0.8$, $\eta = 0.001$ iken elde edilmiştir.

Yapılan denemeler sonucunda denebilir ki; yapay sinir ağları uygulamalarında performans artışı ve yerel minimuma takılmayı engelleyebilmek için tavsiye edilen işlemler, saklı katman nöron sayısını arttırmak, öğrenme oranını azaltmak, farklı ağırlık katsayıları kullanarak değişik denemeler yapmaktır.

Uygulamada, sinir ağına tanıtılan giriş verileri ile ağ eğitimi tamamlanmış, ulaşılmak istenen hedef değerler giriş veri uzayı içerisindeyse sonuca ulaşılmış, giriş veri uzayı dışında, sinir ağı, çıkış değeri belirleyememiştir. Bu sonuçlar doğrultusunda sinir ağı interpolasyon yapabilmiş ama ekstrapolasyon yapamamıştır.

Giriş değerleri olarak kullanılan deney verilerinin sınırlı ve az sayıda olmasından dolayı, sinir ağı çıkışında işlev yaklaşımı yapılırken ağ, kendisine tanıtılan veriler ile çıktısı hedeflenen veriler arasındaki ilişkiyi kabaca kurmuş, öğrendiği giriş değerlerinden yola çıkarak ortalama değer atamaları yapmıştır. Giriş değerleri kümesinin genişletilmesi, ağa giriş katmanında tanıtılacak nöronların arttırılması demektir. Bunun sonucunda eğitim kümesi genişleyecek, kullanılan ağırlık katsayıları artacak, ağdaki öğrenme süreci uzayacak ama aynı zamanda da çıkış değerleri daha etkin olarak hesaplanacaktır. Bu fikir doğrultusunda giriş katmanında

ağa tanıtılan 9 nöronlu veri kümesini çoğullamak için matematiksel orantı yöntemi ile giriş veri uzayı genişletilmiş, sonuçta sinir ağının daha kolay eğitilebildiği gözlenmiştir.

Sonuçta, tez çalışması kapsamında, pek çok alanda olduğu gibi elektriksel delinme dayanımı hesaplamalarında da yapay sinir ağlarının kullanılabileceği gösterilmiştir. Ancak duyarlı sonuçlar için, giriş veri kümesinin yeterince büyük olması gerekliliği, aksi halde ağ çıktısının kabaca matematiksel yöntemlerle de yapılabilecek bir ortalama değer hesaplaması problemine dönüşebileceği gözlenmiştir.

Bu çalışmada göz önüne alınan yaklaşım, gazlar, sıvılar, karma yalıtkanlar gibi değişik yalıtkan ortam ve elektrik alan koşullarında boşalma olaylarının incelenmesine uygulanarak geniş bir çalışma alanı elde edilebilir. Bu amaçla koşulları tanımlı deneysel verilere gereksinim vardır. Bugüne kadarki ve bundan sonra yapılacak olan deneysel çalışmalar veri havuzunu dolduracaktır. Bu da pahalı, zahmetli ve zaman alıcı deneysel çalışmaların yükünü hafifletecektir. Ayrıca yapay sinir ağlarının yöntem ve algoritma olarak değişik özelliklerini kullanarak farklı çalışmalar da yapılabilir.

KAYNAKLAR

- [1] **Ataman, F., Kaynak, T.** “Yapay Zeka İçeren Çözümlerin Sistem Modelleme Yoluyla Bilgisayar Ortamında İrdelenmesi” 9.Yapay Zeka ve Sinir Ağları Sempozyumu, 21-23 Haziran 2000, Bildiriler Sayfa 353-360
- [2] **G. W. NG** “Application of Neural Networks to Adaptive Control of Nonlinear Systems”, 1997
- [3] **Arslan, H.** “YSA ile Robotlarda Hareket Kontrolü” İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü Doktora Tezi, 1999
- [4] **Gökçay, M. M.**, “Yapay Sinir Ağları ve Uygulamaları” İstanbul Üniversitesi Fen Bilimleri Enstitüsü Yüksek Lisans Tezi, 1993
- [5] **Demir, Y. K.** “Yapay Sinir Ağları ile Ulaştırma Taleplerinin Modellenmesi” İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü Yüksek Lisans Tezi, 1997
- [6] **Başbuğ, R.** “Yapay Sinir Ağları” BYTE, Şubat 1994, Sayfa 163-168
- [7] **Sutton, R.** “Two Problems with Backpropagation and other Steepest-Descent Learning Procedures for Networks, Proceedings of the 8th Annual Conference on the Cognitive Science Society, 823-831, 1986
- [8] **Metrotra, K., Mohan, C. K., Ranka, S.** “Elements of Artificial Neural Networks”, 1997
- [9] **Mumyalmaz, B.** “Y.S.A. Kontrolüne Dayalı Statik VAR Kompanzatörünün Pspice ile Simülasyonu” Yıldız Teknik Üniversitesi Yüksek Lisans Tezi, 1998
- [10] **Hush, D. R.** “Error Surfaces for Multilayer Perceptrons”, International Joint Conference on Neural Networks, vol 1, 759-764, 1991
- [11] **Efe, M. Ö., Kaynak, O.** “Yapay Sinir Ağları ve Uygulamaları” Boğaziçi Üniversitesi Yayınları, 2000
- [12] **Serhadlıoğlu, G.** “Yapay Sinir Ağlarının Teknik Analizde Kullanımı ve Bir Uygulama”, Marmara Üniversitesi Sosyal Bilimler Enstitüsü İşletme Ana Bilim Dalı Yüksek Lisans Tezi, 1999
- [13] **Şeker, S.** “Artificial Neural Networks for System Identification and Control” Ders Notları

- [14] Shynk, J. J., Roy, S. "Convergence Properties and Stationary Points of a Perceptron Learning Algorithm", Proc. IEEE, vol 78, no. 10, Oct 1990, pp. 1599-1604
- [15] Lippmann, R. P. "An Introduction to Computing with Neural Nets", IEEE ASSP Mag., Vol. 4, pp. 4-22, April 1987
- [16] Sjöberg, J., Hjalmarsson, H., Ljung, L. "Neural Networks in System Identification", Department of Electrical Engineering, Linköping University, Sweden
- [17] Sağiroğlu, Ş. "Levenberg-Marquardt Metodunun Bir Robot Sensörün Yapay Sinir Ağı Modellenmesinde Kullanılması", 5. Türk Yapay Zeka ve Yapay Sinir Ağları Sempozyumu, İstanbul, 27-28 Haziran 1996, Bildiriler sayfa no. 323-328
- [18] Hammerstrom, D., "Working with Neural Networks", IEEE Spectrum, July 1993, pp. 46-53
- [19] Murray, A. F., "Applications of Neural Networks", Kluwer Academic Publishers, 1995
- [20] Hammerstrom, D., "Neural Networks at Work", IEEE Spectrum, June 1993, pp.26-32
- [21] Zhu, K., "ECG Feature Recognition Using Artificial Neural Networks", Department of Electronic Systems Engineering, University of Essex, 1992
- [22] Lippman, R. P., "Pattern Classification Using Neural Networks", IEEE Communications Magazine, pp. 47-64, November 1989
- [23] Lippman, R. P., "Review of Neural Networks for Speech Recognition", Neural Comp., vol. 1, pp. 1-38, 1989
- [24] Tesauro, G., Sejnowski, T. J., "A Neural Network that Learns to Play Backgammon", Neural Information Processing Systems, D. Anderson, ed., pp. 794-803, American Institute of Physics, New York, 1988
- [25] Kalenderli, Ö., Şeker, S., Mardikyan, K., Önal, E., Ayaz, E., "Numerical Computation of Breakdown Voltage by Neural Networks Approach", Numelec 2000 3rd European Conference on Numerical Methods in Electromagnetism, France, March 20-22, 2000
- [26] Widrow, B., Lehr, M.A., "30 Years of Adaptive Neural Networks: Perceptron, Madaline, and Backpropagation", Proc. IEEE, vol. 78, no. 9, September 1990, pp. 1415-1442.

- [27] **Balcı, M. T.**, “Yapay Sinir Ağlarının Modellenmesi ve Uygulama Alanları”, İstanbul Üniversitesi Fen Bilimleri Enstitüsü Yüksek Lisans Tezi, 1995.
- [28] **Ghourab, M. E.**, “Application of Artificial Neural Networks for Insulation Diagnosis”, 9th International Symposium on High Voltage Engineering, August 28-September 1, 1995.
- [29] **Hines, J. W.**, “Matlab Supplement to Fuzzy and Neural Approaches in Engineering”, Wiley – Interscience publication, New York, 1997.
- [30] **Şafak, K. K., Türkay, O. S.**, “Üniversal Motor Dinamiğinin Yapay Sinir Ağları ile Tanılanması”, Endüstri ve Otomasyon Dergisi, sayfa.34-37, Şubat 1999.
- [31] **Bosignoli, R., Mesquita, E. N., Silva Filho, J. I., Souza, H. M.**, “Artificial Neural Network as a Tool for Transmission Line Ampacity Analysis”, 9th International Symposium on High Voltage Engineering, August 28-September 1, 1995.
- [32] **Wayne, K., Schroeder, P.E.**, “Parameter Estimation Using a Backpropagation Neural Network”
- [33] **Ölmez, T., Dokur, Z.**, “Dalgacık Dönüşümü ve Hibrit Yapay Sinir Ağı ile EKG Vurularının Sınıflandırılması”, IEEE SİU-2000, 8. Sinyal İşleme ve Uygulamaları Kurultayı, 12-14 Haziran 2000, Belek, Antalya.
- [34] **Grandvalet, Y., Canu, S., Meizel, D., Langlois, T.**, “Preconditioning for Improving Convergence of Back Propagation Algorithm in Control Applications”
- [35] **Bayram, M.**, “Elektrik Tesisleri Laboratuvar Deneyleri” İstanbul Teknik Üniversitesi Matbaası, Gümüşsuyu, 1974.
- [36] **Kalenderli, Ö., Yılmaz, G.**, “İnce Yalıtkan Tabakaların Elektriksel Dayanımının Belirlenmesinde Deney Koşullarının Etkisi”, Elektrik-Elektronik-Bilgisayar Mühendisliği 8. Ulusal Kongresi, Gaziantep, s. 700-703, 6-12 Eylül 1999.
- [37] **Spencer, R., G.**, “Two-Layer Back Propagation Example Using Matlab’s Neural Network Toolbox”, <http://www.brainlink.com/~neurawar/Ron/bp.1.html>
- [38] **Önal, E., Kalenderli, Ö., Şeker, S., Ayaz, E.**, “A study on breakdown voltage estimation of CO₂+SF₆ mixture using feedforward neural network Approach”, Eleco '99 International Conference on Electrical and Electronics Engineering, Bursa, Turkey, pp. 131-134, December 1-5, 1999.

- [39] **Yılmaz, G., Kalenderli, Ö.**, “The effect of thickness and area on the electric strength of thin dielectric films”, 1996 IEEE International Symposium on Electrical Insulation, Montréal, Québec, Canada, Vol. 2, pp. 478-481, June 16-19, 1996.



ÖZGEÇMİŞ

Melike Oğuz, 1977 yılında Bursa'da doğdu. 1999 yılında İstanbul Teknik Üniversitesi Elektrik-Elektronik Fakültesi, Elektrik Mühendisliği Bölümü'nden mezun oldu. Aynı yıl İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü Elektrik Mühendisliği Yüksek Lisans Programı'na kayıt oldu. Halen bu programda yüksek lisans eğitimine devam etmektedir.

