ISTANBUL TEKNIK ÜNIVERSITESI ★ FEN BILIMLERI ENSTITÜSÜ

YAPAY SINIR AGI ILE
ELEKTROT VE IZOLATÖR BIÇIM OPTIMIZASYONU

YÜKSEK LISANS TEZI

Müh. Suna BOLAT

(504011041)

Tezin Enstitüye Verildigi Tarih : 5 Mayis 2003

Tezin Savunuldugu Tarih : 30 Mayis 2003

Tez Danismani	**Doç. Dr. Özcan KALENDERLI**
Diger Jüri Üyeleri	**Doç. Dr. Serhat SEKER**
	Doç. Dr. Sedef KENT

MAYIS 2003

ÖNSÖZ

Büyük güçlerde enerji üretimi ve üretilen enerjinin çok uzun tüketim bölgelerine iletimi, teknolojik ve ekonomik bakimdan ancak yüksek gerilim ile mümkündür. Iletim gerilimin yükselmesiyle birlikte, elektriksel alanda önem kazanmaktadir. Yüksek gerilim tekniginde, elektrot sistemlerini olusturan elektrot ve izolatör gibi elemanlar, elektriksel alani etkileyen en önemli etkenlerdir. Elektrot izolatörün sekli ve yerlesimi, ekonomik ve etkili bir elektrot sistemi tasarimi açisindan çok önemlidir.

Bu tez çalismasinda, farkli izolatör tipleri için darbe atlama gerilimi belirlenmesi, elektrot içim optimizasyonu ve izolatör konum açisi optimizasyonu problemlerinin çözülmesinde yeni bir yaklasim, Yapay Sinir Agi algoritmasi kullanilmistir. Yapay sinir agi kullanilmasi, sonuçlara deneysel ve sayisal yöntemlere göre çok daha hizli ve ekonomik olarak ulasilmasini saglar.

Bana bu çalismayi yapma olanagi veren, çalismalarimi büyük bir sabir ve özenle inceleyerek, karsilastigim zorluklari asmak için bana yol gösteren degerli hocam Doç. Dr. Özcan Kalenderli'ye, çalismalarini benimle paylasarak, tezimin ilerlemesine yardimci olan Yük. Müh. Hayri Yildirim'a, ayrica çalismalarimin her asamasinda bana destek olan arkadaslarim ve aileme içten tesekkürlerimi sunarim.

Suna BOLAT Mayis, 2003

IÇINDEKILER

ÖNSÖZ ... **ii**
KISALTMALAR ... **vi**
TABLO LISTESI ... **vii**
SEKIL LISTESI ... **viii**
SEMBOL LISTESI ... **x**
ÖZET ... **xi**
SUMMARY ... **xii**

1 GIRIS ... **1**

2 YAPAY SINIR AGI NEDIR? ... **4**
2.1 Giris ... 4
2.2 Insan Beyni ... 5
2.3 Yapay Sinir Aglarinin Tarihsel Gelisimi ... 7
2.4 Yapay Sinir Aglarinin Özellikleri ... 9
2.4.1 Dogrusal Olmama ... 10
2.4.2 Ögrenme ... 10
2.4.3 Genelleme ... 10
2.4.4 Uyarlanabilirlik ... 10
2.4.5 Dagitilmis Birlesik Bellek ... 11
2.4.6 Hata Toleransi ... 11
2.4.7 Paralel Islem Yapma ... 11
2.4.8 Analiz Ve Tasarim Kolayligi ... 11
2.5 Nöron Modelleri ... 11
2.5.1 Aktivasyon Fonksiyonlari ... 13
2.6 Ag Yapilari ... 15
2.6.1 Tek Katmanli Ileri Beslemeli Aglar ... 15
2.6.2 Çok Katmanli Ileri Beslemeli Aglar ... 16
2.6.3 Yinelemeli Aglar ... 17
2.7 Yapay Sinir Aginin Mühendislik Uygulamalari ... 18
2.7.1 Kimya Mühendisligi ... 18
2.7.2 Insaat Ve Yapi Mühendisligi ... 18
2.7.3 Elektrik Ve Elektronik Mühendisligi ... 18
2.7.4 Imalat Ve Makine Mühendisligi ... 19
2.7.5 Sistem Ve Kontrol Mühendisligi ... 19
2.7.6 Anten Ve Uygulamalari ... 19

3 YAPAY SINIR AGLARINDA ÖGRENME ... **20**
3.1 Giris ... 20
3.2 Ögrenme Sekilleri ... 21
3.2.1 Egiticili Ögrenme ... 21
3.2.2 Egiticisiz Ögrenme ... 22

3.2.2.1 Egiticisiz Ögrenme 22
3.2.2.2 Özdenetimli Ögrenme 22
3.3 Ögrenme Kurallari 23
3.3.1 Hata Düzeltmeli Ögrenme 23
3.3.2 Bellek Tabanli Ögrenme 25
3.3.3 Hebb Ögrenme Kurali 25
3.3.4 Yarismali Ögrenme 26
3.3.5 Boltzman Ögrenmesi 27
3.4 Yapay Sinir Aglarinda Çok Kullanilan Ögrenme Algoritmalari 28
3.4.1 Genlikte Ayrik Algilayici (Perseptron) Ögrenme Kurali 28
3.4.2 Genlikte Sürekli Algilayici (ADALINE) Ögrenme Kurali 30
3.4.3 Katmanli Genlikte Sürekli Algilayici – MADALINE (Multiple-ADALINE) Ögrenme Kurali 30
3.4.4 Çok Katmanli Perseptronlarda Ögrenme Algoritmalari 31
3.4.4.1 Geriye Yayilim (Back Propagation) Algoritmasi 32
3.4.4.2 Delta – Bar – Delta Algoritmasi 33
3.4.4.3 Baglasimsiz Yakinsama (Decoupled Momentum) Algoritmasi 35
3.4.4.4 Hizli Yayilim (Quick Propagation) Algoritmasi 36
3.5 Diger Yapay Sinir Aglari Ve Ögrenme Algoritmalari 38
3.5.1 Vektör Kuantalama (Learning Vector Quantization) 38
3.5.2 Hopfield Agi 39
3.5.3 Elman Ve Jordan Aglari 40
3.5.4 Kohonen Agi 42
3.5.5 Uyarlamali Rezonans (Adaptive Resonance Theory – ART) Teorisi 43

4 GERIYE YAYILIM (BACK PROPAGATION) ALGORITMASI 45
4.1 Giris 45
4.2 Standart Geriye Yayilim Algoritmasi (Dik Inis Algoritmasi – Gradient Descent Algorithm) 46
4.2.1 Hatanin Minimizasyonu 46
4.2.2 Yapay Sinir Aglarinda Ögrenme Ve Ezberleme Problemi 47
4.2.3 Geriye Yayilim Algoritmasinda Agirliklarin Güncellenmesi 48
4.2.3.1 Aktivasyon Fonksiyonunun Türevleri 48
4.2.3.2 Çikis Katmanindaki Agirliklarin Güncellenmesi 49
4.2.3.3 Ara Katman Agirliklarinin Güncellenmesi 52
4.2.4 Geriye Yayilim Algoritmasinda Gizli Katman Nöron Sayisinin Belirlenmesi 55
4.3 Hizli Egitim 55
4.3.1 Degisen Ögrenme Hizi 55
4.3.2 Esnek Geriye Yayilim (Resilient Back Propagation) Algoritmasi 56
4.3.3 Eslenik Gradyen Algoritmasi 58
4.3.3.1 Fletcher – Reeves Güncellemesi 59
4.3.3.2 Polak – Ribiére Güncellemesi 60
4.3.3.3 Powell – Beale 60
4.3.3.4 Orantili Eslenik Gradyen 61
4.3.4 Newton Ögrenme Algoritmalari 61
4.3.5 Levenberg – Marquardt Algoritmasi 62

5 YAPAY SINIR AGI ILE YÜKSEK GERILIM IZOLATÖRLERININ DARBE ATLAMA GERILIMLERININ BULUNMASI 64
5.1 Giris 64
5.2 Atlama Gerilimi Deneyi 65
5.3 Uygulamada Kullanilan Yapay Sinir Agi 67
5.3.1 Egitim Kümesinin Normalizasyonu 69
5.3.2 Yapay Sinir Aginda Kullanilan Parametrelerin Bulunmasi 69

5.3.3 Algoritmanin Uygulanmasi..... 72
5.4 Sonuç..... 73

6 YAPAY SINIR AGI ILE ELEKTROT BIÇIM OPTIMIZASYONU..... 75
6.1 Giris..... 75
6.2 Elektrot Biçim Optimizasyonu..... 76
6.3 Uygulamada Kullanilan Yapay Sinir Agi Yapisi..... 77
6.4 Egitim Kümesinin Giris – Çikis Örüntülerinin Elde Edilmesi..... 78
6.4.1 Egitim Kümesi..... 78
6.5 Algoritmanin Uygulanmasi..... 81
6.6 Sonuç..... 83

7 YAPAY SINIR AGI ILE IZOLATÖR KONUM AÇISI OPTIMIZASYONU..... 85
7.1 Giris..... 85
7.2 Gaz Yalitimli Sistemlerde Ara Tutucular..... 86
7.3 Uygulamada Kullanilan Yapay Sinir Agi..... 87
7.4 Egitim Kümesinin Giris – Çikis Örüntülerinin Elde Edilmesi..... 87
7.4.1 Egitim Kümeleri..... 89
7.5 Algoritmanin Uygulanmasi..... 91
7.5.1 Epoksi Reçine Izolatörün Konum Açisinin Yapay Sinir Agi Ile Bulunmasi..... 92
7.5.2 Cam Izolatörün Konum Açisinin Yapay Sinir Agi Ile Bulunmasi..... 94
7.5.3 Porselen Izolatörün Konum Açisinin Yapay Sinir Agi Ile Bulunmasi..... 96
7.6 Sonuç..... 98

8 SONUÇLAR VE ÖNERILER..... 100

KAYNAKLAR..... 103

EK A..... 109

EK B..... 121
EK B1..... 122
EK B2..... 123
EK B3..... 125

ÖZGEÇMIS..... 127

KISALTMALAR

YSA	: Yapay sinir agi
Perseptron	: Genlikte ayrik algilayici (perceptron)
ADALINE	: Genlikte sürekli algilayici (adaptif linear neuron)
MADALINE	: Katmanli genlikte sürekli algilayici (Multiple-Adaline)
SOM	: Özdenetimli harita (Self Organizing Feature Map)
ART	: Uyarlamali rezonans (Adaptive Resonance Theory) teorisi
rms	: Hatanin etkin degeri (root-mean-squared error)
sse	: Toplam karesel hata (sum squared error)
mse	: Ortalama karesel hata (mean squared error)
mae	: Ortalama mutlak hata (mean absolute error)

TABLO LISTESI

Sayfa No

Tablo 5.1	**:** Darbe atlama geriliminin YSA ile belirlenmesinde kullanilan egitim kümesi.....	68
Tablo 5.2	**:** Ögrenme hizinin hatalara etkisi.....	71
Tablo 5.3	**:** Momentum katsayisinin hatalara etkisi.....	71
Tablo 5.4	**:** Gizli katmandaki nöron sayisinin hatalara etkisi.....	72
Tablo 5.5	**:** YSA sonuçlari ile gerçek degerlerin karsilastirilmasina bir örnek	73
Tablo 6.1	**:** Çubuk – düzlem elektrot sistemi için giris örüntü vektörleri.....	80
Tablo 6.2	**:** Çubuk – düzlem elektrot sistemi için çikis örüntü vektörleri.....	81
Tablo 6.3	**:** YSA sonuçlari ile gerçek degerlerin karsilastirilmasina bir örnek	82
Tablo 6.4	**:** Kritik bölgenin koordinatlari.....	83
Tablo 7.1	**:** Epoksi reçine izolatör için egitim kümesi.....	90
Tablo 7.2	**:** Cam izolatör için egitim kümesi.....	90
Tablo 7.3	**:** Porselen izolatörün egitim kümesi.....	91
Tablo 7.4	**:** Epoksi reçine izolatör için momentum katsayisinin hatalara etkisi.....	93
Tablo 7.5	**:** Epoksi reçine izolatör için ögrenme hizinin hatalara etkisi.....	93
Tablo 7.6	**:** Epoksi reçine izolatör için gizli katmandaki nöron sayisinin hatalara etkisi.....	93
Tablo 7.7	**:** Cam izolatör için momentum katsayisinin hatalara etkisi.....	95
Tablo 7.8	**:** Cam izolatör için ögrenme hizinin hatalara etkisi.....	95
Tablo 7.9	**:** Cam izolatör için gizli katman nöron sayisinin hatalara etkisi.....	95
Tablo 7.10	**:** Porselen izolatör için momentum katsayisinin hatalara etkisi.....	97
Tablo 7.11	**:** Porselen izolatör için ögrenme hizinin hatalara etkisi.....	97
Tablo 7.12	**:** Porselen izolatör için gizli katman nöron sayisinin hatalara etkisi.....	97

SEKIL LISTESI

Sayfa No

Sekil 2.1 : Biyolojik sinir sisteminin blok diyagram gösterimi..... 5
Sekil 2.2 : Biyolojik sinir hücresi (nöron) yapisi 6
Sekil 2.3 : Sinir hücresinde iletim..... 7
Sekil 2.4 : Bir nöronun dogrusal olmayan modeli..... 12
Sekil 2.5 : Esik fonksiyonu..... 13
Sekil 2.6 : Parçali dogrusal fonksiyon..... 14
Sekil 2.7 : Lojistik sigmoid fonksiyonu..... 14
Sekil 2.8 : Tek katmanli ileri beslemeli ag..... 16
Sekil 2.9 : Bir gizli katmanli, ileri beslemeli ag..... 17
Sekil 2.10 : Yinelemeli yapay sinir agi..... 19
Sekil 3.1 : Egiticili ögrenmenin blok diyagrami..... 21
Sekil 3.2 : Egiticisiz ögrenmenin blok diyagrami..... 22
Sekil 3.3 : Özdenetimli ögrenmenin blok diyagrami..... 22
Sekil 3.4 : Hata düzeltmeli ögrenme..... 23
Sekil 3.5 : Basit bir yarismali ögrenme aginin mimari yapisi..... 26
Sekil 3.6 : Genlikte Ayrik Algilayici (Perseptron) yapisi..... 29
Sekil 3.7 : Genlikte sürekli algilayici (ADALINE) yapisi..... 30
Sekil 3.8 : ADALINE girisi olan MADALINE yapisi..... 30
Sekil 3.9 : Çok katmanli yapay sinir agi..... 30
Sekil 3.10 : Vektör kuantalama agi..... 38
Sekil 3.11 : Hopfield agi..... 40
Sekil 3.12 : a) Elman agi, b) Jordan agi..... 41
Sekil 3.13 : Kohonen agi..... 43
Sekil 3.14 : ART-1 agi..... 44
Sekil 4.1 : Hata minimizasyonun geometrik yorumu..... 47
Sekil 4.2 : Egitimin durdurulmasi..... 48
Sekil 4.3 : Çikis katmanindaki j nöronunun isaret akis diyagrami..... 50
Sekil 4.4 : j gizli katmanina bagli k çikis nöronunun isaret akis diyagrami.. 52
Sekil 5.1 : Atlama deneyinde kullanilan yüksek gerilim izolatörleri..... 65
Sekil 5.2 : Deney düzeni..... 66
Sekil 5.3 : Toplam karesel hatanin iterasyon sayisina göre degisimi..... 72
Sekil 6.1 : Yük benzetim yöntemiyle hesaplanan potansiyel hatasi..... 78
Sekil 6.2 : Elektrik alanin uzakliga göre degisimi..... 79
Sekil 6.3 : Çubuk – Düzlem elektrot sistemi..... 79
Sekil 6.4 : Ortalama karesel hatanin iterasyon sayisina göre degisimi..... 83

Sekil 7.1 **:** Gaz yalitimli sistem içindeki ara tutucu olarak kullanilan izolatör..... 86
Sekil 7.2 **:** Izolatör yüzeyindeki elektriksel alanlar ve bilesenleri..... 87
Sekil 7.3 **:** Epoksi reçine izolatör için toplam karesel hatanin iterasyon sayisina göre degisimi..... 94
Sekil 7.4 **:** Cam izolatör için toplam karesel hatanin iterasyon sayisina göre degisimi..... 96
Sekil 7.5 **:** Porselen izolatör için hatanin iterasyon sayisina göre degisimi. 98
Sekil A.1 **:** Üç adet noktasal yük ve potansiyeli hesaplanacak olan A_i noktasi..... 113
Sekil A.2 **:** Çok yalitkanli bir sistemin benzetimi..... 118

SEMBOL LISTESI

x	: Ag girisi
w	: Uyarlanabilir agirliklar
b	: Referans degeri (bias)
f	: Aktivasyon fonksiyonu
v	: Aktivasyon fonksiyonunun girisi
u	: Net giris
y	: Ag çikisi
d	: Istenen çikis
e	: Hata isareti
E	: Performans fonksiyonu
E_{ort}	: Ortalama hata
η	: Ögrenme hizi
a	: Momentum katsayisi
δ	: Yerel gradyen
$\bar{\delta}$	: Gradyenin ortalamasi
e	: Konveks agirlik faktörü
?	: Ögrenme katsayisi artma faktörü
ß	: Ögrenme katsayisi azaltma faktörü
f	: Keskin sinirlama (hard limiting) fonksiyonu
η_{min} , η_{max}	: Adim araligi sinirlari
δ^+	: Azalma faktörü
δ^-	: Azalma faktörü
H	: Hessian matrisi
J	: Jakobien matrisi
µ	: Marquardt parametresi
t	: Sicaklik
p	: Basinç
U	: Gerilim
$\vec{E}_1, \vec{E}_2$	: Elektrik alani vektörleri
$\vec{D}_1$, $\vec{D}_2$	: Deplasman vektörleri
a_1 , a_2	: Elektrik alani vektörlerinin sinir yüzeye dik normalle yaptiklari açilar
E_{t1} , E_{t2}	: Elektrik alanlarin tegetsel bilesenleri
E_{n1} , E_{n2}	: Elektrik alanlarin normal bilesenleri
D_{t1} , D_{t2}	: Deplasman vektörlerinin tegetsel bilesenleri
D_{n1} , D_{n2}	: Deplasman vektörlerinin normal bilesenleri
e_{r1} , e_{r2}	: Bagil dielektrik sabitleri
a	: Elektrot açikligi

Yapay Sinir Agi ile Elektrot ve Izolatör Biçim Optimizasyonu

ÖZET

Yüksek gerilim elektrot sistemlerinin etkin tasarimi, sistemi olusturan elektrot ve izolatör gibi elemanlar hakkinda bilgileri gerektirir. Bu tez çalismasinda, bir elektrot sisteminde izolatör varliginin ve tipinin atlama gerilimine etkisi, sistemi olusturan elektrotlarin biçiminin elektriksel alana etkisi ve kullanilan izolatörün konum açisinin elektrik alan bilesenlerine etkisi optimizasyon problemi olarak, yapay sinir agi ile incelenmistir.

Farkli basinç ve ortam kosullari altinda bulunan yüksek gerilim izolatörlerinin, darbe atlama geriliminin yeni bir yaklasimla, yapay sinir agi ile incelenmesi 4. bölümde sunulmustur. Kullanilan yapay sinir agi yapisinin tasarlanmasi için yapilan çalismalar, ayrintili olarak açiklanmistir. Yapay sinir aginin egitilmesi için kullanilan veriler, deneysel çalisma sonucunda elde edilmistir. Farkli basinç, sicaklik, bagil nem ve izolatör tipi için elde edilen veriler ile geriye yayilim algoritmasi ile egitildiginde, atlama gerilimi degerlerinin çok küçük bir hata ile belirlendigi gözlenmistir.

Elektrot sistemlerinde, yalitkanda düzgün elektrik alan dagilimi saglamak, sistemin güvenilirligi ve ömrü açisindan çok önemlidir. Düzgün olmayan alan dagilimi, yalitkanda atlama veya kismi bosalmalarin meydana gelmesine neden olur. Bu sebeple, elektrot yüzeyinde düzgün alan dagilimi saglayabilecek optimum elektrot biçimi kullanilmalidir. Bu çalismada, çubuk – düzlem elektrot sisteminde, çubuk elektrotun biçimi yapay sinir agi ile optimize edilerek, düzgün alan dagilimi saglanmistir. Bu amaçla yapay sinir agi, girisinde elektrik alan degerleri, çikisinda ise r-koordinatlari bulunan bir egitim kümesi ile egitilmistir. Ögrenme sürecinde, geriye yayilim algoritmasinin hizli türlerinden olan Levenberg – Marquardt algoritmasi kullanilmistir. Egitimin tamamlanmasindan sonra uygulanan test kümesi ile, yüksek gerilim tekniginde elektrot biçim optimizasyonu probleminin yapay sinir agi ile çözümünü, çok etkin ve hizli oldugu görülmüstür.

Çalismada ayrica, özellikle gaz yalitimli sistemlerde ara tutucu olarak kullanilan dogrusal egimli izolatörlerin optimum konum açisi belirlenmistir. Elektrotlarin yerlesim sekillerinden ve tabakali elektrot sistemlerinde sinir yüzeyde kirilma ilkesinde yararlanilarak elektrik alan hesaplamalari yapilmis ve elektrot açikligi ve konum açisi degerleri ile birlikte yapay sinir agina tanitilmistir. Geriye yayilim algoritmasi ile egitilen yapay sinir agina, egitim sonrasinda, egitim kümesinin disindan bir test kümesi uygulanarak, yapay sinir aginin güvenilirligi sinanmistir.

Çalismalar, yüksek gerilim teknigindeki bu tür problemler için, yapay sinir agi kullanilmasinin genel, etkili, hizli ve ekonomik olarak, sonuçlara ulasilmasini sagladigini göstermistir.

Electrode and Insulator Contour Optimization By Artificial Neural Networks

SUMMARY

Designing a cost-effective high voltage electrode system with high performance requires knowledge about the devices, which composed the electrode system, such as, insulator and electrode. In this thesis, effects of insulator shape, electrode configuration and insulator position on flashover voltages and, electric fields of high voltage insulator, and effects of electrode contour on stress distribution along the electrode contour are examined to solve optimization problems by using artificial neural networks.

Determination of impulse flashover voltages of post insulators under different pressure and ambient conditions using by a new approach, artificial neural networks has been presented Chapter 4. Studies on design of the artificial neural network that used in this study have been reported in detail. Training data used in artificial neural network is obtained from an experimental study of impulse flashover characteristics of cast resin support insulators under low air pressure in dry air. Experimental data obtained for different pressure, temperature, relative humidity and type of insulator is used to train in the neural network. Artificial neural network can give results with mean absolute error of 0,9 % when compared with experimentally obtained results. The results of computation show that the trained artificial neural network can give flashover voltage very efficiently and accurately.

To obtain homogen electric field in any insulation is important for electrical reliability and life of system. Otherwise, electric field is non-homogen and breakdown or partial discharge phenomena early becomes in the insulation. Therefore, electric fields on electrode contours and in insulation are calculated and optimized. In the determination of electric fields is used various methods as analytical, numerical or experimental methods. In this study, electrical fields in rod – plane electrode system are calculated by Charge Simulation Method and electrode optimization is performed by using artificial neural network. By optimizing the rod electrode contour, uniform stress distribution has been obtained along the electrode surface. To optimize the rod electrode contour, a multilayer feedforward artificial neural network with Levenberg – Marquardt algorithm. Levenberg – Marquardt algorithm is one of the faster back propagation algorithms, which reaches the desired results very fast. The artificial neural network is first trained with the results of a rod – plane electrode system. Then the trained network is used to give an optimized electrode contour in such a way that a desired stress distribution is obtained on the electrode surface. After the training is completed, a test case, which is within the range of input patterns but not included in the training set is supplied to the network. It shows that, the method can be used efficiently for optimization of electrode contour in high voltage techniques.

Also, in this thesis, determination of optimum contact angle of the insulators having linear contour that especially used in gas insulated systems as spacer has been presented. Electric fields have been calculated according to electrode positions and diffraction principle on boundaries in insulation with multidielectric. Values of calculated electric fields, electrode spacing and contact angle have been introduced to the artificial neural network. External test data has been applied to the artificial

neural network trained with back propagation algorithm to test the reliability of the network.

Presented studies show that, using artificial neural network obtains general, cost-effective, fast results for the flashover voltage, field and optimization problems in high voltage technology.

1 GİRİŞ

Teknolojinin gelişmesi ve sanayileşmenin hızlanması, elektrik enerjisine olan ihtiyacı büyük ölçüde arttırmıştır. Günümüzde her şekilde ihtiyaç duyduğumuz elektrik enerjisi, üretim yerlerinden tüketicilere ulaştırılmalıdır. Üretim ve tüketim yerleri arasındaki uzaklıklar nedeniyle, enerji ancak yüksek gerilim kullanılarak taşınabilir. Bu zorunluluk, yüksek gerilim tekniğinin gelişmesini sağlamıştır.

Yüksek gerilim tekniğindeki gelişmeler, kullanılan yalıtkan ve iletken malzemelerin geliştirilmesi ve kullanılan sistemler üzerindeki çalışmaların arttırılmasına bağlanabilir. Yüksek gerilim elektrot sistemlerinin, elektriksel alanlarının, teorik ve deneysel olarak ayrıntılı olarak incelenmesi ve yalıtkan ortamlarda boşalma olaylarını açıklayacak araştırmaların yapılması, bu çalışmaların başında gelir. Bu tür çalışmalar arttırıldıkça, yüksek gerilim elektrot ve izolatörlerinin önemi daha iyi anlaşılmıştır. Yüksek gerilim tekniğinde önemli yer tutan bu elemanların, en etkin şekilde çalıştırılabilmesi için araştırmalar yapılmaktadır.

Elektrik alanın etkisinde bulunan elektrot sistemlerindeki yalıtkan malzemeler, bu alanın etkisiyle zorlanırlar. Elektriksel zorlanmalar, yüzeysel boşalma ya da yalıtkanın, varsa izolatörün delinmesiyle sonuçlanabilir. Enerji iletim hatlarının geriliminin, çeşitli yük durumları için yükselmesinden dolayı, yalıtımın daha büyük elektriksel zorlanmalar altında çalışması gerekebilir. Yüzeysel boşalmaları ve delinmeleri en aza indirebilmek ve elektrot sistemlerinden en verimli sonucu alabilmek için, yalıtım malzemesinin her yerde eşit zorlanması gerekir. Bunun için, yüzeyde düzgün bir elektrik alan dağılımı sağlanmalıdır. Düzgün olmayan alanlarda boşalma gerilimin yükseltilmesi durumunda korona şeklinde kısmi boşalmayla başlar ve daha sonra tam delinmeyle sonuçlanır. Düzgün alanda ise, delinme koşulu alanın her noktasında aynı anda gerçekleştiğinden, elektrotlardan biri üzerinden başlayan boşalma olayı, tam delinmeyle sonuçlanır. Bu nedenle düzgün alanda boşalma başlangıç gerilimi ile, yalıtkan maddenin delinme gerilimi eşittir [1].

Yüksek gerilim düzenlerinde kullanılan elektrot ve izolatörlerin şekli, yüzeydeki elektrik alanını etkileyen en önemli faktördür. Yüzeyde düzgün alan elde edebilmek için, yüksek gerilim düzenindeki elektrotların ve izolatörlerin biçiminin optimize

edilmesi gerekir. Optimum elektrot ve izolatör biçiminin kullanılması, düzgün alan sağlamanın yanında, elektrik alanının değerinin, kabul edilebilir sınırlar içinde tutulmasını da sağlar. Bu da, elektrot sisteminin ömrünün uzamasına ve yüksek gerilim araçlarının performansının artmasına yardımcı olur.

Elektrot ve izolatörlerin biçiminin optimizasyonunda, genellikle iteratif yöntemler kullanılmaktadır. İteratif yöntemlerde, optimum elektrot veya izolatör biçimi, istenen alan dağılımını elde etmek için, lineer interpolasyonla güncellenerek belirlenir. İteratif olarak biçimin değiştirilmesi yöntemi, problemden probleme farklılıklar gösterir. Bunun yanında, pek çok problem için, her iterasyonda alan hesabı yapılması gerektiğinden, sonuca ulaşmak çok zaman almaktadır. Bu çalışmada, bu tür zorluklardan kaçınmak için bir yapay sinir ağı algoritması kullanılmıştır. Yapay sinir ağı kullanılarak, herhangi bir biçim optimizasyonu problemi için, genel bir çözüme ulaşmak mümkündür. Ayrıca, yapay sinir ağının eğitiminde kullanılacak veriler için, sınırlı sayıda alan hesabı yapmak yeterlidir. Böylece, iteratif yöntemlerin her adımda yapılan alan hesaplamalarından kaynaklanan zaman alıcılığı ve bununla beraber diğer zorlukları giderilmiş olur.

Son on yıl içinde, yapay sinir ağları uygulamalarıyla ilgili çeşitli alanlarda çok geniş araştırmalar yapılmıştır. Bunun sonucu olarak da, yapay sinir ağları literatürü oldukça hızlı şekilde genişlemiştir. Yapay sinir ağları, çok geniş uygulama alanına sahiptir. Elektrik enerji sistemlerinde yapay sinir ağları, yük tahmininde, güvenlik değerlendirilmesinde, kapasite kontrolünde, alarm değerlendirilmesinde çok etkin olarak kullanılmaktadır. Yüksek gerilim sistemlerinde ise yapay sinir ağı uygulamaları, başta kısmi boşalma örüntülerinin tanınması olmak üzere, atlama gerilimi bulunması gibi her türlü boşalma olayının modellenmesinde, yıldırım tahmini ve yerinin bulunmasında, yüksek gerilim elemanlarının biçim optimizasyonunda kullanılmaktadır. Yüksek gerilim sistemlerinde yapay sinir ağı uygulamalarından önemli bir tanesi de, elektrot sistemlerinde biçim optimizasyonudur.

Elektrot ve izolatör biçim optimizasyonu, elektrodun çevresi veya izolatörün yüzeyi boyunca düzgün bir alan dağılımı elde edecek biçimi bulmak ve böylece etkin ve yalıtım maliyeti daha düşük bir yalıtım sistemi tasarlamak amacıyla yapılmaktadır. Optimizasyonun yapay sinir ağı kullanılarak yapılması da, daha kısa sürede oldukça etkin sonuçlar elde edilmesini sağlar.

Bu çalışmada, elektrot sistemlerinin iki önemli elemanı, elektrot ve izolatörle ilgili çalışmalar yapılacaktır. Çalışma temel olarak üç bölümden oluşmaktadır.

Birinci bölümde, farklı basınç ve ortam koşulları altında izolatörlerin atlama gerilimlerinin hesaplanmasında, yeni bir yaklaşım açıklanacaktır. Bu amaçla standart geriye yayılım algoritması ile eğitilen ileri beslemeli çok katmanlı bir yapay sinir ağı kullanılacaktır.

Daha sonra arada izolatörü bulunmayan bir çubuk – düzlem elektrot sistemi için, elektrot yüzeyi boyunca düzgün alan dağılımı elde edecek şekilde, optimum elektrot biçimi bulunacaktır. Bu optimizasyon probleminin çözülmesinde kullanılan yapay sinir ağı, çok katmanlı ileri beslemeli bir ağdır. Ağda kullanılacak ağırlıkların hesaplanması için, eğitim sürecinde, yakınsama probleminden etkilenmeyen Levenberg – Marquardt algoritması kullanılacaktır.

Çalışmanın üçüncü bölümünde ise, özellikle gaz yalıtımlı sistemlerde ara tutucu olarak kullanılan doğrusal biçimli mesnet izolatörünün konum açısı, yapay sinir ağı kullanılarak optimize edilecektir. Konum açısının optimizasyonu için, geriye yayılım algoritması (back-propagation) kullanılacaktır.

2 YAPAY SİNİR AĞI NEDİR?

2.1 Giriş

Beynin üstün özellikleri, bilim adamlarını üzerinde çalışmaya zorlamış ve beynin nörofiziksel yapısından esinlenerek matematiksel modeli çıkarılmaya çalışılmıştır. Beynin bütün davranışlarını tam olarak modelleyebilmek için fiziksel bileşenlerinin doğru olarak modellenmesi gerektiği düşüncesi ile çeşitli yapay hücre ve ağ modelleri geliştirilmiştir. Böylece Yapay Sinir Ağları denen yeni ve günümüz bilgisayarlarının algoritmik hesaplama yönteminden farklı bir bilim alanı ortaya çıkmıştır. Yapay sinir ağları; yapısı, bilgi işleme yöntemindeki farklılık ve uygulama alanları nedeniyle çeşitli bilim dallarının da kapsam alanına girmektedir.

Yapay sinir ağı, geleneksel dijital bilgisayarlardan tamamen farklı olarak, insan beyninin öğrenme şeklinin uygulanmaya çalışıldığı bir yöntemdir. İnsan beyni çok karmaşık, doğrusal olmayan bir bilgi işleme sistemidir. İnsan beyni, nöron adı verilen yapısal öğelerini, günümüzdeki en hızlı sayısal bilgisayardan, kat kat daha hızlı bir şekilde, kesin hesaplama yapabilme yeteneğine sahiptir. Örnek olarak, beyin algısal tanıma işini, yaklaşık olarak 100 – 200 ms'de tamamlar, böyle bir işin tamamlanması, günümüz bilgisayarlarında ise günler alır.

En genel haliyle bir yapay sinir ağı, beyinin belirli bir işi veya fonksiyonu yerine getirme yönteminin modelini tasarlayan bir makinedir. Ağ genellikle, elektronik bileşenler kullanarak tamamlanır veya bilgisayar yazılımlarıyla benzetimi yapılır. Yapay sinir ağları, öğrenme adı verilen bir süreçle, kullanışlı hesaplamalar yapmaktadır. Yapay sinir ağı, iyi bir performans elde edebilmek için, sinir hücresi veya işleme birimi adı verilen, basit hesaplama hücreleri arasında sağlam bir bağlantı kullanır. Bu bilgilere dayanarak, yapay sinir ağının tanımı şu şeklide yapılabilir (Haykin, 1999). Yapay sinir ağı, deneysel bilgiyi saklama ve kullanıma hazır hale getirme yeteneğine sahip basit işleme birimlerinden oluşan, çok yoğun, paralel ve dağılmış düzende çalışan bir işlemcidir.

Bu işlemci iki yönden insan beynine benzemektedir:

1. Ağ, bilgiyi ortamdan öğrenme yoluyla elde eder

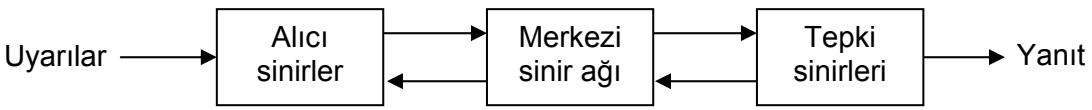
2. Gerekli bilgiyi saklamak için, sinir hücreleri arasındaki bağlantılar kullanılır.

Ağın bilgiyi ortamdan almasını sağlayan öğrenme işlemi, öğrenme algoritması adıyla bilinen, ağın sinaptik ağırlıklarını güncelleyerek istenen modele ulaşılmasını sağlayan fonksiyonlardır.

Yapay sinir ağları bilgisayarlar gibi programlanmak yerine, sinir hücreleri arasındaki bağlantı ağırlıkları, eşik değerleri ve ağ yapısı gibi parametrelerin değiştirilip, ayarlanmasıyla eğitilir. Yapay sinir ağındaki bellek ve işlemci birimleri ayrık değildir, paralel ve asenkron çalışır. Büyük boyutta fazlalıklı ve dağılmış olduğu için hataya toleranslıdır. Öğrenme sürecinde özdenetimli olan yapay sinir ağları, bilgiyi sinir hücrelerini birleştiren bağlantılarda saklayabilir ve bilgi değiştirilebilir. Yapay sinir ağlarında işlem hızı mili saniyeler mertebesindedir [5].

2.2 İnsan Beyni

Biyolojik sinir sistemi, merkezinde sürekli olarak bilgiyi alan, yorumlayan ve uygun bir karar üreten beynin (merkezi sinir ağı) bulunduğu 3 katmanlı bir sistem olarak açıklanır. Alıcı sinirler (receptor) organizma içerisinden ya da dış ortamlardan algıladıkları uyarıları, beyine bilgi ileten elektriksel sinyallere dönüştürür. Tepki sinirleri (effector) ise, beyinin ürettiği elektriksel darbeleri organizma çıktısı olarak uygun tepkilere dönüştürür. Şekil (2.1) de bir sinir sisteminin blok gösterimi verilmiştir. İnsan sinir sistemi, şekil (2.1) de gösterildiği gibi, üç aşamalı bir sistem olarak incelenebilir (Arbib,1987).

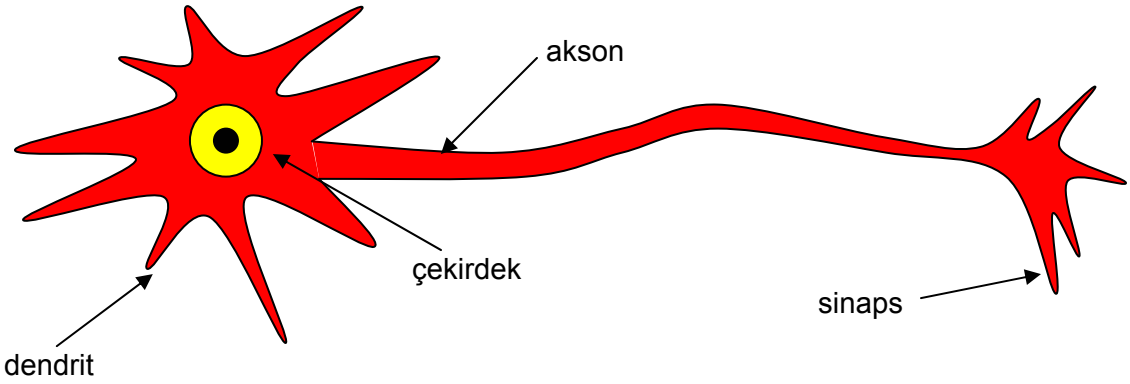


Şekil 2.1 Biyolojik sinir sisteminin blok diyagram gösterimi

Sistemdeki merkezi sinir ağı beyindir, sürekli olarak bilgiyi alır, fark eder ve kararlara uygun hale getirir. Merkezi sinir ağında bilgiler, alıcı ve tepki sinirleri arasında ileri ve geri besleme yönünde değerlendirilerek uygun tepkiler üretilir. Alıcı, insan vücudundan veya çevreden gelen uyarıyı, elektriksel darbelere çevirerek, bilgiyi beyine iletir. Verici, hücresel ağ tarafından oluşturulan elektriksel darbeleri, sistem

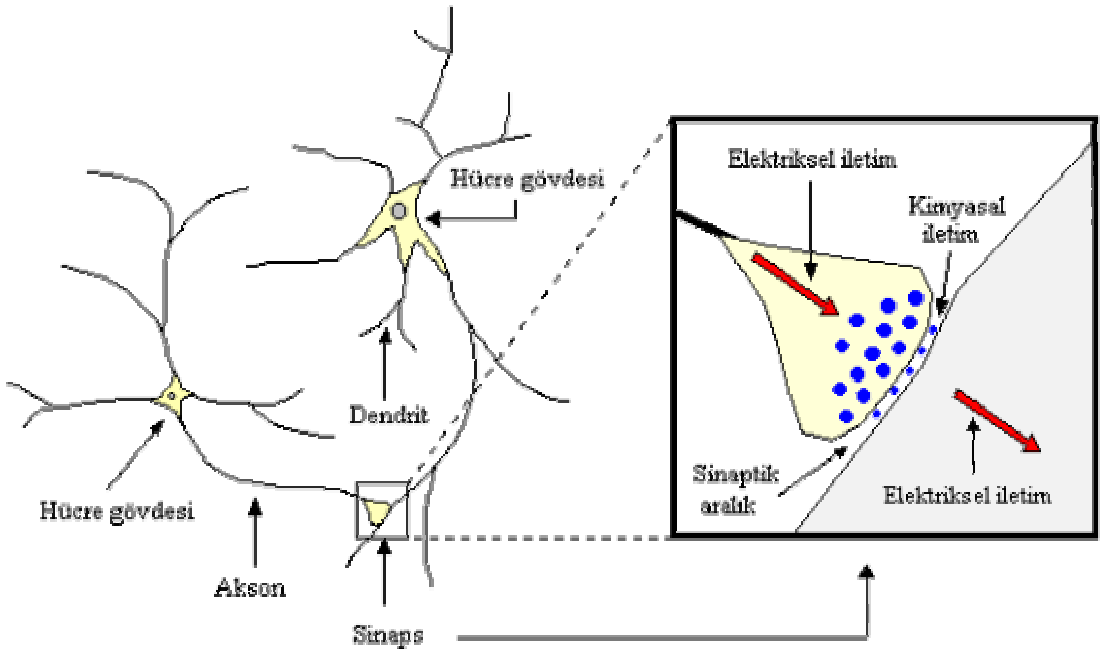
çıkışı olarak fark edilen yanıtlara dönüştürür. Bu yönüyle biyolojik sinir sistemi, kapalı çevrim denetim sisteminin karakteristiklerini taşır [5-6].

Canlılardaki sinir sistemi hücrelerine nöron adı verilir ve insan beyninde yaklaşık 10 milyar sinir hücresi olduğu tahmin edilmektedir. Beyinin yapısal öğesi olarak nöronları düşünmek, beyini daha kolay anlaşılabilir bir hale getirmiştir. Sinir hücresi; hücre gövdesi, dendritler ve aksonlar olmak üzere 3 bileşenden meydana gelir. Dendritler, diğer hücrelerden aldığı bilgileri hücre gövdesine bir ağaç yapısı şeklinde ince yollarla iletir. Aksonlar ise elektriksel darbeler şeklindeki bilgiyi hücreden dışarı taşıyan daha uzun bir yoldur. Aksonların bitimi, ince yollara ayrılabilir ve bu yollar, diğer hücreler için dendritleri oluşturur. Şekil (2.2) de görüldüğü gibi akson-dendrit bağlantı elemanına sinaps denir. Biyolojik nöronun dört önemli parçası, yapay sinir ağlarında da benzer işlevlerle kullanılmıştır.



Şekil 2.2 Biyolojik sinir hücresi (nöron) yapısı

Şekil (2.2) de görülen biyolojik sinir hücresinde, sinir hücresinin dendritlerine ulaşan bilgiler toplanır ve sinir hücresinin sonundaki sinapslara, akson boyunca iletilir (Şekil 2.3). Bu bilgiler genellikle elektriksel darbelerdir, ancak sinapstaki kimyasal ileticilerden etkilenir. Dendritler bu kimyasal iletimi alıp, çekirdeğe ulaştıracak yapıdadır. Bir nöronun dendritlerine gelen işaretler, entegre olarak veya beraber toplanarak bileşik işareti oluştururlar. Belirli bir sürede bir hücreye gelen girişlerin değeri, belirli bir eşik değerine ulaştığında hücre bir tepki üretir. Bu bilgiler, eğer gelen uyarı belirli bir eşik değerini aşarsa, diğer sinir hücresine geçirilir. Bu durumda sinir hücresi uyarılmış olur. Gelen uyarı, eşik değerini aşamazsa, bilgi daha ileri taşınamaz ve sinir hücresi pasif bırakılmış olur. İki sinir arasındaki bağlantılar birbiriyle uyumludur. Yani bağlantı yapısı büyük ölçüde değişebilir. Beynin öğrenme yeteneği, bu uyuma bağlıdır.



Şekil 2.3 Sinir hücresinde iletim

İnsan beyninin 10 milyar sinir hücresinden ve 60 trilyon sinaps bağlantısından oluştuğu düşünülürse, son derece karmaşık ve etkin bir yapı olduğu anlaşılır. Diğer taraftan bir sinir hücresinin tepki hızı, günümüz bilgisayarlarına göre oldukça yavaş olmakla birlikte duyusal bilgileri son derecede hızlı değerlendirebilmektedir. Bu nedenle insan beyni; öğrenme, birleştirme, uyarlama ve genelleştirme yeteneği nedeniyle son derece karmaşık, doğrusal olmayan ve paralel dağılmış bir bilgi işleme sistemi olarak tanımlanabilir.

Yapay sinir ağlarında amaç, insan beyninin çalışma sistemini modellemektir. Ancak biyolojik sinir hücrelerine ait işlemlerin tam olarak kopyalanması mümkün değildir. Bununla beraber teknolojik yetersizlikler de, çalışmalarda büyük ölçüde basitleşmelere gidilmesine neden olmuştur [5-10].

2.3 Yapay Sinir Ağlarının Tarihsel Gelişimi

Yapay sinir ağlarının tarihsel gelişimi incelenirken, bilgisayarlardan da söz etmek gerekir. Charles Babbage adlı matematik profesörü, ilk çözümleyici makineyi (Analytical Engine) yapmıştır. Çözümleyici makine, delikli kartlarla komutlar verildiğinde, herhangi bir aritmetik işlemi yapabilen, sayıları saklayacak bellek birimine sahip bir makinedir. İşlemlerin ard arda ve sırasıyla yapılmasını sağlayan ardışık kontrol gibi özelliklerinden dolayı, çağdaş bilgisayarın öncüsü olarak görülür.

Çözümleyici makinelerdeki gelişmelerden sonra, 1943 – 1946 yıllarında, Jon Von Neuman tarafından geliştirilen makine, ENIAC (Electronic Numerical Integrator and Computer), bilgiyi adresle belirli hafızada saklama, aritmetik – mantık birimlerine sahip olma, hataya toleranslı olmama gibi özellikleriyle kendini göstermiştir. Von Neuman makinesi, yazılıma bağlı olarak programlanan, bellek ve işlemci üniteleri farklı, işlem hızı nano saniyeler mertebesinde olan bir makinedir.

Von Neuman'la aynı yıllarda, matematikçi Pitts ile psikiyatr ve nörolog olan Mc Culloch, basit yapılı lojik fonksiyonlarla ilk sinir hücresi modelini oluşturdular. 1950 yılında, Von Neuman, Mc Culloch – Pitts modelini geliştirmiştir (EDVAC – Electronic Discrete Variable Automatic Computer). 1949 yılında, Hebb, kendi adıyla anılan öğrenme kuramını, yapay sinir ağlarına uyguladı. Öğrenen, uyarlamalı sistemlere esin kaynağı olan öğrenme kuramı şöyledir: "A hücresinin bir aksonu, B hücresini uyaracak ve tekrarlı veya sürekli olarak tetikleyecek kadar yakında ise, hücrelerde B'yi tetikleyen A'nın etkinliğini arttıracak bir büyüme işlemi veya metabolik değişiklik olur". 1956 yılında ortaya atılan bir teoriyle, benzeşim sonuçları elde edildi. Marvin Minsky, 1951 yılında kendi kendine ağırlıkları ayarlayabilen bir algoritma (Snark) sundu, fakat bu algoritma, işe yarar hiçbir bilgi işleme fonksiyonunu gerçekleyememiştir. Marvin daha sonra, 1954 yılında Takviyeli Öğrenme (RL – Reinforcement Learning) kuralını oluşturdu. 1957 yılında Bellman tarafından Dinamik Programlama oluşturulduktan sonra, yapay sinir ağlarında çok kullanılan Genlikte Ayrık Algılayıcı (Perceptron), Frank Rosenblatt tarafından literatüre tanıtıldı (1958).

Genlikte ayrık algılayıcıdan sonra ortaya çıkan ADALINE (adaptif linear neuron) algoritmasının farkı eğitim aşamasıdır. Bu algoritma, Widrow ve Hoff tarafından en küçük ortalama karesel hata, diğer adıyla gradyen azalması yöntemi, basit Doğrusal bir nörona uygulanarak geliştirilmiştir. Adaline'dan sonra Widrow MADALINE (Multiple adaline) adını verdiği, eğitilebilen, katmanlı, adaptif elemanlı bir yapay sinir ağını geliştirmiştir.

1969 yılında, Minsky ve Papert tarafından yapılan bir çalışmayla, perseptronun XOR lojik kapısını gerçekleyemediği ispatlanmıştır. 1969 – 1982 yılları arasında, yapay sinir ağlarıyla ilgili yapılan çalışmalar, karanlık bir döneme girmiştir. Bu yıllarda, Grossberg ve Kohonen tarafından öndüzenlemeli farklı iki yapı kurulmuştur.

1974 yılında Werbos, Geriye Yayılım (Back - Propagation) algoritmasını ortaya attı. Willshaw ve Malsburg 1976 yılında Özdenetimli Harita (SOM - Self Organizing

Feature Map) adındaki kümelenme algoritmasını kurdular. 1982 yılında Hopfield kendi adıyla anılan Hopfield Ağı'nı geliştirdi. 1982 yılında Kohonen, özdenetimli harita ağı kuramını beyindeki oluşumların karşılaştırmalı haritasını çıkarabilmek için ortaya koydu. Kohonen'in oluşturduğu ağ, Willshaw ve Malsburg'un oluşturduğu özdenetimli ağa çok benzemektedir. Aynı yıl Parker, geriye yayılım algoritmasını tartışmaya açtı. Genelleştirilmiş Delta Öğrenme Kuralı olarak da bilinen geriye yayılım algoritması, 1986 yılında Rumelhart ve McClelland tarafından tekrar gündeme getirilmiştir.

1980lerin ortalarında, termodinamik modellerde gelişmeler görülmüştür. Bu modellere Smolennsky'nin Uyum Teorisi ve Boltzman Makinası örnek olarak gösterilebilir.1985 yılında Ackley Hinton ve Sejnowski, Boltzman makinasını kullanarak bir öğrenme kuralı geliştirmişlerdir.

1986 yılında, Rumelhart ve McClelland'ın yazdıkları kitabın yayınlanması, yapay sinir ağlarına olan ilgiyi birden arttırmıştır. Ertesi yıl ilk yapay sinir ağları konferansı, IEEE tarafından düzenlenmiştir. Uluslararası Yapay Sinir Ağları Kuruluşu (International Neural Network Society) de aynı yıl faaliyete geçerek, 1988 yılında ilk bildirisini yayınlamıştır.

1988 yılında Broomhead ve Lowe, Radyal Tabanlı Fonksiyon Ağı (Radial Basis Function Network) adıyla yeni bir yapı oluşturdular. Ağ yapılarının değişik kombinasyonları olarak, ileri ve geri beslemeli ağ yapıları da literatüre tanıtıldı. Günümüzde, her yıl düzenlenen yapay sinir ağları konferanslarında yeni öğrenme algoritmaları, yeni mimari yapılar ortaya atılmaktadır [5].

2.4 Yapay Sinir Ağlarının Özellikleri

Yapay sinir ağlarının hesaplama ve bilgi işleme gücünü, paralel dağılmış yapısından, öğrenebilme ve genelleme yeteneğinden aldığı söylenebilir. Genelleme, eğitim ya da öğrenme sürecinde karşılaşılmayan girişler için de yapay sinir ağının uygun tepkileri üretmesi olarak tanımlanır. Bu üstün özellikleri, yapay sinir ağının karmaşık problemleri çözebilme yeteneğini gösterir. Günümüzde birçok bilim alanında yapay sinir ağları, aşağıdaki özellikleri nedeniyle etkin olmuş ve uygulama yeri bulmuştur.

2.4.1 Doğrusal Olmama

Yapay sinir ağının temel işlem elemanı olan hücre, doğrusal değildir. Dolayısıyla hücrelerin birleşmesinden meydana gelen yapay sinir ağı da doğrusal değildir ve bu özellik bütün ağa yayılmış durumdadır. Bu özelliği ile yapay sinir ağı, doğrusal olmayan karmaşık problemlerin çözümünde en önemli araç olmuştur.

2.4.2 Öğrenme

Aslında öğrenmeden kasıt, ilgili problemdeki girdi-çıktı ilişkisini en güzel tanımlayacak optimum ağırlıkların bulunmasıdır. Problemden alınan örneklerden yararlanılarak ilgili problemi kendisine uygulanan örneklerden öğrenmeye çalışır. Probleme farklı bir çözüm sağlar. Yapay sinir ağının istenen davranışı gösterebilmesi için, amaca uygun olarak ayarlanması gerekir. Bu, hücreler arasında doğru bağlantıların yapılması ve bağlantıların uygun ağırlıklara sahip olması gerektiği anlamına gelir. Yapay sinir ağının karmaşık yapısı nedeniyle, bağlantılar ve ağırlıklar önceden ayarlı olarak verilemez ya da tasarlanamaz. Bu nedenle yapay sinir ağı, istenen davranışı gösterecek şekilde, ilgilendiği problemden aldığı eğitim örneklerini kullanarak problemi öğrenmelidir.

2.4.3 Genelleme

Yapay sinir ağı, ilgilendiği problemi öğrendikten sonra eğitim sırasında karşılaşmadığı test örnekleri için de istenen sonucu üretebilir. Örneğin, karakter tanıma amacıyla eğitilmiş bir yapay sinir ağı, bozuk karakter girişlerinde de doğru karakterleri verebilir yada bir sistemin eğitilmiş yapay sinir ağı modeli, eğitim sürecinde verilmeyen giriş sinyalleri için de sistemle aynı davranışı gösterebilir.

2.4.4 Uyarlanabilirlik

Yapay sinir ağı, ilgilendiği problemdeki değişikliklere göre ağırlıklarını ayarlar. Yani, belirli bir problemi çözmek amacıyla eğitilen yapay sinir ağı, problemdeki değişimlere göre tekrar eğitilebilir, değişimler devamlı ise gerçek zamanda da eğitime devam edilebilir. Bu özelliği ile yapay sinir ağı, uyarlamalı örnek tanıma, sinyal işleme, sistem tanılama ve denetim gibi alanlarda etkin olarak kullanılır.

2.4.5 Dağıtılmış Birleşik Bellek

Yapay sinir ağlarının en önemli özelliklerinden biri de bilgiyi depolamalarıdır. Nöral hesaplamalarda bilgi ağırlıklar üzerine dağıtılmıştır. Bağlantıların ağırlıkları nöral ağın bellek birimidir. Bu ağırlıklar ağın o andaki sahip olduğu bilgiyi veya uygulanan örneklerden öğrenmiş olduğu davranışı verir. Bu bilgiler, ağdaki birçok ağırlıklar üzerine (bellek birimine) dağıtılır.

2.4.6 Hata Toleransı

Yapay sinir ağı, çok sayıda hücrenin çeşitli şekillerde bağlanmasından oluştuğundan, paralel dağılmış bir yapıya sahiptir ve ağın sahip olduğu bilgi, ağdaki bütün bağlantılar üzerine dağılmış durumdadır. Bu nedenle, eğitilmiş bir yapay sinir ağının bazı bağlantılarının hatta bazı hücrelerinin etkisiz hale gelmesi, ağın doğru bilgi üretmesini önemli ölçüde etkilemez. Bu nedenle, geleneksel yöntemlere göre hata toleransı son derece yüksektir.

2.4.7 Paralel İşlem Yapma

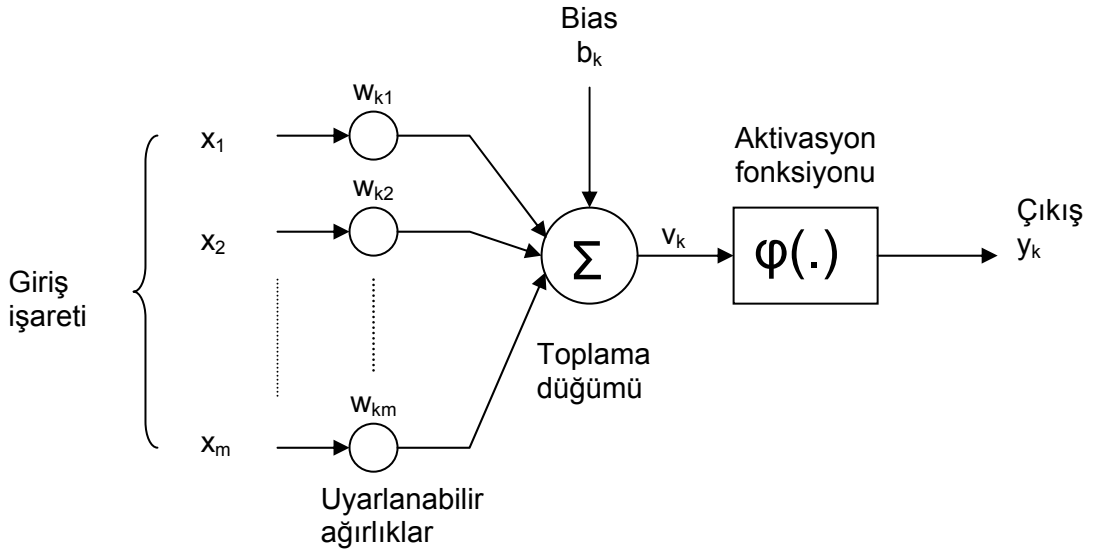
Yapay sinir ağı, paralel yapısı nedeniyle büyük ölçekli entegre devre (VLSI) teknolojisi ile gerçeklenebilir. Bu özellik, yapay sinir ağının hızlı bilgi işleme yeteneğini artırır ve örnek tanıma, işaret işleme, sistem kimliklendirme ve denetim gibi gerçek zamanlı uygulamalarda tercih edilir.

2.4.8 Analiz ve Tasarım Kolaylığı

Yapay sinir ağının temel işlem elemanı olan hücrenin yapısı ve modeli, bütün yapay sinir ağı yapılarında yaklaşık aynıdır. Dolayısıyla, yapay sinir ağının farklı uygulama alanlarındaki yapıları da standart yapıdaki bu hücrelerden oluşacaktır. Bu nedenle, farklı uygulama alanlarında kullanılan yapay sinir ağları benzer öğrenme algoritmalarını ve teorilerini paylaşabilirler. Bu özellik, problemlerin yapay sinir ağları ile çözümünde önemli bir kolaylık getirecektir.

2.5 Nöron Modelleri

Nöron, bir yapay sinir ağının işlemesinde temel olan bir bilgi işleme birimidir.



Şekil 2.4 Bir nöronun Doğrusal olmayan modeli

Şekil (2.4)'teki blok diyagramda, bir nöronun modeli görülmektedir. Bu blok diyagram, yapay sinir ağlarının tasarımında temel alınmaktadır. Nöron modelinin üç esas elemanı vardır:

1. Sinaps ya da bağlantı hatları
2. Toplayıcı
3. Aktivasyon fonksiyonu

Şekil (2.4)'te görülen model, ayrıca bir dışardan girişe sahiptir. b_k ile gösterilen, bias adı verilen referans değeri, aktivasyon fonksiyonunun net girişini pozitif ise yükseltir, negatif ise düşürür.

Matematiksel terimlerle, k nöronu iki eşitlikle ifade edilebilir:

$$u_k = \sum_{j=1}^{m} w_{kj}\, x_j \tag{2.1}$$

$$y_k = \varphi\left(v_k\right) \tag{2.2}$$

Burada, x_1, x_2, ..., x_m giriş işaretleri; w_{k1}, w_{k2}, ..., w_{km} k nöronunun sinaptik ağırlıkları; u_k giriş işaretlerine bağlı Doğrusal birleştirici çıkış; b_k bias terimi; φ(.) aktivasyon fonksiyonu ve y_k nöronun çıkış işaretidir. Şekil (2.4)'te görülen v_k, aktivasyon fonksiyonunun net girişidir. Bias yokken $v_k = u_k$'dır, kullanılması halinde ise;

$$v_k = u_k + b_k \tag{2.3}$$

olarak hesaplanır.

2.5.1 Aktivasyon Fonksiyonları

φ(v) ile gösterilen aktivasyon fonksiyonu, nöronun çıkışını, v'ye göre belirler. Aktivasyon fonksiyonlarının üç temel tipi aşağıda açıklanmıştır.

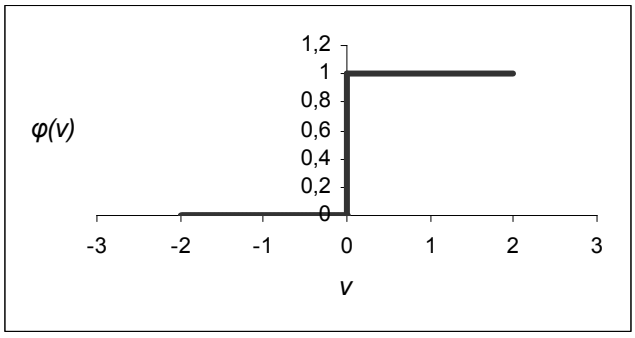
1. Eşik Fonksiyonu:

$$\varphi(v) = \begin{cases} 1, & v \geq 0 \\ 0, & v < 0 \end{cases} \tag{2.4}$$

Burada v, net girişitir ve şu şekilde ifade edilir:

$$v_k = \sum_{j=1}^{m} w_{kj}\, x_j + b_k \tag{2.5}$$

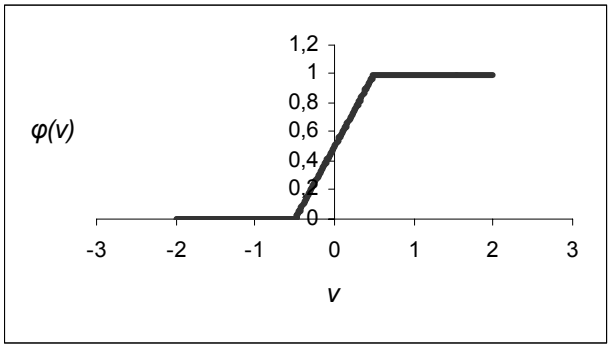
McCulloch – Pitts modeline göre, nöronun çıkışı v pozitifken 1, aksi halde 0'dır.



Şekil 2.5 Eşik fonksiyonu

2. Parçalı Doğrusal Fonksiyon:

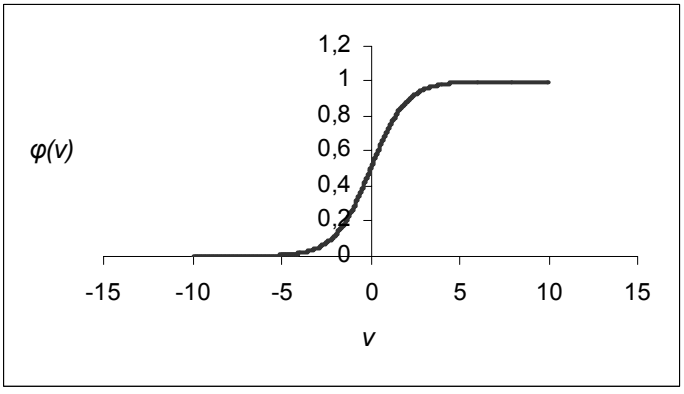
$$\varphi(v) = \begin{cases} 1, & v \geq +\frac{1}{2} \\ v, & +\frac{1}{2} > v > -\frac{1}{2} \\ 0, & v \leq -\frac{1}{2} \end{cases} \tag{2.6}$$



Şekil 2.6 Parçalı doğrusal fonksiyon

3. Sigmoid Fonksiyonu: Sigmoid fonksiyonu, yapay sinir ağlarında çok sık kullanılan, s biçimli bir fonksiyondur. Bu fonksiyon, doğrusal ve doğrusal olmayan davranışlar arasında çok iyi bir denge kurar. Aşağıda tanımlanan sigmoid fonksiyonu (Şekil 2.7), lojistik sigmoid fonksiyonudur:

$$\varphi(v) = \frac{1}{1 + e^{(-a\,v)}} \tag{2.7}$$



Şekil 2.7 Lojistik sigmoid fonksiyonu

Fonksiyonun matematiksel ifadesindeki a sabiti, eğim parametresidir. Eğim parametresinin değeri arttıkça, sigmoid fonksiyonu, eşik fonksiyonuna yaklaşır. Sigmoid fonksiyonu, değerleri 0 – 1 arasında olan, sürekli ve türevi alınabilen bir fonksiyondur. Yapay sinir ağlarında, aktivasyon fonksiyonunun türevi alınabilir olması, önemli bir özelliktir. Bu seçimin sebebi, dönüşüm işleminin analitik kontrolünü kolaylaştırmaktır. Özellikle en küçük kareler metodunu kullanan öğrenme algoritmalarında, hesaplamalar için fonksiyonun türevine ihtiyaç duyulur. Doğrusal ve eşik fonksiyonları bu özelliği taşımadıkları için, sigmoid fonksiyonu daha çok tercih edilen bir aktivasyon fonksiyonudur [5].

(2.4), (2.6) ve (2.7) eşitliklerinde tanımlanan aktivasyon fonksiyonları, 0 – 1 arasındadır. Bazı durumlarda, çıkışın –1 ve +1 aralığında olması istenir. Bu durumda eşik fonksiyonu yerine, signum fonksiyonu kullanılabilir:

$$\varphi(v) = \begin{cases} 1, & v > 0 \\ 0, & v = 0 \\ -1, & v < 0 \end{cases} \tag{2.8}$$

Sigmoid fonksiyonu yerine de, hiperbolik tanjant fonksiyonu kullanılabilir:

$$\varphi(v) = \tanh(v) \tag{2.9}$$

2.6 Ağ Yapıları

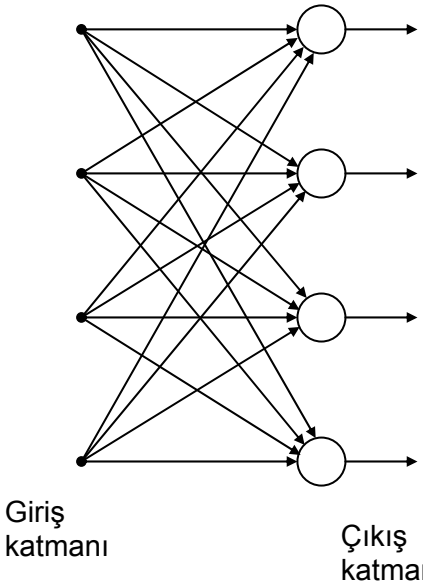
Bir yapay sinir ağının nöronları, ağı eğitmek için kullanılan öğrenme algoritmalarıyla birbirlerine bağlı yapıdadır. Öğrenme algoritmaları, yapay sinir ağının tasarımında kullanılırlar.

Genel olarak üç farklı temel ağ yapısı tanımlanabilir: Tek katmanlı ileri beslemeli ağ, çok katmanlı ileri beslemeli ağ ve yinelemeli geri beslemeli ağ.

2.6.1 Tek Katmanlı İleri Beslemeli Ağlar

Katmanlı yapay sinir ağında nöronlar, katman şekline göre dizilirler. En basit katmanlı ağ yapısı, bir giriş ve bir çıkış katmanı bulunan, iki katmanlı, ileri beslemeli ağdır. Çıkış katmanındaki herbir nöron, düzeltilebilen ağırlıklar yoluyla, bütün giriş

nöronlarından işaret alır. İleri beslemeli ağlarda, çıkıştan girişe doğru bir işaret akışı yoktur. Giriş katmanında bir hesaplama yapılmadığı için bu katman sayılmaz ve ağ "tek katmanlı" olarak adlandırılır. Şekil (2.8)'de giriş ve çıkış katmanında 4 nöron bulunan bir tek katmanlı ileri beslemeli ağ görülmektedir.



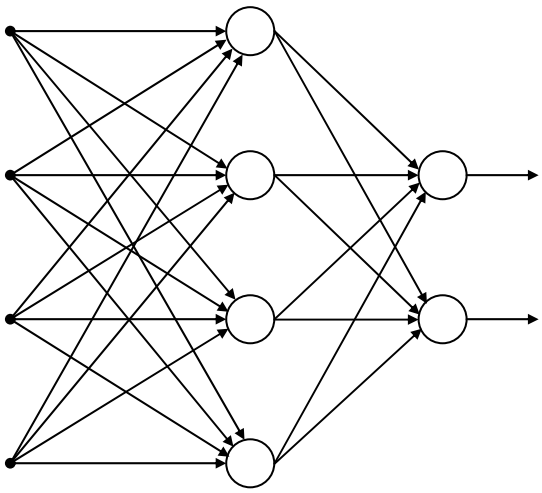
Şekil 2.8 Tek katmanlı ileri beslemeli ağ

2.6.2 Çok Katmanlı İleri Beslemeli Ağlar

İleri beslemeli yapay sinir ağlarının ikinci sınıfı olan bu ağ yapısı, bir veya birden fazla gizli katmana sahip olmasıyla farklılık gösterir. Hesaplamaların yapıldığı gizli katman, giriş ve çıkış katmanlarını kullanışlı bir şekilde bir araya getirir. Ağa bir veya daha çok gizli katman eklenmesi, tek katmanlı ağların hesaplayamadığı, Doğrusal ayrıştırılabilir olmayan fonksiyonları da hesaplamayı kolaylaştırır.

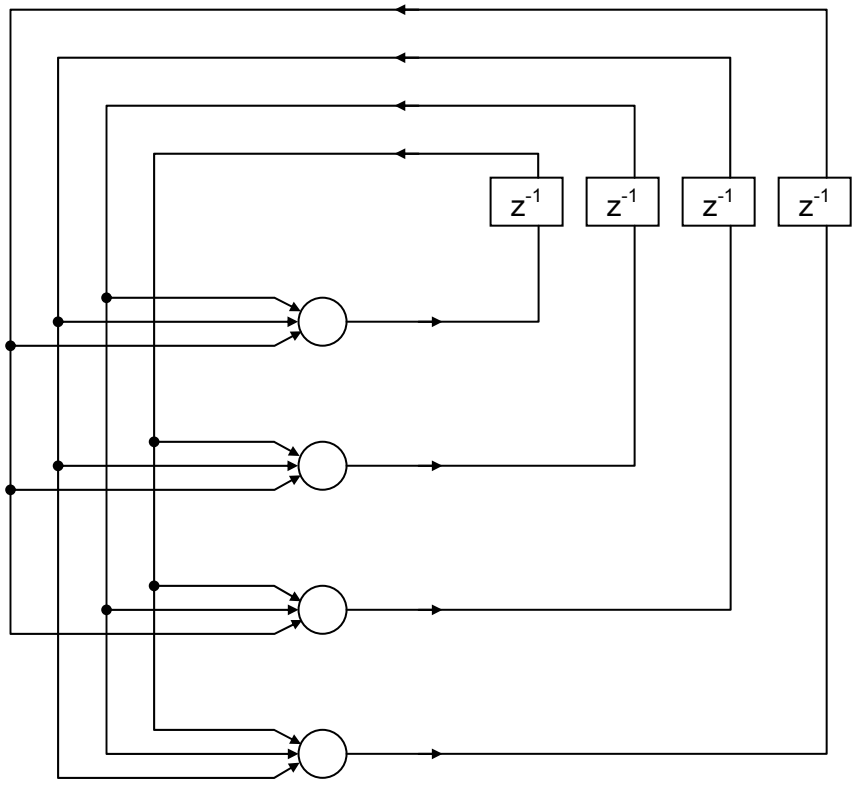
Giriş katmanındaki örüntü (giriş vektörü), ikinci katmana (birinci gizli katman) uygulanır. İkinci katmanın çıkışı, üçüncü katmana giriş olarak uygulanır ve bu işleme, ağın çıkışına kadar devam edilir. Her katmandaki nöronların girişleri, bir önceki katmanın çıkışlarıdır. Çıkış katmanındaki nöronlar, girişe uygulanan örüntülerin ağda oluşturduğu toplam yanıtı verir.

Şekil (2.9)'da, giriş katmanında ve tek gizli katmanda 4 nöronu, çıkış katmanında iki nöronu olan bir çok katmanlı ileri beslemeli ağ yapısı görülmektedir.



Şekil 2.9 Bir gizli katmanlı, ileri beslemeli ağ

2.6.3 Yinelemeli Ağlar



Şekil 2.10 Yinelemeli yapay sinir ağı (gizli katmanı ve kendi kendini beslemesi yok)

Yinelemeli yapay sinir ağında diğer yapılardan farklı olarak, bir geri besleme döngüsü bulunmaktadır. Örnek olarak bir yinelemeli yapay sinir ağı, herbir nöronu, çıkış işaretini diğer nöronların girişini besleyecek şekilde, tek katmana sahip olabilir.

Şekil (2.10)'da, kendi kendini besleyen döngüleri olmayan, gizli katmansız bir yinelemeli ağ yapısı görülmektedir.

Geri beslemenin varlığı, ağın öğrenme yeteneğini ve performansını arttırır.

2.7 Yapay Sinir Ağının Mühendislik Uygulamaları

Yapay sinir ağları, birçok önemli mühendislik problemlerin çözümünde kullanılmışlardır. Aşağıda, kimya mühendisliği, inşaat ve yapı mühendisliği, elektrik ve elektronik mühendisliği, imalat ve makine mühendisliği, sistem ve kontrol mühendisliği alanlarında bulunan uygulamalarından bazıları açıklanmıştır.

2.7.1 Kimya Mühendisliği

Kimyasal reaktör seçimi, dinamik işlemlerde hata belirlenmesi, endüstriyel polimerizasyonda eritme akış indisi tahmini, endüstriyel mayalama işleminin modellenmesi, biyokimyasal işlemlerde mikrobik konsantrasyonların tahmininde uygulanmıştır.

2.7.2 İnşaat ve Yapı Mühendisliği

Konstrüksiyon projelerinde kaynak seviyelerini belirleme, bir rezervuardan çıkışı kontrol, biyolojik bilgiler yardımıyla nehir suyu kalitesinin sınıflandırılması, nehirlerin akışının tahmin edilmesi, sonlu-eleman-temelli yapısal analiz işleminin modellenmesi, yapı malzemelerinin iç yapılarındaki çatlakların tespit edilmesi, depreme maruz betonarme çerçevelerde emniyetli yatay taşıyıcı tahmininde uygulanmıştır.

2.7.3 Elektrik ve Elektronik Mühendisliği

Hastaların alarmla kontrolü, gürültülü resimlerin kalitesini artırma, görüntülerin sıkıştırılması, gürültü filtreleme ve resimlerdeki kenar bilgisinin çıkarılması, güç sisteminde harmoniklerin tahmini, gezgin haberleşme sisteminde kanal dağıtımı, ultrasonik müziklerden objelerin sınıflandırılması, optik okuyucu sistemler için resimlerin ön işlemesi alanlarında kullanılmıştır.

2.7.4 İmalat ve Makine Mühendisliği;

Metal kesme tezgahının kontrolü, akustik salınım ve iş parçası kuvvetiyle iş parçası yatağının kontrolü ve güç tüketimi ve iş parçası ivmesi, hücreli imalat için grup teknolojisi parça gruplarının tasarımı, hareket eden nesneler için engelsiz yol planlaması, makine parametrelerinin optimizasyonu, makine arızalarının sınıflandırılması, malzemelerin ısı transferinin belirlenmesi, uçak kanat kutularının yapısının tasarımında uygulanmıştır.

2.7.5 Sistem ve Kontrol Mühendisliği

Esnek kollu robotun kontrolü, bir model helikopterin havada kontrolü, çok değişkenli (mafsallı) robotun yörünge koordinasyonu, iki sıvı tank sisteminin akış seviye kontrolü, anestezi derinliği kontrolü ve ölçülmesi, bir robot için optimal yolun bulunması, banyo suyu sıcaklığının kontrolü, endüstriyel robot kontrolü, sistem kimliklendirme gibi birçok alanda uygulanmıştır.

2.7.6 Anten ve Uygulamaları

Yapay sinir ağlarında bir dinamik öğrenme algoritması kullanılarak anten dizi elemanlarından elde edilen işaretler arasındaki faz farklılıklarının karşılaştırılmasıyla radar izleme gerçekleştirilmiştir. Mikrodalga parlaklığına bağlı jeofiziksel parametrelerin belirlenmesi, dinamik öğrenme algoritması sinir ağı kullanılarak gerçekleştirilmiştir. Uzaktan kontrollü görüntü sınıflama işlemi, danışmanlı dinamik öğrenme algoritması ve danışmansız (ART2) öğrenme algoritmaları kullanılarak yapılmıştır. Yapay sinir ağı kullanılarak anten dizi analizi yapılmıştır. Aynı zamanda, anten dizi tasarımında, genetik algoritma ve tabu araştırma algoritması kullanılmıştır. Osilatör yapay sinir ağı ile elektromagnetik işaretlerin etkileşim modelleri oluşturulmuştur [5-10].

3 YAPAY SİNİR AĞLARINDA ÖĞRENME

3.1 Giriş

Bir yapay sinir ağının en önemli özelliği, ağın çevreden öğrenme yeteneği ve öğrenme süreci boyunca performansını arttırmasıdır. Yapay sinir ağı çevreden, ağırlıkların ayarlandığı bir interaktif işlem sonucunda öğrenir. İdeal olarak ağ, öğrenme sürecinin her iterasyonunda daha bilgili hale gelir.

Burada sözü edilen bilgi, insan için depolanmış enformasyonu yorumlama; makine için, modeli öngörme veya modele uygun yanıt verme olayıdır.

Öğrenme, ağın içinde bulunduğu ortamdan etkilenerek, serbest parametrelerini (ağırlıklar, aktivasyon fonksiyonu, eşik...) değiştirmesi işlemidir. Öğrenme şekli, parametrelerin nasıl değiştirildiği ile belirlenir.

Tanıma göre öğrenme, aşağıdaki olaylar sonucu meydana gelir:

1. Yapay sinir ağı çevresi tarafından uyarılır.
2. Bu uyarı sonucunda yapay sinir ağının serbest parametreleri, değişim geçirir.
3. Yapay sinir ağı, iç yapısında meydana gelen değişimlerden dolayı, çevreye yeni bir yolla yanıt verir.

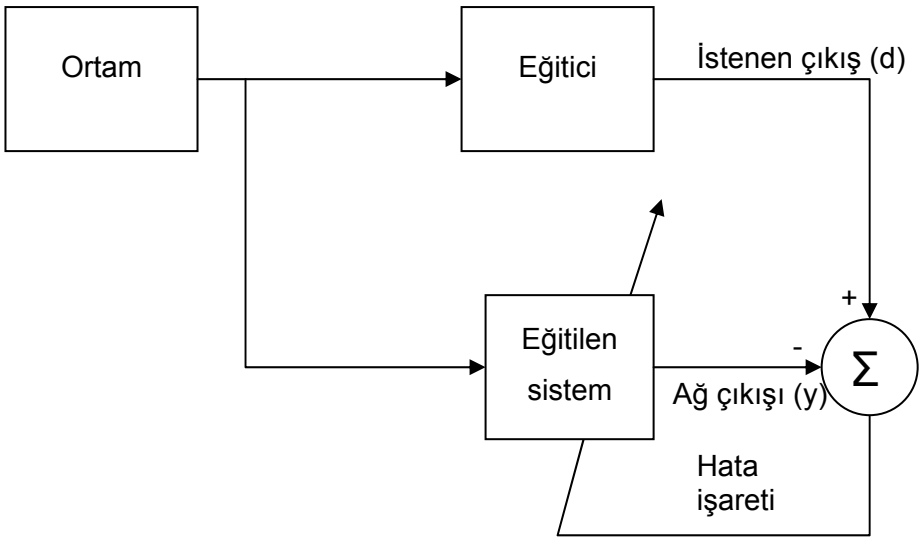
Öğrenme problemine tavsiye edilen kurallar ile çözüm aramak, öğrenme algoritması olarak adlandırılır. Yapay sinir ağlarının tasarımında, kullanılacak tek bir öğrenme algoritması yoktur. Herbirinin diğerine karşı üstün taraflarının olduğu, birbirinden farklı tasarımlar için uygun görülen pek çok öğrenme algoritması vardır. Temel olarak öğrenme algoritmaları, nöronların sinaptik ağırlıklarının ayarlanma şekillerine göre birbirlerinden ayrılırlar. Başka bir etken de, yapay sinir ağının çevresiyle olan ilişkisidir [5-7].

3.2 Öğrenme Şekilleri

Temel olarak öğrenme işlemi ikiye ayrılır:

1. Eğiticili öğrenme
2. Eğiticisiz öğrenme

3.2.1 Eğiticili Öğrenme



Şekil 3-1 Eğiticili öğrenmenin blok diyagramı

Eğiticili öğrenmede, sisteme verilen giriş verilerine karşılık, istenen çıkış değerleri kullanılarak, ağın eğitimi sağlanır. Eğitilen sisteme ait ağırlıklar, eğitim kümesi ve hata aracılığıyla değiştirilir. Hata, ağın çıkışıyla, elde bulunan referans değerinin (istenen çıkış) karşılaştırılmasıyla elde edilir.

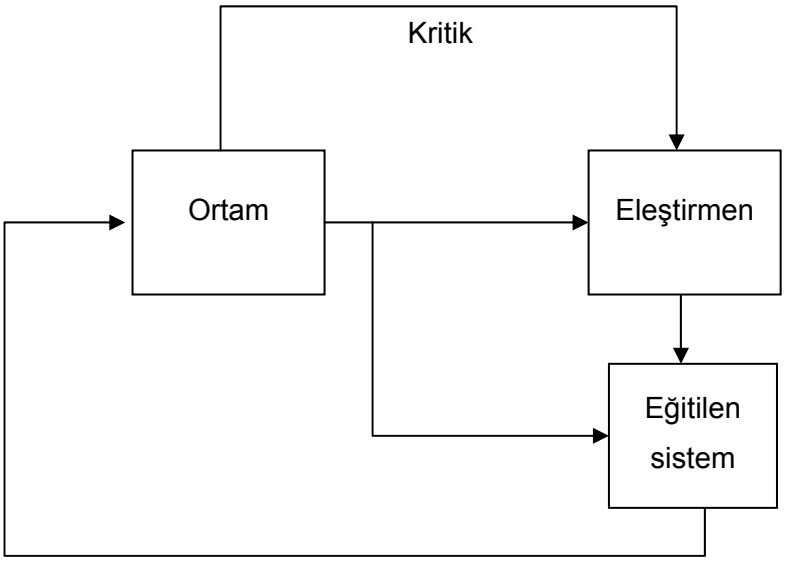
$$e_j(n) = d_j(n) - y_j(n) \tag{3.1}$$

Burada e hata işareti, d istenen çıkış ve y ağın bulduğu çıkış değerleridir. Hataya göre, bağlantı ağırlıkları en uygun çıkışı elde etmek için sonradan düzenlenebilir. Sonuçta ortam hakkında bilgiye sahip olan eğitici, ortam hakkında bilgiye sahip olmayan eğitilen sisteme bilgiyi aktarır (öğrenme).

Widrow-Hoff tarafından geliştirilen delta kuralı ve Rumelhart ve McClelland tarafından geliştirilen genelleştirilmiş delta kuralı veya geriye yayılım (back propagation) algoritması eğiticili öğrenme algoritmalarına örnek olarak verilebilir.

3.2.2 Eğiticisiz Öğrenme

3.2.2.1 Eğiticisiz Öğrenme



Şekil 3-2 Eğiticisiz öğrenmenin blok diyagramı

Eğiticisiz öğrenmede, öğrenme işleminin sonucu önceden bilinmez. Öğrenme işlemi içinde, nöronları birleştiren bağlantıların ağırlıkları, verilen giriş değerlerine bağlı olarak belirli bir aralık içinde düzenlenir. Burada amaç, birbirine benzeyen birimleri, belirli değer aralıkları içinde gruplamaktır.

Öğrenme işleminin her adımında, istenilen yanıtı sağlayacak eğitici yoktur fakat bir beklenti vardır. Bu beklenti, eleştirmen tarafından eğitilen sisteme aktarılarak doğru cevaba ulaşılmaya çalışılır. Eğitilen sistem, elde edilecek yanıta yol açacak her etkiyi kendisi oluşturur ve eleştiriye göre yorumlayarak öğrenme işlemi tamamlanır.

Grossberg tarafından geliştirilen Uyarlamalı Rezonans Teorisi (Adaptive Resonance Theory – ART) veya Kohonen tarafından geliştirilen SOM (Self Organizing Map) öğrenme kuralı eğiticisiz öğrenmeye örnek olarak verilebilir.

3.2.2.2 Özdenetimli Öğrenme



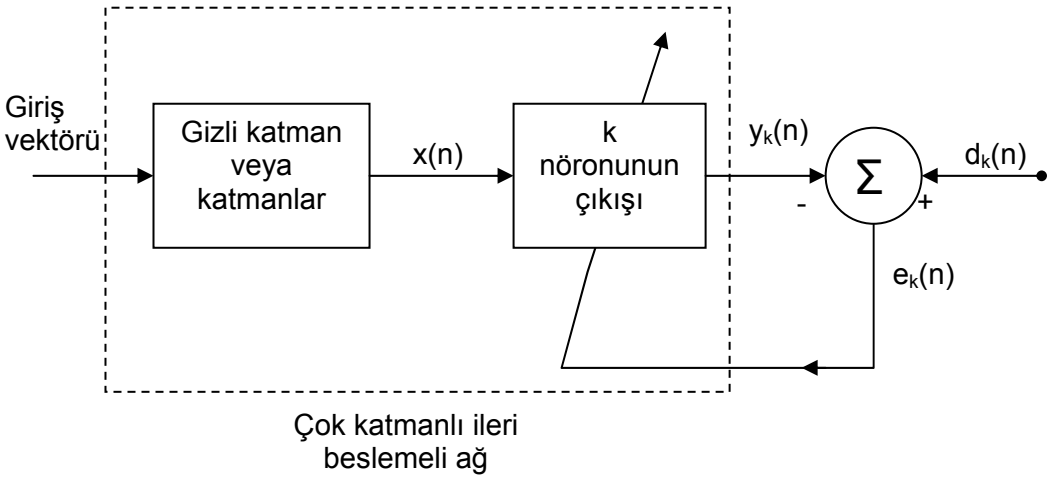
Şekil 3-3 Özdenetimli öğrenmenin blok diyagramı

Özdenetimli öğrenmede, hedeflenen bir çıkış değeri, istenen çıkışı verecek bir eğitici veya eleştirmen yoktur. Burada amaç, birbirlerine benzer birimleri, belirli değer aralıkları içinde gruplamaktır. Bu öğrenmede girişlerin istatistiksel düzeni belirlendikten sonra, sistem sınıflamayı başaracak beceriyi kendi oluşturur [5-6].

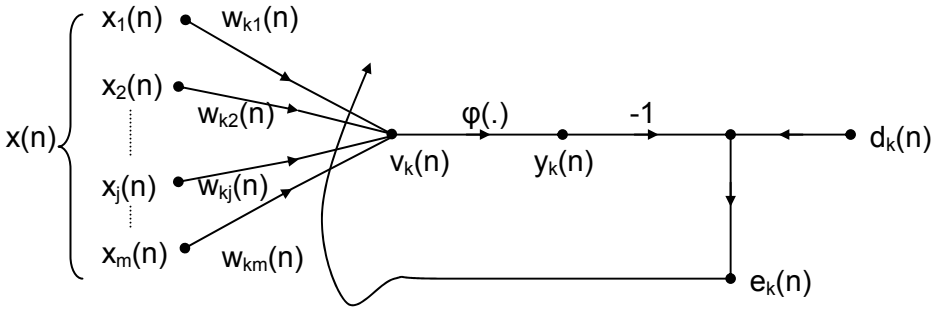
3.3 Öğrenme Kuralları

3.3.1 Hata Düzeltmeli Öğrenme

Bu öğrenme kuralını incelemek için, ileri beslemeli bir yapay sinir ağının k nöronunun çıkış katmanındaki hesaplama düğümü düşünülür (Şekil (3.4)).



(a) Bir yapay sinir ağının blok diyagramı (sadece çıkıştaki nöron göz önüne alınmış)



(b) Çıkış nöronunun işaret akış diyagramı

Şekil 3-4 Hata düzeltmeli öğrenme

k nöronu, bir veya daha çok gizli katman tarafından oluşturulan işaret vektörü, x(n) tarafından uyarılır. n argümanı, ayrık zamanı ya da k nöronunun sinaptik ağırlıklarının ayarlandığı, iteratif işlemin zaman adımını ifade eder. k nöronunun çıkış işareti $y_k(n)$ ile gösterilir. Yapay sinir ağının yalnızca çıkışını ifade eden bu çıkış

işareti, $d_k(n)$ ile gösterilen istenen ya da hedeflenen çıkış ile karşılaştırılır. Bunun sonucu olarak $e_k(n)$ ile gösterilen hata işareti üretilmiş olur.

$$e_k(n) = d_k(n) - y_k(n) \tag{3.2}$$

Hata işareti $e_k(n)$, k nöronunun sinaptik ağırlıklarını düzenleyici ayarlamalar uygulayan bir kontrol mekanizması gerçekler. Düzenleyici ayarlamalar, $y_k(n)$ çıkışını her adımda istenen $d_k(n)$ çıkışına daha yakın hale getirir. Bu amaca ani hatayı minimize ederek ulaşmak mümkündür. Ani hata fonksiyonu:

$$E(n) = \frac{1}{2} e_k^2(n) \tag{3.3}$$

şeklinde hesaplanabilir.

k nöronunun ağırlıklarının adım adım ayarlanması, sistem kararlı hale gelene (ağırlıklar değişmeyene) kadar devam eder. Bu noktada, öğrenme işlemi durdurulur.

Bu bölümde anlatılan öğrenme işlemi hata düzeltmeli öğrenmedir. Ani hata fonksiyonunun minimize edilmesi, delta kuralı veya Widrow-Hoff kuralı adlarıyla bilinen öğrenme kuralında kullanılır. $w_{kj}(n)$, n. Adımda, işaret vektörü x(n)'nin $x_j(n)$ elemanı tarafından uyarılan k nöronunun ağırlığı olsun. Delta kuralına göre, n. adımda w_{kj} ağırlığına uygulanan $\Delta w_{kj}(n)$ değişimi

$$\Delta w_{kj}(n) = \eta\, e_k(n)\, x_j(n) \tag{3.4}$$

şeklinde hesaplanabilir. Burada η öğrenme hızını belirleyen bir pozitif sabittir.

Delta kuralında hata işareti doğrudan ölçülebilir. Bu ölçümün mümkün olması, ağa dış bir kaynaktan k nöronuna istenen çıkışın uygulandığını gösterir.

Sinaptik ağırlığın güncellenmiş değeri şu şekilde bulunur:

$$w(n+1) = w(n) + \Delta w(n) \tag{3.5}$$

Şekil (3.4) (b)'de görüldüğü gibi, hata düzeltmeli öğrenme kapalı döngülü bir geri besleme sistemine örnektir. Bu sistemde, sistemin kararlılığını etkileyen tek parametre vardır, bu da öğrenme hızıdır. Öğrenme hızının seçilmesi, öğrenme

işleminin kesinliğini ve güvenilirliğini de etkiler. Kısacası öğrenme hızı, hata düzeltmeli öğrenmede, öğrenme performansı açısından çok önemli bir rol oynar.

Hata düzeltmeli öğrenme, tek katmanlı ileri beslemeli ağlarda ve çok katmanlı ileri beslemeli ağlarda kullanılır [5].

3.3.2 Bellek Tabanlı Öğrenme

Bellek tabanlı öğrenmede, $\{(xi, di)\}_{i=1}^{N}$ şeklinde ifade edilen giriş – çıkış örneklerini, doğru olarak sınıflar. Burada x_i giriş vektörü, d_i ise bu girişe karşılık gelen istenen yanıttır. Örnek olarak bir ikili sınıflama problemi düşünülsün, problemde C_1 ve C_2 olarak gösterilen iki sınıf vardır. Bu örnekte, istenen çıkış C_1 sınıfı için 0 (ya da 1), C_2 sınıfı için de 1 (ya da 0) değerini alacaktır. Bir test vektörü x_{test}'in (önceden bilinmeyen) sınıflandırılması gerektiğinde, algoritma x_{test}'in yerel komşusunda eğitim verisini elde eder ve çözümler.

Bütün bellek tabanlı öğrenme algoritmalarında iki önemli etken vardır:

1. Ölçüt, test vektörü x_{test}'in yerel komşusunu bulmak için kullanılır.
2. Öğrenme algoritması, x_{test}'in yerel komşusundaki eğitim örneklerine uygulanır.

Algoritmalar, kullanılan etkene göre birbirinden ayrılır.

Basit bir bellek tabanlı öğrenme, en yakın komşu kuralı olarak adlandırılır. Test vektörü x_{test}'in en yakın komşusunda bulunan eğitim örneği, yerel komşu olarak tanımlanır. $x'_N \in \{x_1, x_2, ..., x_N\}$ vektörü için x_{test}'in en yakın komşusu;

$$\min_i d(x_i, x_{test}) = d(x'_N, x_{test}) \quad (3.6)$$

olarak bulunabilir. Burada $d(x_i, x_{test})$, x_i ve x_{test} vektörleri arasındaki öklid uzaklığıdır.

Bellek tabanlı öğrenmelerin önemli bir üyesi, radyal tabanlı fonksiyonlu ağlardır [5].

3.3.3 Hebb Öğrenme Kuralı

Öğrenme kuralları arasında en eski ve en ünlü olanı, Hebb öğrenme kuramıdır. Bu kuram Bölüm (2.3)'te verilmiştir. Burada sadece matematiksel modelden söz edilecektir.

n adımında w_{kj} sinaptik ağırlıklarına uygulanacak olan ağırlık değişimi genel olarak şu şekilde gösterilir:

$$\Delta w_{kj}(n) = F\,(y_k(n), x_j(n)) \tag{3.7}$$

(3.7)'deki eşitlik pek çok şekilde ifade edilebilir. Burada F(.,.) giriş ve çıkış işaretlerinin bir fonksiyonudur [5].

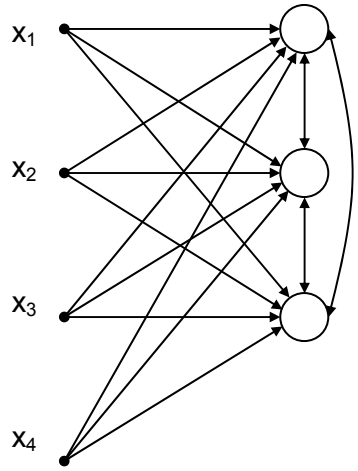
En basit Hebb öğrenme kuralı, aşağıdaki şekilde ifade edilebilir:

$$\Delta w_{kj}(n) = \eta\, y_k(n)\, x_j(n) \tag{3.8}$$

3.3.4 Yarışmalı Öğrenme

Yarışmalı öğrenmede, adından da anlaşılacağı gibi, yapay sinir ağının çıkış nöronları, aktif hale gelebilmek için aralarında yarışırlar. Herhangi bir anda, yalnızca bir çıkış nöronu aktif olabilir. Bu özellik yarışmalı öğrenmenin, bir grup giriş örüntüsünü sınıflamak için kullanılacak istatistiksel özellikleri keşfetmek için uygun olmasını sağlar.

Yarışmalı öğrenmenin en basit şeklinde yapay sinir ağı, girişlere bağlı bir tek çıkış katmanından oluşur. Şekil (3.5)'te görüldüğü gibi, ağ nöronlar arasında geri beslemeye sahip olabilir. Burada açıklanan ağ yapısında geri besleme bağlantıları, her nöronun yanal olarak bağlı olduğu nöronları yavaşlatmasıyla, yanal yavaşlatma meydana getirir.



Şekil 3-5 Basit bir yarışmalı öğrenme ağının mimari yapısı

Kazanan k nöronu için, aktivasyon fonksiyonunun girişi, diğer bütün girişlerden büyük olmalıdır. Kazanan k nöronunun y_k çıkış işareti 1, yarışmayı kaybeden bütün nöronların çıkış işareti ise 0'dır.

$$y_k = \begin{cases} 1 & v_k > v_j \quad \text{bütün j'ler için, } j \neq k \\ 0 & \text{aksi halde} \end{cases} \tag{3.9}$$

j giriş düğümünü, k nöronuna bağlayan sinaptik ağırlık w_{kj} olsun. Giriş düğümleri boyunca dağılan sinaptik ağırlığın sabit değeri, herbir nörona bölüştürülmüştür;

$$\sum_j w_{kj} = 1 \qquad \text{bütün k'lar için} \tag{3.10}$$

Bir nöron, sinaptik ağırlıklarını, aktif ve aktif olmayan giriş düğümlerini değiştirdikten sonra öğrenir. Eğer bir nöron belirli bir giriş örüntüsüne yanıt vermiyorsa, bu nöronda bir öğrenme işlemi gerçekleşmemiş demektir. Eğer belirli bir nöron yarışmayı kazanırsa, bu nöronun her giriş düğümü, sinaptik ağırlığının bir kısmından vazgeçer ve vazgeçilen ağırlık, aktif giriş düğümlerine eşit olarak dağıtılır. Standart yarışmalı öğrenme kuralına göre, ağırlık değişimi şu şekilde tanımlanır:

$$\Delta w_{kj} = \begin{cases} \eta\,(x_j - w_{kj}) & \text{k nöronu kazanan} \\ 0 & \text{k nöronu kaybeden} \end{cases} \tag{3.11}$$

burada η öğrenme hızıdır [5].

3.3.5 Boltzman Öğrenmesi

Ludwig Boltzman'ın anısına Boltzman öğrenme kuralı adı verilen bu kural, istatistiksel mekanik düşüncelerinden türetilen, stokastik bir öğrenme algoritmasıdır. Boltzman öğrenme kuralı kullanılarak tasarlanan yapay sinir ağına, "Boltzman Makinesi" denir.

Boltzman makinesinde, nöronlar yinelemeli bir yapı oluştururlar ve ikili bir sistemde işletilirler. Nöronlar ya +1 ile gösterilen aktif konumda, ya da –1 ile gösterilen pasif konumda olabilir. Makine, bir E enerji fonksiyonu ile tanımlanır.

$$E = -\frac{1}{2}\sum_j \sum_{k \atop j \neq k} w_{kj}\, x_k\, x_j \tag{3.12}$$

$j \neq k$ iken, makinedeki nöronların hiçbirinin, kendi kendine geri beslemesi yoktur. Makine, öğrenme işleminin herhangi bir adımında, rasgele bir nöron seçilerek

(örneğin k nöronu) işletilir. T gibi bir sıcaklıkta, k nöronu x_k'dan – x_k'ya sıçrarken, olasılık

$$P(x_k \rightarrow -x_k) = \frac{1}{1+e^{-\frac{\Delta E_k}{T}}} \tag{3.13}$$

şeklinde hesaplanabilir. Burada ΔE_k, sıçrama sonucunda meydana gelen, makinenin enerji fonksiyonundaki değişimdir. T fiziksel bir sıcaklık değil, yalancı sıcaklıktır (pseudotemperature). Kural tekrarlı olarak uygulanırsa, makine ısıl dengeye ulaşır.

Boltzman makinesindeki nöronlar, görünür ve gizli olmak üzere iki işlevsel gruba ayrılır. Görünür nöronlar, ağ ile işletildiği ortam arasında bir girişim meydana getirir. Gizli nöronlar ise, daima serbest hareket eder.

İki farklı işletim yolu vardır:

1. Bağlama Koşulu: Görünür nöronlar, ortam tarafından belirlenmiş bir konuma bağlanırlar.

2. Serbest Koşma Koşulu: görünür ve gizli tüm nöronlar, serbest şekilde işletilebilir.

j ve k nöronları arasındaki korelasyon, ağ bağlama koşulunda işletilirken ρ_{kj}^{+} ile; serbest koşma koşuluda işletilirken ise ρ_{kj}^{-} ile gösterilir. Makine ısıl dengedeyken, her iki korelesyonun da, olası bütün durumlar için ortalaması alınır. Boltzman öğrenme kuralına göre, j nöronunu k nöronuna bağlayan ağırlıktaki değişim şu şekilde bulunur:

$$\Delta w_{kj} = \eta\,(\rho_{kj}^{+} - \rho_{kj}^{-}), \qquad j \neq k \tag{3.14}$$

Korelasyonlar, +1 ile –1 değerleri arasındadır [5].

3.4 Yapay Sinir Ağlarında Çok Kullanılan Öğrenme Algoritmaları

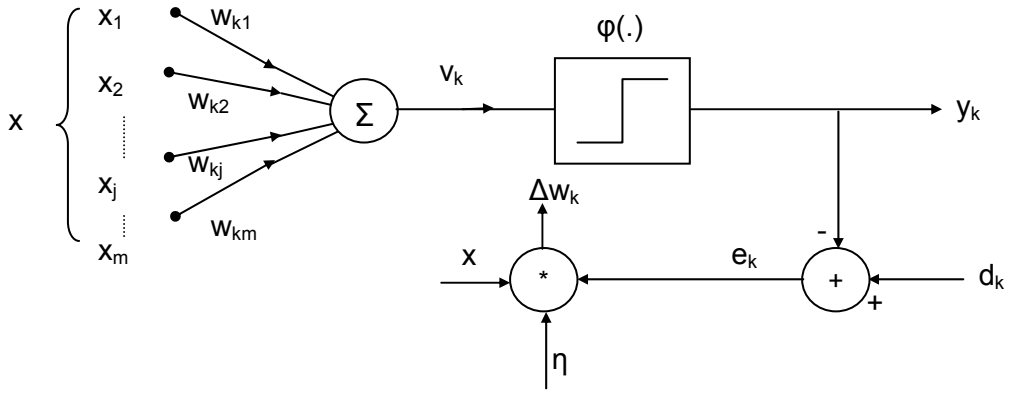
3.4.1 Genlikte Ayrık Algılayıcı (Perseptron) Öğrenme Kuralı

Genlikte ayrık algılayıcı, ilk olarak 1958 yılında F. Rosenblatt tarafından bulunmuştur. Genlikte ayrık algılayıcı ikili veriyi giriş olarak kabul eden ve buna göre sonuç üreten, bir yapay sinir ağıdır. Bu ağ, bir giriş ve bir çıkış katmanından oluşan,

basit bir yapıdadır. Sadece doğrusal ayrıştırılabilir olan fonksiyonları çözebilir. VE (AND), VEYA (OR) gibi temel mantık işlemcilerini çözebilirken, XOR gibi daha karmaşık mantıksal işlemleri çözemez. Şekil (3.6)'da, genlikte ayrık algılayıcı (perseptron) yapısı görülmektedir.

Genlikte ayrık algılayıcı, birinci katmanında sabit ağırlıklar ve sabit fonksiyonlar; ikinci katmanında bağlantı ağırlıkları eğitim kümesiyle belirlenebilen tek bir nörondan oluşur. Girişlere doğrudan bağlı tek bir nöron değildir. Birinci katman değişmeyen bir yapıya sahipken, ikinci katman eğitilebilir bir yapıdadır.

Şekil (3.6)'da, Δw ağırlıktaki değişim, v_k aktivasyon fonksiyonunun girişi, η öğrenme hızı, x ağın girişi, y_k ağın çıkışı, d_k istenen çıkış ve e_k hata fonksiyonudur.



Şekil 3-6 Genlikte Ayrık Algılayıcı (Perseptron) yapısı

$$y_k(n) = \begin{cases} 1 & v(n) > \Theta \\ 0 & v(n) < \Theta \end{cases} \tag{3.15}$$

burada Θ eşik değeri, vk ise aktivasyon fonksiyonunun girişidir.

$$v_k = \sum_{j=1}^{m} x_k(n) w_k(n) \tag{3.16}$$

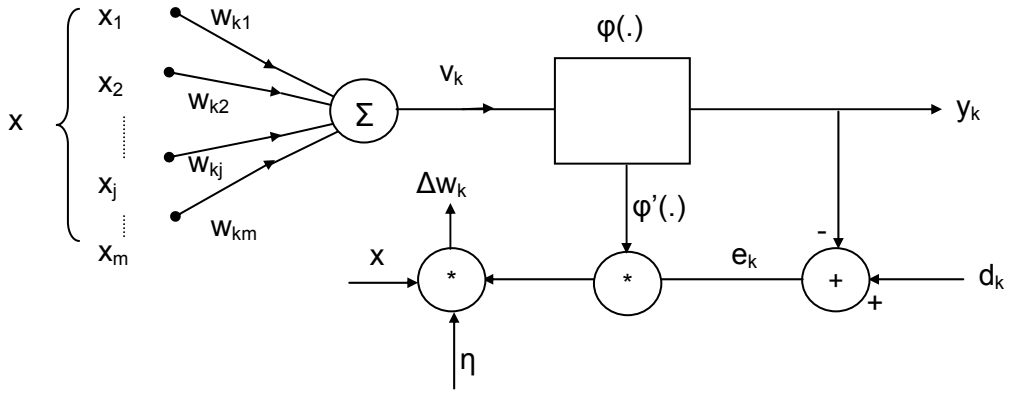
Öğrenme aşağıdaki kurala göre gerçekleşir:

$$\Delta w_k(n) = \eta \left[d_k(n) - y_k(n) \right] \tilde{X} \tag{3.17}$$

Eşitlik (3.17)'de $\tilde{X}$, giriş vektörlerinin tamamını ifade etmektedir. Bu kurala göre, ağırlıklar ancak çıkış istenen çıkıştan farklı ise değişir. Bütün girişler için, istenen çıkış elde edilince, eğitim durdurulur [5-10].

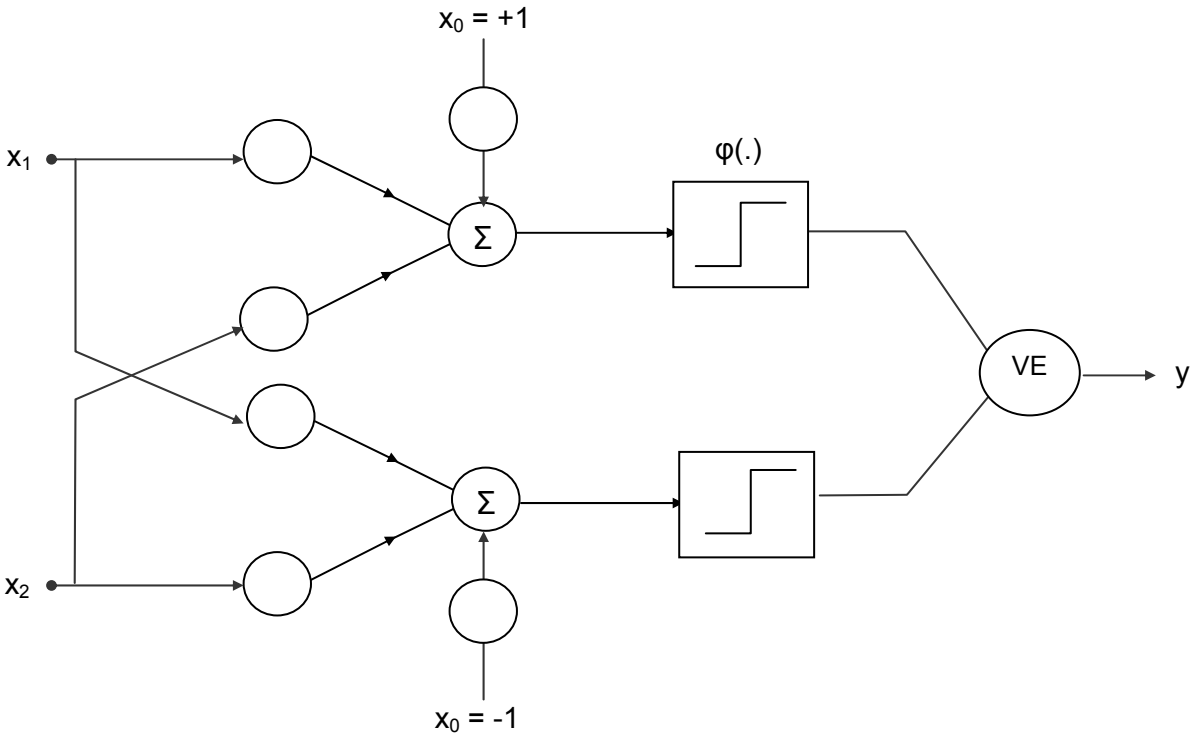
3.4.2 Genlikte Sürekli Algılayıcı (ADALINE) Öğrenme Kuralı

Genlikte sürekli algılayıcının, genlikte ayrık algılayıcıdan farkı eğitimidir. Genlikte sürekli algılayıcı, giriş işareti olarak [1 , 0] ikili değerleri alır ve bias terimini işleme katmaz. Şekil (3.7)'de, genlikte sürekli algılayıcının yapısı gösterilmiştir. Genlikte sürekli algılayıcı eğitimde, delta öğrenme kuralını kullanır.



Şekil 3-7 Genlikte sürekli algılayıcı (ADALINE) yapısı

3.4.3 Katmanlı Genlikte Sürekli Algılayıcı – MADALINE (Multiple – ADALINE) Öğrenme Kuralı



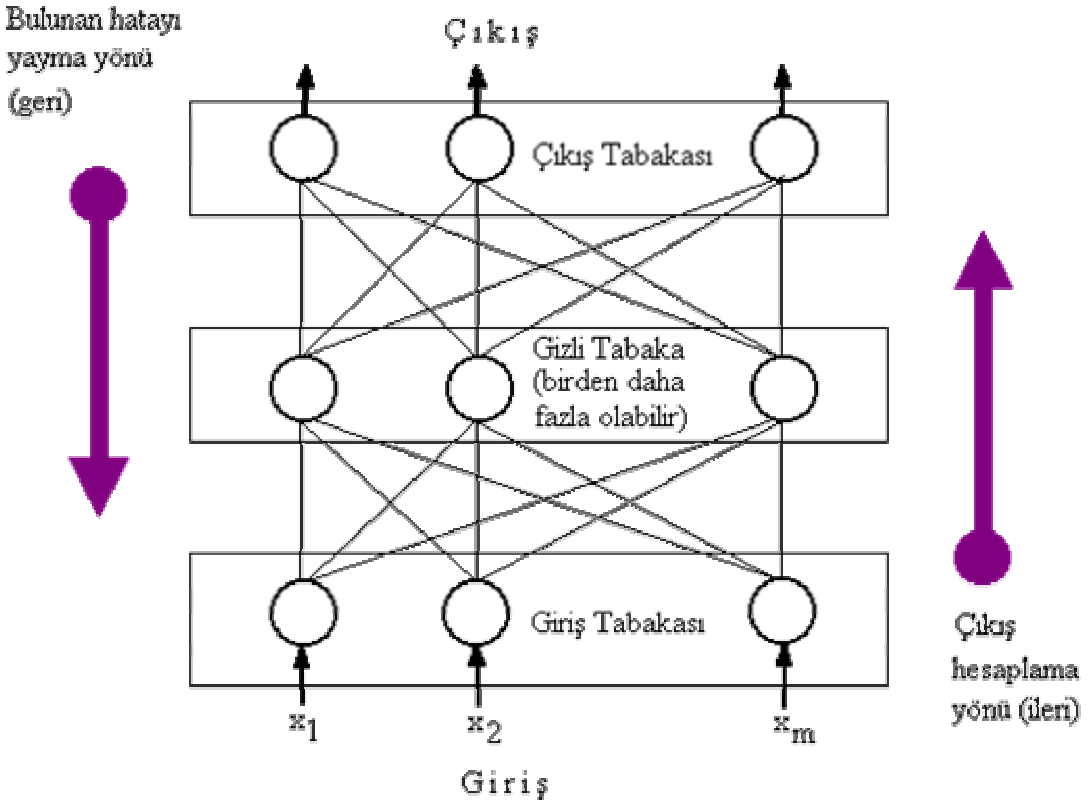
Şekil 3-8 ADALINE girişi olan MADALINE yapısı

Widrow ve Hoff tarafından geliştirilen MADALINE yapısı, birden fazla adaptif eleman içeren yapay sinir ağı modellerindendir. Ridgway, Hoff ve Glanz MADALINE

yapısının matematiksel analizini beraber geliştirmişlerdir. Bu yapıda, giriş katmanına adaptif ADALINE birimleri verilerek, bu birimler mantıksal işlemci üzerinden geçirilir. Mantıksal işlemci, yapay sinir ağının çıkış değerini üretir. MADALINE yapısı Şekil (3.8)'de gösterilmiştir.

3.4.4 Çok Katmanlı Perseptronlarda Öğrenme Algoritmaları

Bir çok katmanlı perseptron modeli, şekil (3.9)'de gösterilmiştir. Bu ağ modeli özellikle mühendislik uygulamalarında en çok kullanılan yapay sinir ağıdır. Bir çok öğrenme algoritmasının bu ağı eğitimde kullanılabilir olması, bu modelin yaygın kullanılmasının sebebidir.



Şekil 3-9 Çok katmanlı yapay sinir ağı

Bir çok katmanlı perseptron, bir giriş, bir veya daha fazla gizli katman ve bir de çıkış katmanından oluşur. Bir katmandaki bütün işlem elemanları bir üst katmandaki bütün işlem elemanlarına bağlıdır. Bilgi akışı ileri doğru olup, geri besleme yoktur. Bunun için ağ, ileri beslemeli yapay sinir ağı olarak adlandırılır. Giriş katmanında herhangi bir bilgi işleme yapılmaz. Buradaki işlem elemanı sayısı tamamen uygulanan problemler giriş sayısına bağlıdır. Gizli katman sayısı ve gizli katmanlardaki nöron sayısını bulmak için çeşitli yöntemler geliştirilmişse de, deneme

yanılma ile bulmak çok daha uygun sonuçlar alınmasını sağlar. Çıkış katmanındaki eleman sayısı ise yine uygulanan probleme dayanılarak belirlenir [5, 10, 14, 15].

3.4.4.1 Geriye Yayılım (Back propagation) Algoritması

Geriye yayılım algoritması, 1972 yılında, Harvard'daki bir grup araştırmacı tarafından ortaya atılmıştır. İteratif bir yöntem olan algoritma, çıkışları belli bir değere üstel bir yaklaşıklıkla getirmektir. Bunun için, ortalama karesel hata minimize edilmeye çalışılır.

Geriye yayılım öğrenmesi sırasında ağ, her giriş örüntüsünü, çıkış nöronlarında sonuç üretmek üzere gizli katmanlardaki nöronlardan geçirir. Daha sonra çıkış katmanındaki hataları bulabilmek için, beklenen sonuçla, elde edilen sonuç karşılaştırılır. Bundan sonra, çıkış hatalarının türevi çıkış katmanından geriye doğru gizli katmanlara geçirilir. Hata değerleri bulunduktan sonra, nöronlar kendi hatalarını azaltmak için ağırlıklarını ayarlar. Ağırlık değiştirme denklemleri, ağdaki ortalama hata karesini en küçük yapacak şekilde düzenlenir. Öğrenme algoritması aşağıda notasyonuyla birlikte açıklanmıştır.

n. iterasyonda, j. nöronun çıkışındaki hata işareti, j. nöron çıkış katmanındaysa;

$$e_j(n) = d_j(n) - y_j(n) \tag{3.14}$$

şeklinde tanımlanır. Ağın öğrenme sırasındaki hatası geriye yayılım algoritmasıyla en küçük değere indirilir. Hata fonksiyonu, N eğitim kümesindeki tüm örüntü sayısı olmak üzere, (3.15) denklemiyle bulunur:

$$E(n) = E_{ort}(n) = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{2}\sum_{j\in C} e_j^2(n) \tag{3.15}$$

(3.15) denklemindeki C kümesi, ağın çıkış katmanındaki bütün nöronları içerir. Hatayı en küçük değere indirmek için en uygun ağırlık seçimi şu şekilde gerçekleştirilir:

1. $\delta E/\delta w$ hesaplanır.

2. Ağırlıklar w(n)'den w(n+1)'e güncellenir:

$$w(n+1) = w(n) + \Delta w(n) \tag{3.16}$$

$$\Delta w_{ji}(n+1) = \eta\, \delta_j(n)\, y_i(n) \tag{3.17}$$

(3.17) denklemi ile yazılan kurala delta kuralı denilmektedir. Ağ ağırlıklarını güncellemek için kullanılan bu eşitlikte, η öğrenme hızıdır. Yerel gradyen denilen δ_{ji}, j. nöronun çıkış katmanında veya gizli katmanda olmasına göre farklı hesaplanır. φ aktivasyon fonksiyonunu belirtmek üzere, eğer j. nöron çıkış katmanında ise;

$$\delta_{ji} = e_j(n)\varphi_j'(v_j(n)) \tag{3.18}$$

bağıntısından, eğer j. nöron gizli katmanda ise;

$$\delta_{ji} = \varphi_j'(v_j(n))\sum_k \delta_k(n) w_{kj}(n) \tag{3.19}$$

bağıntısından hesaplanır.

Geriye yayılım algoritması, uygulanması kolay bir yöntemdir ve yapay sinir ağlarında en çok kullanılan algoritmalardan biridir. Algoritmanın pekçok türevi değişik tür sistemlere başarıyla uygulanabilmektedir. Momentum terimli geriye yayılım, adaptif geriye yayılım, adaptif kazançlı geriye yayılım, esnek yayılım, Gauss – Newton, Levenberg – Marquardt, birleşik gradyen azalması gibi öğrenme algoritmaları, geriye yayılım algoritmasının türevlerinden bazılarıdır [5-8].

Bu algoritma, Bölüm 4'te ayrıntılı olarak incelenecektir.

3.4.4.2 Delta – Bar – Delta Algoritması

Delta-Bar-Delta algoritması, çok katmanlı perseptronlarda yerel optimizasyon yöntemine dayalı adaptif öğrenme hızını, ağdaki yakınsamayı arttırmak için kullanan, sezgisel bir yaklaşımdır. Deneysel çalışmalar, ağırlık uzayının her boyutunun tüm hata yüzeyi açısından tamamen farklı olabileceğini göstermiştir. Hata yüzeyindeki değişimleri açıklamak için, özellikle ağın her bağlantısı kendi öğrenme katsayısına sahip olmalıdır. Bu düşünce, tek ağırlık boyutu için uygun adım büyüklüğü, tüm ağırlık boyutları için uygun olmayabilir. Bununla birlikte, her bir bağlantıya bir öğrenme oranı atanarak ve bu öğrenme oranının zamanla

değişmesine izin verilirse, yakınsama zamanını azaltmak için daha çok serbestlik derecesi sağlanmış olur.

İleri beslemeli yapay sinir ağı yapıları, çoğu zaman karmaşıktır. Ağdaki her bağlantı için en uygun öğrenme katsayıları kümesini belirlemek oldukça zaman alıcı olabilir. Gradyenin geçmişteki değerlerini kullanarak, yerel hata yüzeyinin eğriliğini çıkarmak için sezgisellik uygulanabilir. Bir bağlantı için ağırlık değişimlerinin işareti, birkaç ardışık zaman adımları sırayla değiştiği zaman, bağlantı için öğrenme oranı azalmalıdır. Bağlantı ağırlık boyutu hata yüzeyi ile ilgili büyük bir eğriliğe sahiptir. Bağlantı ağırlık değişimleri bir kaç ardışık zaman adımları için aynı işarete sahip olduğundan bağlantı için öğrenme oranı arttırılmalıdır. Ağırlık boyutu hata yüzeyi ile ilgili küçük bir eğriliğe sahiptir ve yüzey önemli bir mesafe için aynı doğrultudaki eğime göre devam eder.

Delta – Bar – Delta algoritmasında, gradyen azalması doğrultusu ile ağırlık uzay yönünün çakışması gerekmez. Ardışıl iterasyonlar arasında, ağırlık düzeltmesi zıt doğrultularda ise adım boyu küçültülür, aksi halde adım boyutu büyütülür. Bu öğrenme algoritmasında, her bağlantı için değişken öğrenme oranı $\eta(k)$ atanır ve ağırlık geriye yayılım algoritmasına benzer şekilde güncelleştirilir.

$$w(n+1) = w(n) + \Delta w(n) \tag{3.20}$$

$$\Delta w_{ji}(n+1) = \eta(n)\, \delta_{ij}(n)\, y_i(n) \tag{3.21}$$

Delta – Bar – Delta algoritmasında, öğrenme hızının bulunması için, gradyenin ortalaması, $\bar{\delta}_{ij}(n)$ kullanılır.

$$\bar{\delta}_{ij}(n) = (1-\varepsilon)\delta_{ij}(n-1) + \varepsilon\bar{\delta}_{ij}(n-1) \tag{3.22}$$

Denklem (3.22)'de görülen ε konveks ağırlık faktörü, pozitif sabit bir değerdir. Algoritmanın öğrenme hızının güncelleştirilmesi denklem (3.23) ve (3.24)'te gösterilmiştir:

$$\Delta\eta_{ij}(n) = \begin{cases} \kappa & \bar{\delta}_{ij}(n-1)\delta_{ij}(n) > 0 \\ -\beta\,\eta_{ij}(n) & \bar{\delta}_{ij}(n-1)\delta_{ij}(n) < 0 \\ 0 & \text{aksi halde} \end{cases} \tag{3.23}$$

$$\eta_{ij}(n+1) = \eta_{ij}(n) + \Delta\eta_{ij}(n) \tag{3.24}$$

(3.23) denkleminde κ öğrenme katsayısı artma faktörü, β öğrenme katsayısı azaltma faktörü, $\delta_{ij}(n)$ herhangi bir andaki gradyendir. Denklem (3.25)'teki gibi hesaplanır:

$$\delta_{ijj}(n) = \frac{\partial E(n)}{\partial w_{ij}(n-1)} \tag{3.25}$$

Bu değerler, ağ tasarımcısı tarafından belirlenir. (3.23)'ten görülebileceği gibi algoritma, öğrenme katsayılarını doğrusal olarak artırmakta, fakat geometrik olarak azaltmaktadır.

Jacobs (1988), Delta – Bar – Delta algoritmasına momentum terimi ekleyerek performansı arttırmıştır.

Delta – Bar – Delta algoritması, küçük parametre değişimlerine karşı çok duyarlıdır. Özellikle κ öğrenme katsayısı artma faktörünün büyük seçilmesi ağda sapmalara yol açarken, küçük seçilmesi de öğrenme sürecinin uzamasına neden olmaktadır [7,10].

3.4.4.3 Bağlaşımsız Yakınsama (Decoupled Momentum) Algoritması

Bağlaşımsız yakınsama algoritmasında her w_{ij} ağırlık değerinin kendi adaptif öğrenme hızı η_{ij}, ve momentum katsayısı α_{ij} vardır (Minai ve Williams, 1992). Öğrenme hızı, hata yüzeyine göre değişkenlik gösterir. Düzlem bölgelerinde hızla artarken, eğimli alanlarda yavaşlar. Momentum katsayısı da, öğrenme hızı gibi değişkendir. Platolarda artarken, minimum yakınlarında üstel olarak azalır. Bağlaşımsız yakınsama algoritmasının öğrenme hızı ve momentum katsayısı aşağıda gösterilen şekilde hesaplanabilir.

$$\Delta\eta_{ij}(n) = \begin{cases} \kappa_1 e^{-\gamma_1 \bar{\varphi}_{ij}(n)} & \bar{\delta}_{ij}(n-1)\delta_{ij}(n) > 0 \\ -\beta_1 \eta_{ij}(n) & \bar{\delta}_{ij}(n-1)\delta_{ij}(n) < 0 \\ 0 & \text{aksi halde} \end{cases} \tag{3.26}$$

$$\eta_{ij}(n+1) = \eta_{ij}(n) + \Delta\eta_{ij}(n) \tag{3.27}$$

$$\Delta\alpha_{ij}(n) = \begin{cases} \kappa_m e^{-\gamma_m \bar{\varphi}_{ij}(n)} & \bar{\delta}_{ij}(n-1)\delta_{ij}(n) > 0 \\ -\beta_m \eta_{ij}(n) & \bar{\delta}_{ij}(n-1)\delta_{ij}(n) < 0 \\ 0 & \text{aksi halde} \end{cases} \quad (3.28)$$

$$\alpha_{ijj}(n) = \min[\alpha_{max}, \alpha_{ij}(n-1) + \Delta\alpha_{ij}(n-1)] \quad (3.29)$$

Yukarıda geçen ifadelerde kullanılan, gradyenin ortalaması, $\bar{\delta}_{ij}(n)$ ve gradyen, $\delta_{ij}(n)$ denklem (3.22) ve denklem (3.25)'te açıklandığı gibi hesaplanır. Öğrenme hızındaki artış doğrusal ve azalış üsteldir.

Bağlaşımsız yakınsama algoritmasında, öğrenme hızı ve momentum katsayısının hesaplanmasında kullanılan parametreler, ağ tasarımcısı tarafından belirlenir.

Minia ve Williams 1992'de Delta – Bar – Delta algoritması ile bağlaşımsız yakınsama algoritmasını birleştirerek yeni bir algoritma geliştirmişlerdir. Bu yeni algoritma, momentum katsayısının hesabı için denklem (3.28) ve denklem (3.29)'u, öğrenme hızının değişimi için de denklem (3.23) ve denklem (3.24)'ü kullanır.

Minia ve Williams (1992), bağlaşımsız yakınsama algoritması, yakınsama hızı açısından geriye yayılım ve Delta – Bar – Delta algoritmalarından daha iyi olduklarını savunur. Ancak, tasarımcının sisteme tanıtması gereken parametrelerin çok olması ve algoritmanın uç bölgelerde aşırı duyarlık göstermesi yüzünden, algoritma uygulanabilirlik açısından sınırlıdır [7].

3.4.4.4 Hızlı Yayılım (Quick - Propagation) Algoritması

Hızlı yayılım (Quickprop) algoritması, Scott Fahlman tarafından Carbegie Mellon Üniversitesi'nde geliştirilen ve Newton yöntemine dayanan, çok katmanlı perseptronların eğitilmesi için kullanılan bir sezgisel öğretme algoritmasıdır.

Sezgisel algoritmalar, deneyime ve eğitilmiş tahmine dayalı kurallar içermektedir. Bu algoritmalar optimum çözümü garanti edemeyip, sadece ona yakın çözümü bulmayı garanti etmektedir.

Sezgisel yöntemleri kullanmanın sağladığı bazı avantajlar şöyle özetlenebilir:

1. Sezgisel yöntemler karar verici mekanizma için sadeleştirici olabilir.

2. Sezgisel yöntem, herhangi bir tam yöntemin parçası olarak öğrenme amacıyla kullanılabilir.

3. Gerçek dünya problemleri için her zaman matematiksel formülasyon kurmak çok kolay olmayabilir. Bu basitleştirme sonucu oluşan hata, bir sezgisel yöntemin sağladığı optimale yakın çözümün sahip olduğu hatadan daha büyük olabilir.

Problem uzayında mümkün olan en kısa sürede, bir çözüm bulmak için ağırlık uzayında gradyeni son derece küçük azaltmak gerekebilir. Bunun yerine çözümden fazla uzaklaşmaksızın mümkün olan en büyük adımlarla kısa sürede çözüme ulaşmak arzu edilir. Hata fonksiyonunun gradyeni biliniyorsa, güvenli bir şekilde ağırlık uzayında daha büyük adımla çözüme ulaşılabilir. Bu algoritmada iki geleneksel yaklaşım birleştirilerek çözülmüştür. Bunlar;

1. Hesaplamanın geçmişteki durumu hakkında bazı sezgilere dayanan (genel veya her bir ağırlık için ayrı) dinamik olarak ağırlıkların ayarlanması.

2. Her bir ağırlığa göre hatanın ikinci türevinin belirgin kullanımı.

Hızlı yayılım algoritması bir ikinci dereceli yöntemden daha sezgiseldir ve çok az da olsa Newton-yöntemine dayanan bu yöntemde iki kabul kullanılır. Her bir ağırlık için, ağırlık hata eğrisi kolları yukarı doğru açık olan bir parabol ile yaklaştırılabilir. Hata eğrisinin eğilimindeki değişim, diğer tüm ağırlıkların aynı andaki değişiminden etkilenmez.

Yukarıda sözü edilen parabol, birbirinden bağımsız her bir ağırlık için şimdiki ve önceki hata gradyeni ve bu gradyenlerin ölçüldüğü noktalar arasındaki değişim kullanılarak belirlenir. Daha sonra, algoritma doğrudan bu parabolün minimum noktasına atlar.

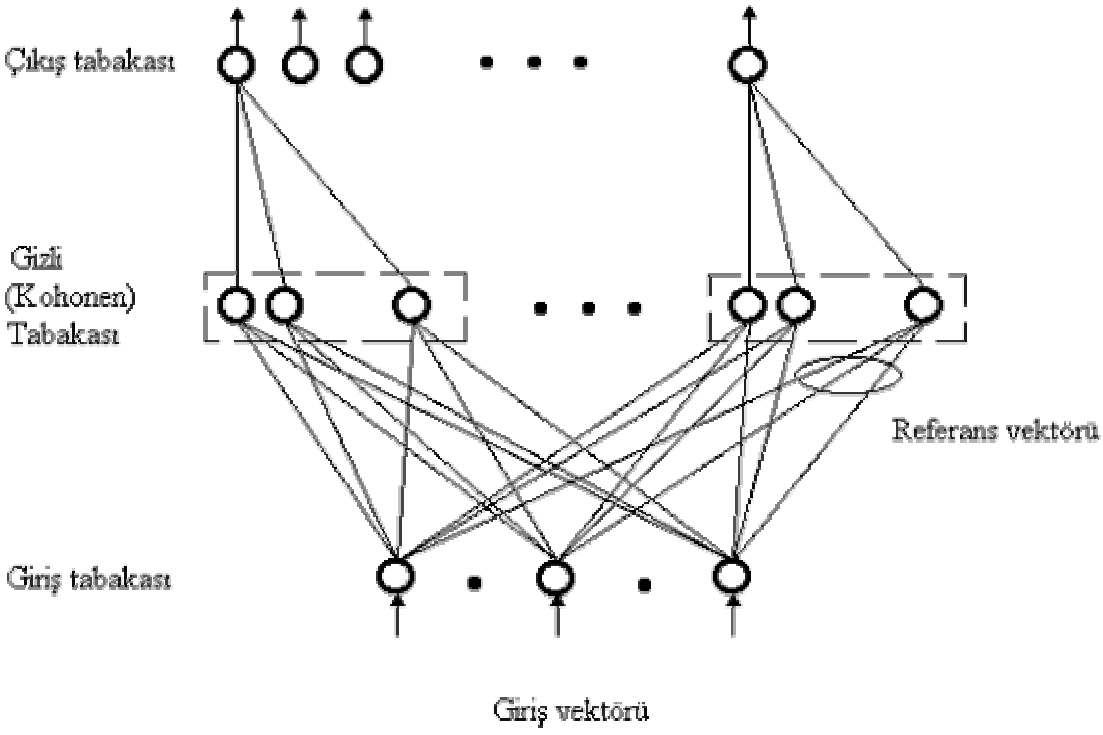
Genellikle hızlı yayılım algoritmasının performans testleri diğer yöntemlerle karşılaştırıldığında oldukça iyidir. Bu algoritma, özellikle gürültü seviyesi az olan dataya sahip problemlerde, genişletilmiş Delta – Bar – Delta kadar iyi sonuç vermektedir. Hızlı yayılım algoritması gürültü seviyesi fazla olan datalı problemlerde

çok iyi çalışır ancak gürültülü dataların bulunduğu problemlerde genişletilmiş Delta – Bar – Delta kullanılmalıdır [7-8, 10].

3.5 Diğer Yapay Sinir Ağları Ve Öğrenme Algoritmaları

3.5.1 Vektör Kuantalama (Learning Vector Quantization)

Şekil (3.10), üç katmandan oluşan vektör kuantalama ağını göstermektedir. Bu katmanlar, giriş katmanı, gizli katman ve çıkış katmanıdır. Vektör kuantalama ağı, giriş katmanı ve gizli katman arasında tamamen, gizli katman ve çıkış katmanı arasında da kısmen bağlıdır. Her çıkış işlemci elemanı, farklı bir gizli işlemci eleman kümesine bağlıdır. Gizli ve çıkış işlemci elemanları arasındaki ağırlıklar 1'e sabitlenmiştir. Giriş – gizli işlemci eleman bağlantılarının ağırlıkları "referans" vektörlerinin elemanlarını oluşturur (her gizli işlemci elemana bir referans vektör atanmıştır). Ağın öğretilmesi sırasında bunlar yeniden değerler alırlar. Hem gizli işlemci elemanlar (Kohonen işlemci elemanları) hem de çıkış işlemci elemanları ikili (binary) çıkışa sahiptir. Ağa bir giriş deseni verildiğinde referans vektörü giriş desenine en yakın olan gizli işlemci eleman kümesi "1", diğerleri "0" üretir. "1" üreten çıkış işlemci elemanı giriş işaretini sınıflar ve her işlemci eleman ayrı bir sınıfa atanmıştır [5].



Şekil 3-10 Vektör kuantalama ağı

En basit vektör kuantalama öğrenme algoritması aşağıdaki adımlarla açıklanabilir.

1. Referans vektörlerinin ağırlıklarının başlangıç değerlerini belirle,
2. Ağa giriş deseni uygula,
3. Giriş deseni ile her referans vektörü arasındaki “Öklit (Euclidian) uzaklığı”nı hesapla,
4. Giriş desenine en yakın referans vektörünün (-ki bu vektör kazanan gizli işlemci elemanının referans vektörüdür) ağılıklarını yeniden düzenle. Eğer bu gizli işlemci eleman, o çıkış işlemci elemanın bağlı olduğu gizli işlemci eleman kümesine bağlı ise referans vektörünü giriş desenine daha yakın hale getir. Aksi takdirde referans vektörünü uzaklaştır,
5. (2) numaralı adıma yeni bir giriş deseni ile dön ve işlemlere bütün giriş desenleri sınıflandırılıncaya kadar devam et.

3.5.2 Hopfield Ağı

Şekil (3.11)’de Hopfield ağının bir şekli gösterilmiştir. Bu ağ genellikle ikili (0 veya 1) ve iki kutuplu (+1 veya -1) girişler kabul eder [5]. Tek katman işlemci elemanları vardır ve her işlemci eleman bir diğerine bağlanmıştır. Bu yapı, geri beslemelidir. Hopfield ağının eğitilmesi sadece bir adım alır ve w_{ij} ağırlıkları aşağıdaki gibi hesaplanır:

$$w_{ij} = \begin{cases} \frac{1}{N}\sum_{c=1}^{p} x_i^c x_j^c, & i \neq j \\ 0, & i = j \end{cases} \qquad (3.30)$$

Burada w_{ij}, i işlemci elemanından j işlemci elemanına olan bağlantının ağırlığını, x_i^c c sınıfı için eğitme giriş deseninin i. elemanını, P sınıf sayısını ve N de işlemci eleman sayısını gösterir. Denklem (3.44)’de $w_{ij}=w_{ji}$ ve $w_{ii}=0$ durumları ağın kararlılığını garantiler. Bilinmeyen bir bilgi ağa girildiğinde, ağın çıkışları bilinmeyen bir desen elemanlarına eşitlenir;

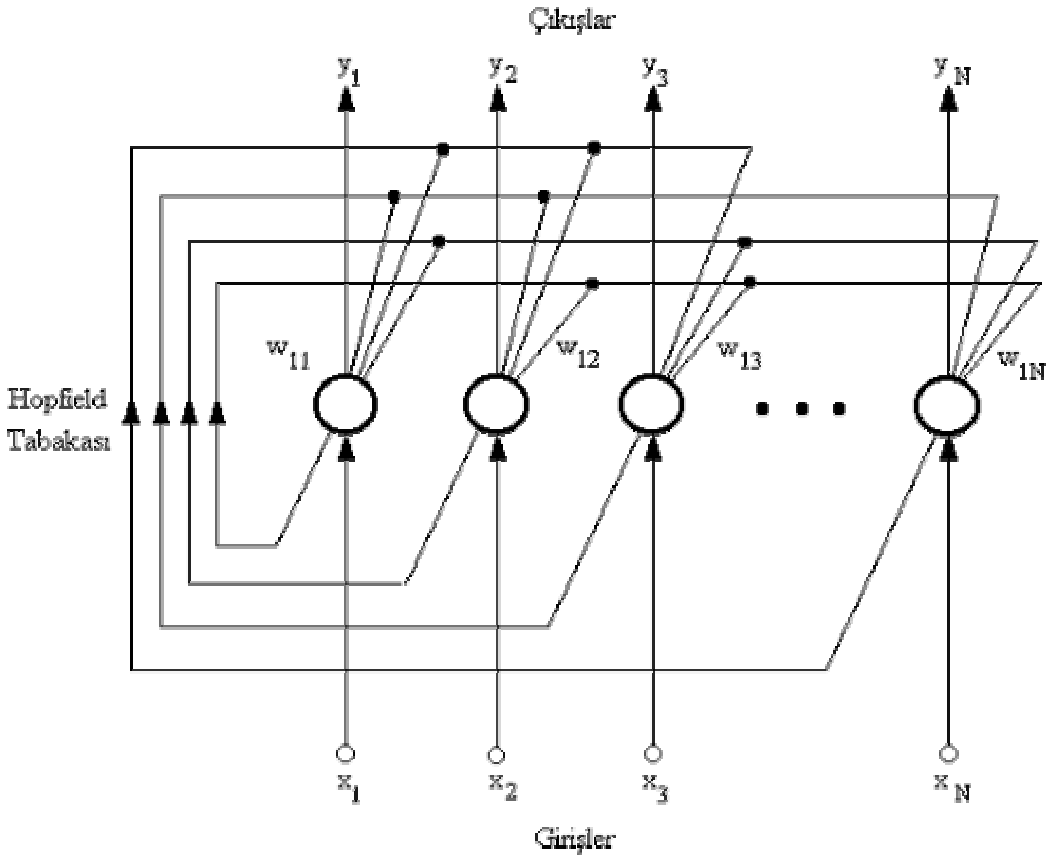
$$y_i(0) = x_i \qquad 1 \leq i \leq N \qquad (3.31)$$

Bu başlangıç değerleri ile başlayarak Hopfield ağı bir denklem (3.32)'yi kullanarak minimum enerji durumuna geçmek için döngüye girer.

$$y_i(n+1) = f[\sum_{j=1}^{N} w_{ij} y_i(n)] \qquad 1 \le i \le N \tag{3.32}$$

burada f, denklem (3.33)'de görüldüğü gibi tanımlanan keskin sınırlama (hard limiting) fonksiyonudur.

$$f(x) = \begin{cases} -1 & x < 0 \\ 1 & x > 0 \end{cases} \tag{3.33}$$

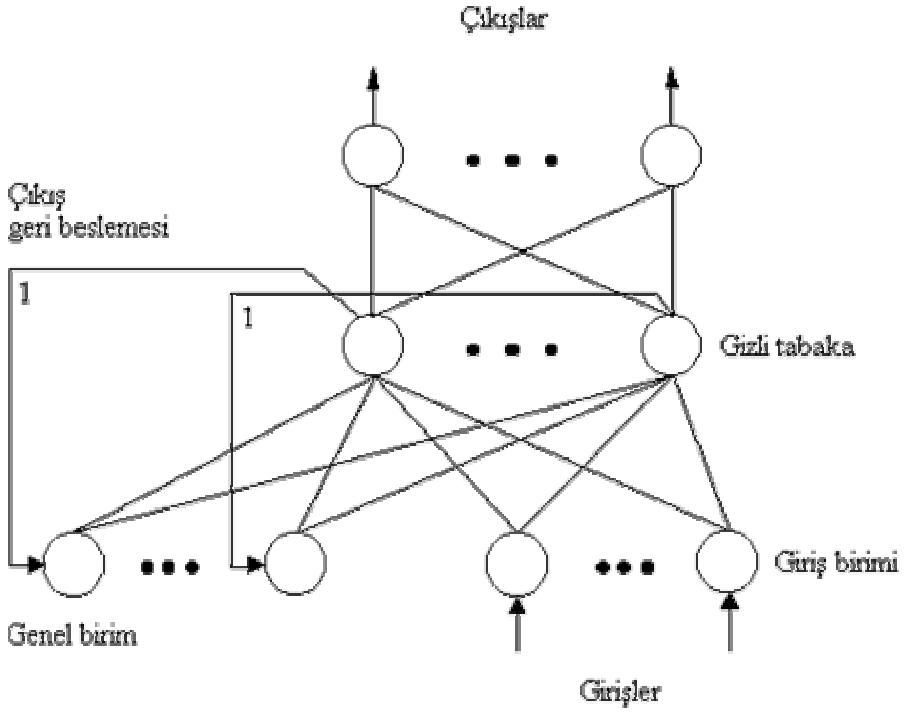


Şekil 3-11Hopfield ağı

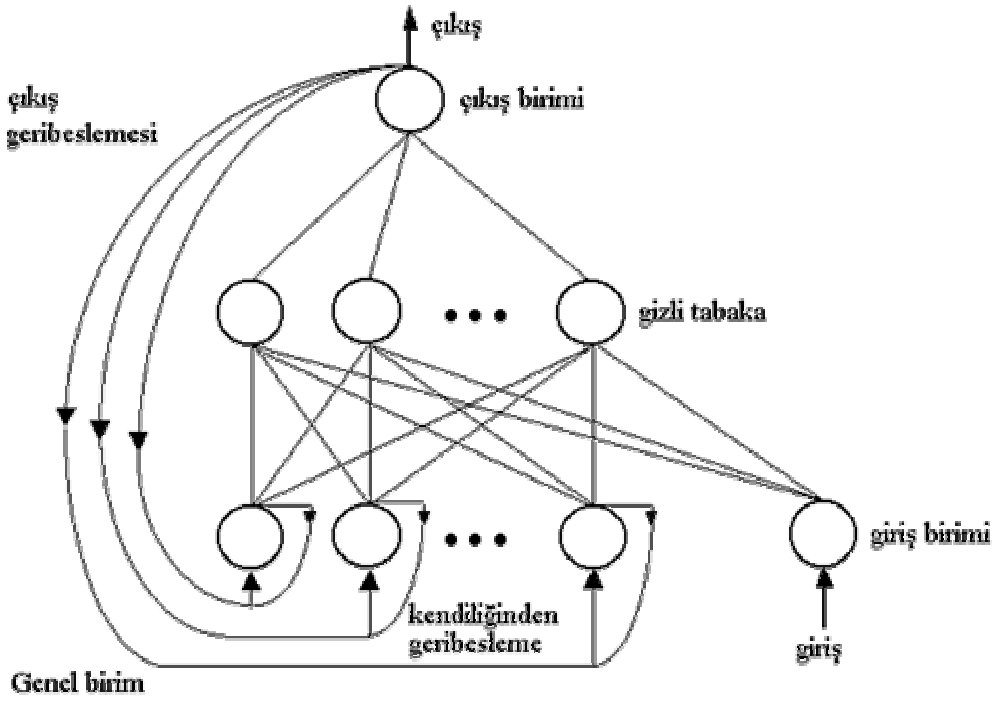
3.5.3 Elman ve Jordan Ağları

Şekil (3.12) a ve b Elman ve Jordan ağlarını göstermektedir. Bu ağlar, çok katmanlı perseptronlara benzer bir yapıdadır. Her iki ağda da gizli katmana ek olarak diğer bir

"durum katmanı" denilen özel bir gizli katman daha vardır. Bu katman gizli katmandan veya çıkış katmanından geri besleme işaretleri alır [9].



(a)



(b)

Şekil 3-12 a) Elman ağı, b) Jordan ağı

Jordan ağının aynı zamanda durum katmanındaki her işlemci elemandan kendisine bağlantıları vardır. Her iki ağda da durum katmanındaki işlemci elemanların çıkışları ileriye doğru gizli katmana verilmektedir. Eğer sadece ileri doğru bağlantılar göz önüne alınır ve geri besleme bağlantılarına sabit değerler verilirse, bu ağlar sıradan ileri beslemeli ağlar haline gelirler.

3.5.4 Kohonen Ağı

Kohonen ağı, bir giriş katmanı ve bir de çıkış katmanı olmak üzere iki katmandan oluşur. Bu ağ Şekil (3.13)'te gösterilmiştir. Çıkış katmanındaki işlemci elemanlar genellikle düzenli iki boyutlu aralıklar olarak düzenlenir. Çıkıştaki her işlemci eleman, bütün giriş işlemci elemanlarına bağlıdır. Bağlantıların ağırlıkları verilen çıkış işlemci elemanı ile ilgili olan referans vektörünün elemanlarını oluşturur [5].

Kohonen ağının öğrenme adımları aşağıdaki gibidir.

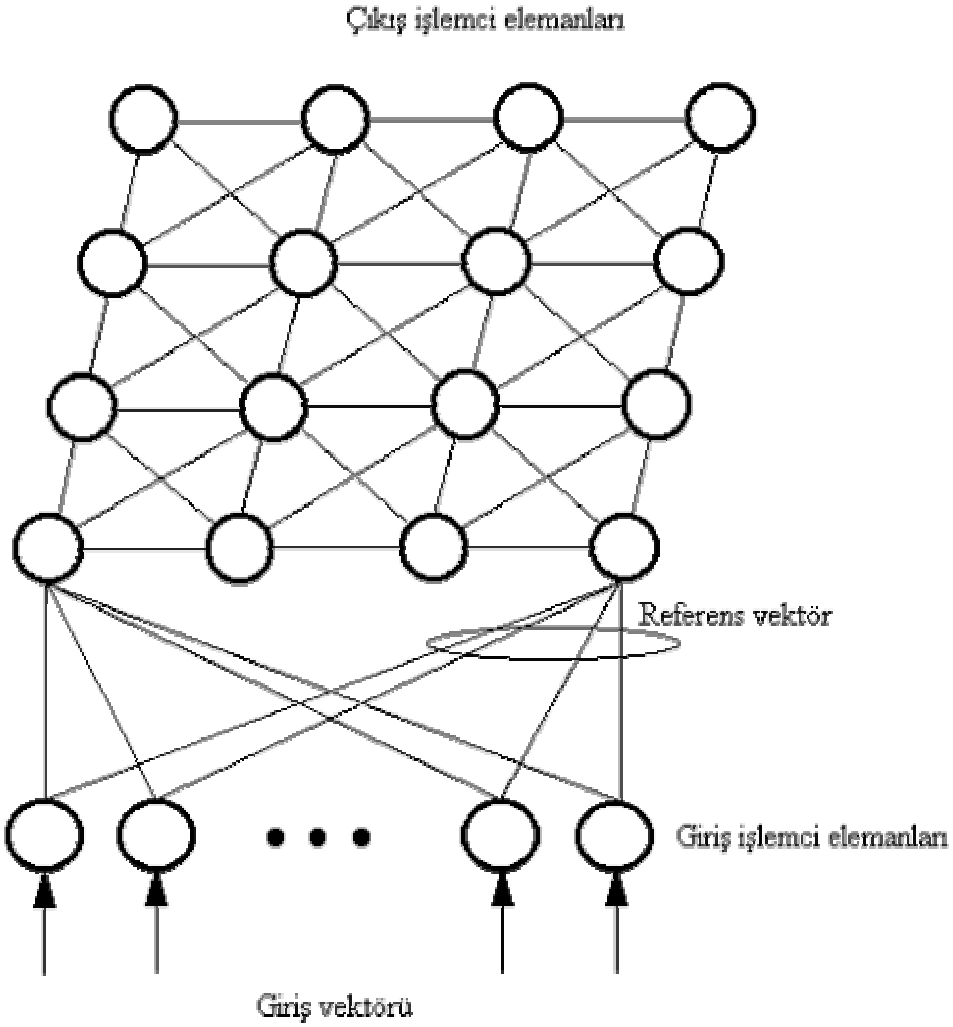
1. Çıkış işlemci elemanlarının bütün referans vektörlerini küçük rasgele değerlere çek,

2. Bir giriş desenini al,

3. Kazanan çıkış işlemci elemanını belirle (Bu giriş desenine en yakın referans vektörüne sahip işlemci elemandır. Referans vektörü ile giriş vektörü arasındaki öklit uzaklığı genellikle uzaklık ölçüsü olarak alınır.),

4. Kazanan işlemci elemanın ve onun komşularının referans vektörünü güncelleştir. Bu referans vektörleri giriş vektörüne yaklaştırılır. Bu yaklaştırma (ayarlama) kazanan işlemci eleman için en fazla ve bu işlemci elemandan uzaklaştıkça daha azdır. Öğrenme ilerledikçe komşuların sayısı azalmakta ve öğrenme sonunda sadece kazanan işlemci elemanın referans vektörü ayarlanmaktadır.

İyi öğrenmiş bir Kohonen ağında birbirine yakın çıkış işlemci elemanlarının referans vektörleri vardır. Öğrenmeden sonra bir etiketleme işlemine başlanır. Bilinen sınıfların giriş desenleri ağa giriş olarak verilir ve bu giriş desenleri tarafından aktif hale gelen çıkış işlemci elemanlarına bu sınıf etiketleri verilir. Vektör kuantalama ağında ise bir çıkış işlemci elemanı eğer yarışmayı diğer çıkış işlemci elemanlarına karşı kazanırsa bir giriş deseni tarafından aktif duruma getirildiği hatırlatılır.



Şekil 3-13 Kohonen ağı

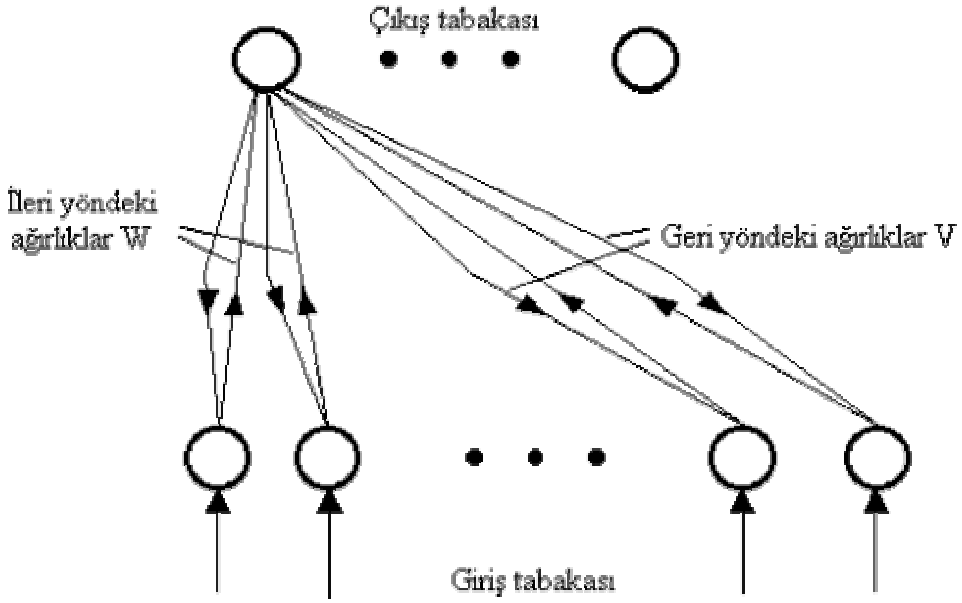
3.5.5 Uyarlamalı Rezonans (Adaptive Resonance Theory – ART) Teorisi

ART ağının, ART-1, ART-2, Fuzzy-ART, MART gibi değişik çeşitleri vardır. Şekil (3.14)'te ikili girişler kabul eden ART-1 gösterilmiştir. ART-2 gerçek değerler kabul etmektedir. Şekil (3.14)'de görüldüğü gibi, ART-1 ağının giriş ve çıkış olmak üzere iki katmanı vardır. İki katman birbirleriyle tamamen bağlantılıdır ve bağlantılar ileri ve geri yöndedir. i. çıkış işlemci elemanlarının ileri yöndeki bağlantılarının ağırlıklarının oluşturduğu w_i vektörü, temsil ettiği sınıfın bir örneğini oluşturur. İleri bağlantılarının w_i vektörlerinin toplamı ART ağının uzun-dönem (long-term) belleğini oluşturur. Bu vektörler, giriş desenine benzerlikleri ölçüsünde, çıkış işlemci elemanlarını belirlerler. Çıkış işlemci elemanlarının geri bağlantılarının w_i vektörleri ise, giriş deseninin, bellekteki desenine yeterince benzeyip benzemediğini (o sınıfa ait olup olmadığını) kesin olarak belirlemek için bir çeşit test "vigilance" amacıyla kullanılır.

Bu test vektörleri diyebileceğimiz v_i vektörleri ART ağının kısa-dönem (short-term) belleğini oluştururlar. w_i, v_i'nin normalize edilmiş halidir [5].

$$w_i = \frac{v_i}{\varepsilon + \sum v_{ji}} \tag{3.34}$$

burada ε küçük bir sabittir ve v_{ji}, v_i'nin j. elemanıdır.



Şekil 3-14 ART-1 ağı

4 GERİYE YAYILIM (BACK PROPAGATION) ALGORİTMASI

4.1 Giriş

Yapay sinir ağlarının parametrelerinin güncelleştirilmesi için literatürde en çok kullanılan yöntem geriye yayılım algoritmasıdır. Ses tanıma problemlerinden, doğrusal olmayan sistem tanıma problemlerine kadar yapay sinir ağları ile çözüm üretilen pekçok alanda kullanılır.

Geriye yayılım algoritmasının eğitimi sırasında ağ, Bölüm (3.3.4.1)'de anlatıldığı gibi, çıkış katmanında sonuç üretmek üzere, giriş örüntüleri gizli katmanlardan geçirilir. Çıkış nöronlarında elde edilen sonuç ile beklenen sonuç karşılaştırılır ve çıkış katmanının hataları bulunur. Hataların bulunmasından sonra ağ, hataların türevini çıkış katmanından geriye doğru gizli katman veya katmanlara geçirilir. Geriye yayılım algoritmasına ismini veren, algoritmanın bu özelliğidir.

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.

Her çıkış ve gizli katmandaki nöronların hata değerleri bulunduktan sonra, nöronlar kendi hatalarını azaltmak için sinaptik ağırlıkları ayarlar. Ağırlık değiştirme denklemleri, ağın performans fonksiyonunu minimum yapacak şekilde düzenlenir. Standart geriye yayılım algoritmasında performans fonksiyonu olarak, ortalama hata karesi kullanılır. Geriye yayılımın kullanılan diğer türevleri de, eksiklikler ve gereklilikler düşünülerek, performans fonksiyonu, ağ parametreler ve/veya ağırlık güncelleme denklemlerinde değişiklikler yapılarak geliştirilebilir [5-10, 12, 13].

4.2 Standart Geriye Yayılım Algoritması (Dik İniş Algoritması – Gradient Descent Algorithm)

Geriye yayılım algoritması, Widrow – Hoff öğrenme kuralının genelleştirilerek, doğrusal olmayan, türevi alınabilir transfer fonksiyonları kullanılan çok katmanlı ağlara uygulanmasıyla geliştirilmiştir. Giriş vektörleri ve bunlara karşılık düşen beklenen çıkış değerleri, ağ bir yaklaşık fonksiyona ulaşana kadar, giriş vektörlerini belirli çıkış vektörleriyle birleştirene kadar veya giriş vektörlerini tasarımcı tarafından belirlenen şekilde sınıflayana kadar ağı eğitmek için kullanılır. Bias terimleri, sigmoidal katmanı ve doğrusal çıkış katmanı olan ağlar, sınırlı sayıda süreksizliği olan herhangi bir fonksiyonu tahmin etme yeteneğine sahiptir. İleri beslemeli ağlarda kullanılan öğrenme algoritmaları, performans fonksiyonunu en küçük yapacak ağırlıkları ayarlayabilmek için, performans fonksiyonunun gradyenini kullanırlar. Geriye yayılım algoritması da, ağ boyunca gradyen hesaplamalarını geriye doğru yapar. En basit geriye yayılım öğrenme algoritması dik iniş algoritmasıdır.

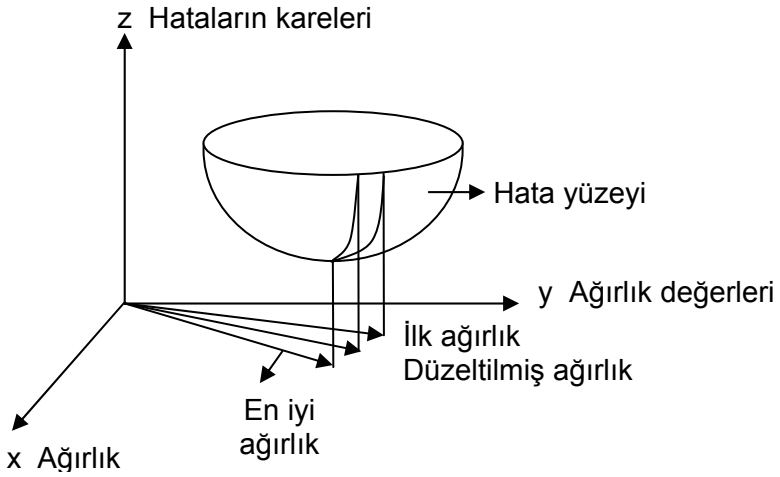
Dik inişten başka, eşlenik gradyen ve Newton yöntemleri gibi diğer standart optimizasyon tekniklerine dayanan basit algoritmaların çeşitli şekilleri vardır.

Geriye yayılım algoritmasıyla eğitilen ağlar, daha önceden tanıtılmayan giriş değerlerine mantıklı sonuçlar verir. Bu algoritmada, eğitimde kullanılan giriş vektörüne benzeyen yeni bir giriş ağa uygulandığında, yeni giriş vektörüne karşı ağın hesapladığı çıkış vektörü, eğitimde kullanılan çıkış vektörüne yakın bir sonuç verir. Bu genelleştirme özelliği, ağın olası bütün giriş çıkış çiftlerini kullanmadan da güzel sonuçlar vermesini sağlar [5-7].

4.2.1 Hatanın Minimizasyonu

Hatanın minimum yapılmasına, geometrik olarak bir yorum getirilebilir. 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. (Şekil 4.1)

Şekil (4.1)'de görülen tasın en alt kısmı, hataların kareleri toplamının en küçüklerine karşı gelmektedir. Eğitim sırasında amaç, ağırlıklar kümesinin en iyisini bulmak, yani tasa benzer şeklin en alt kısmını bulmaktır.

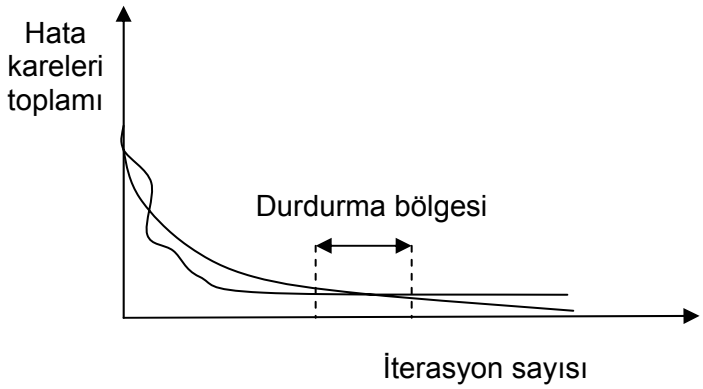


Şekil 4-1 Hata minimizasyonun geometrik yorumu

Geriye yayılım algoritması, o andaki ağırlıklar yerine, yüzey hatasının eğimini hesaplayarak amacına ulaşır. Daha sonra da bu ağırlıkların, tasın alt kısmına doğru artımsal olarak değiştirir. Artımsal olarak tasın üst kısmından alt kısmına doğru ilerleme işlemine “Dik İniş (Steepest Descent)” denir [7].

4.2.2 Yapay Sinir Ağlarında Öğrenme Ve Ezberleme Problemi

Yapay sinir ağlarında öğrenme ve ezberleme sınırının belirlenmesinde iki farklı uygulamadan söz edilir. Birinci tipteki uygulama sonlu sayıda durumdan oluşur ve bu durumların her biri eğitim çiftleri tarafından temsil edilir. Lojik işlemler, XOR problemleri bu durumlara örnektir. Eğitim kümesi, girişlerin alabileceği tüm değerleri içeriyorsa, toplam karesel hata değeri sıfıra ne kadar yaklaşırsa ağ başarımı o kadar yüksek olacaktır. İkinci tipteki 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 yapay sinir ağı, verilen noktalarda sıfıra çok yakın bir ortalama hata karesi elde edecek biçimde eğitilmişse, eğitim çiftleri kümesinde bulunmayan ancak aynı sürekli yüzey denklemlerinden elde edilen bir başka noktada, büyük hata içeren bir sonuç üretmesi olası bir durumdur. Bu nedenle eğitim kümesinin yanında, test kümesi de oluşturulmalı ve bu iki küme mümkün olduğunca az sayıda özdeş eleman bulundurmalıdır.



Şekil 4-2 Eğitimin durdurulması

Eğitim sırasında eğitim kümesine göre parametreler güncelleştirilirken, test kümesi için de gözlenen ortalama hata karesi kontrol edilmelidir. Her iki büyüklüğün de azalma gösterdiği süre boyunca genelleme mantığına uygun bir öğrenme gerçekleşmektedir. Bir süre sonra eğitim kümesi için hata azalmaya devam ederken, test kümesi için hata artmaya devam edecektir. Bu noktada eğitime son verilmelidir; çünkü kullanılan eğitim kümesindeki örüntüler için ezberleme süreci başlamıştır. Bu durum Şekil (4.2)'de gösterilmiştir [7].

4.2.3 Geriye Yayılım Algoritmasında Ağırlıkların Güncellenmesi

Geriye yayılım algoritmasında, daha önce açıklandığı gibi, giriş vektörleri ağa uygulanarak, bu giriş vektörlerine karşılık gelen çıkış değerleri hesaplanır. Ağ çıkışları ile beklenen çıkışların karşılaştırılmasıyla hata değerleri elde edilir. Hatanın azaltılması için ağırlıkların pozitif veya negatif yönde değiştirilmesi gerekliliği belirlenir. Her ağırlık değeri için değişiklik miktarı belirlenerek düzenleme yapılır. Algoritma bütün eğitim örüntüleri için, hata limit değer sınırına düşene kadar tekrar edilir.

4.2.3.1 Aktivasyon Fonksiyonunun Türevleri

Geriye yayılım algoritmasında kullanılan zincir kuralı, aktivasyon fonksiyonun türevini almayı gerektirir. Aktivasyon fonksiyonu olarak kullanılan sigmoid fonksiyonu, tanjant hiperbolik fonksiyonu ve doğrusal fonksiyon için türev denklemleri aşağıda verilmiştir.

1. Doğrusal Fonksiyon: $\varphi(v) = v$, $\varphi'(v) = 1$ (4.1)

2. Sigmoid Fonksiyonu: $\varphi(v) = \frac{1}{1+e^{-av}}$, $\varphi'(v) = a\varphi(v)(1-\varphi(v))$ (4.2)

3. Tanjant Hiperbolik Fonksiyonu: $\varphi(v) = \frac{e^{av} - e^{av}}{e^{av} + e^{av}}$, $\varphi'(v) = a(1-\varphi(v)^2)$ (4.3)

Burada a sabiti eğim parametresidir genellikle 1 seçilir; v ise net giriştir ve şu şekilde ifade edilir:

$$v_k = \sum_{j=1}^{m} w_{kj}\, x_j + b_k \tag{4.4}$$

4.2.3.2 Çıkış Katmanındaki Ağırlıklarının Güncellenmesi

n. iterasyonda, j. nöronun çıkışındaki hata işareti, j. nöron çıkış katmanındaysa;

$$e_j(n) = d_j(n) - y_j(n) \tag{4.5}$$

şeklinde tanımlanır. j nöronu için hata fonksiyonunun ani değeri ise, $\frac{1}{2}e_j^2(n)$ şeklinde tanımlanır. Ani hata değerlerinin bütün nöronlar için toplamı ise toplam hata fonksiyonunu belirtir.

$$E(n) = \frac{1}{2}\sum_{j\in C} e_j^2(n) \tag{4.6}$$

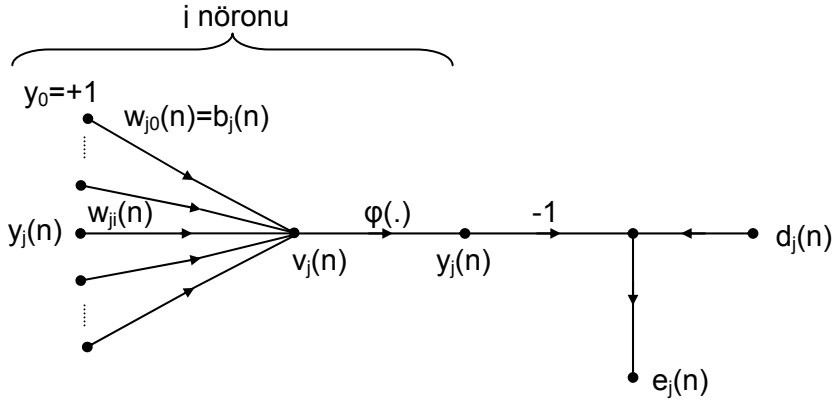
(4.6) denklemindeki C kümesi, ağın çıkış katmanındaki bütün nöronları içerir. Ortalama karesel hata fonksiyonu, N eğitim kümesindeki tüm örüntü sayısı olmak üzere, (4.7) denklemiyle bulunur:

$$E_{ort} = \frac{1}{N}\sum_{n=1}^{N} E(n) \tag{4.7}$$

Ani hata E(n) ve ortalama hata $E_{ort}(n)$, ağın bütün serbest parametrelerinin (ağırlıklar ve bias) fonksiyonudur. Verilen bir eğitim kümesi için, $E_{ort}(n)$ ağın performans

fonksiyonudur. Bu öğrenme algoritmasının amacı, performans fonksiyonunu minimum yapmak için serbest parametrelerin ayarlanmasıdır.

Bu algoritmada, ağırlıklar performans fonksiyonun azalması yönünde ayarlanır. Öğrenme sırasında performans fonksiyonu, dik iniş yöntemiyle en küçük değere indirilir. Geriye yayılım algoritmasında performans fonksiyonu olarak, ortalama karesel hata kullanılır. Ağırlıkların değişim yönünü bulmak için, E(n)'nin w_{ji} ağırlıklarına göre, negatif gradyeni hesaplanır. Böylece ağırlık değerleri, hatayı azaltacak yönde değiştirilebilir. E(n)'yi, ağırlık uzayında bir yüzey olarak düşünmek, daha kullanışlı olacaktır. Ağırlık değişimleri, hata değerinde hızlı azalışın gerçekleştiği negatif gradyen yönünde yapılmaktadır.



Şekil 4-3 Çıkış katmanındaki j nöronunun işaret akış diyagramı

Şekil (4.3)'te görüldüğü gibi j nöronu, solundaki katman tarafından oluşturulan fonksiyon işaretleri tarafından beslenir. Aktivasyon fonksiyonunun girişi, denklem (4.8)'de görüldüğü gibi hesaplanır:

$$v_j(n) = \sum_{i=0}^{m} w_{ji}(n) y_i(n) \tag{4.8}$$

burada m, toplam giriş sayısıdır (bias hariç). n. iterasyonda j çıkış nöronundaki fonksiyon işareti $y_j(n)$,

$$y_j(n) = \varphi_j(v_j(n)) \tag{4.9}$$

şeklinde gösterilebilir.

Geriye yayılım algoritmasında, ağırlıkların değişimi Widrow – Hoff kuralına benzer şekilde gerçekleştirilir. Ağın gradyeni hata fonksiyonunun ağırlıklara göre türevi alınarak bulunabilir. Zincir kuralına göre, gradyen aşağıdaki şekilde ifade edilebilir:

$$\frac{\partial E(n)}{\partial w_{ji}(n)} = \frac{\partial E(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \frac{\partial v_j(n)}{\partial w_{ji}(n)} \tag{4.10}$$

$\frac{\partial E(n)}{\partial w_{ji}(n)}$ kısmi türevi, w_{ji} ağırlığı için ağırlık uzayında arama yönünü hesaplayan duyarlılık faktörüdür.

Denklem (4.6)'da iki tarafın türevi alınırsa denklem (4.11) elde edilir.

$$\frac{\partial E(n)}{\partial e_j(n)} = e_j(n) \tag{4.11}$$

Denklem (4.5)'te, her iki tarafın $y_j(n)$'ye göre türevi alınırsa,

$$\frac{\partial e_j(n)}{\partial y_j(n)} = -1 \tag{4.12}$$

elde edilir. Denklem (4.9)'un $v_j(n)$'ye göre türevi alınarak, denklem (4.13) bulunur.

$$\frac{\partial y_j(n)}{\partial v_j(n)} = \varphi'(v_j(n)) \tag{4.13}$$

Son olarak (4.8) denkleminin $w_{ji}(n)$'ye göre türevi alınarak, denklem (4.14) elde edilir.

$$\frac{\partial v_j(n)}{\partial w_{ji}(n)} = y_i(n) \tag{4.14}$$

(4.11), (4.14) ve (4.10) denklemleri kullanılarak, $\frac{\partial E(n)}{\partial w_{ji}(n)}$ kısmi türevi bulunur:

$$\frac{\partial E(n)}{\partial w_{ji}(n)} = -e_j(n)\varphi_j'(v_j(n))y_i(n) \tag{4.15}$$

$\Delta w_{ji}(n)$, ağırlık düzeltmesi, delta kuralına göre hesaplanır.

$$\Delta w_{ji}(n) = -\eta \frac{\partial E(n)}{\partial w_{ji}(n)} \tag{4.16}$$

denklem (4.16)'da η öğrenme hızıdır. Buradaki – işareti, ağırlık uzayındaki dik inişi temsil eder. Denklem (4.15) ve (4.16) kullanılarak,

$$\Delta w_{ji}(n)=\eta \delta_j(n) y_i(n) \tag{4.17}$$

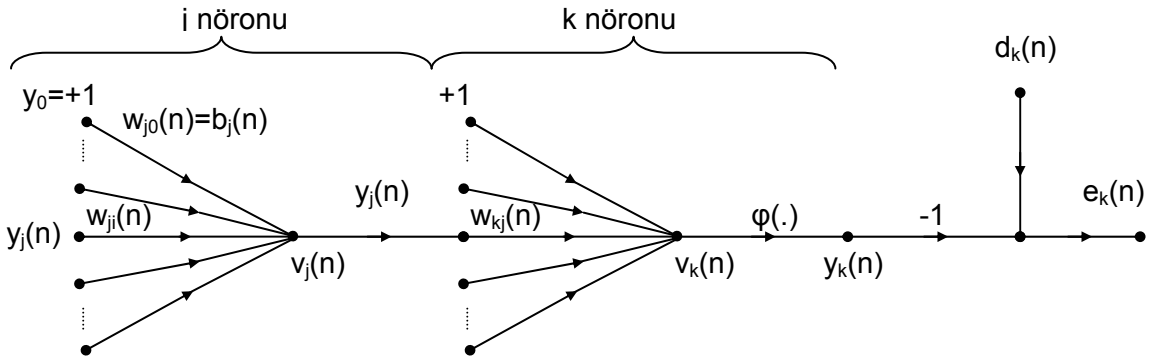
elde edilir. Burada $\delta_j(n)$ yerel gradyen şu şekilde tanımlanır:

$$\begin{aligned} \delta_j(n) &= -\frac{\partial E(n)}{\partial v_j(n)} \\ &= -\frac{\partial E(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \\ &= e_j(n) \varphi_j'(v_j(n)) \end{aligned} \tag{4.18}$$

Yerel gradyen, ağırlıktaki gerekli değişikliğe işaret eder. Denklem (4.18)'e göre, j çıkış nöronu için $\delta_j(n)$ yerel gradyeni, o nöronun hata işareti ile, aktivasyon fonksiyonunun türevinin çarpımına eşittir [5-12].

4.2.3.3 Ara Katman Ağırlıklarının Güncellenmesi

Ara katman ağırlıkları belirlenirken, çıkış katmanında olduğu gibi, çıkış değerleri hakkında bir ön bilgi bulunmamaktadır. Bu nedenle gizli katman çıkışları belirli bir değere hedeflenememektedir.



Şekil 4-4 j gizli katmanına bağlı k çıkış nöronunun işaret akış diyagramı

Şekil (4.4)'te görülen j nöronu gizli katman nöronudur. Denklem (4.18)'e göre, j gizli katman nöronunun $\delta j(n)$ yerel gradyeni aşağıdaki şekilde yeniden tanımlanır.

$$\begin{aligned}\delta_j(n) &= -\frac{\partial E(n)}{\partial y_j(n)}\frac{\partial y_j(n)}{\partial v_j(n)} \\ &= -\frac{\partial E(n)}{\partial y_j(n)}\varphi_j'(v_j(n))\end{aligned}, \qquad \text{j gizli katmanda} \tag{4.19}$$

$\frac{\partial E(n)}{\partial y_j(n)}$ kısmi türevini hesaplamak için ikinci satırda (4.13) kullanılmıştır. Şekil (4.4)'e göre

$$E(n) = \frac{1}{2}\sum_{k\in C} e_k^2(n), \qquad \text{k çıkış katmanında} \tag{4.20}$$

bulunur. (4.20) denkleminde $y_j(n)$'ye göre türev alınırsa

$$\frac{\partial E(n)}{\partial y_j(n)} = \sum_k e_k \frac{\partial e_k(n)}{\partial y_j(n)} \tag{4.21}$$

elde edilir. $\frac{\partial E(n)}{\partial y_i(n)}$ kısmi türevini bulmak için zincir kuralı kullanılır.

$$\frac{\partial E(n)}{\partial y_j(n)} = \sum_k e_k(n)\frac{\partial e_k(n)}{\partial v_k(n)}\frac{\partial v_k(n)}{\partial y_j(n)} \tag{4.22}$$

Şekil (4.4)'ten

$$\begin{aligned}e_k(n) &= d_k(n) - y_k(n) \\ &= d_k(n) - \varphi_k(v_k(n)), \quad \text{k çıkış katmanında}\end{aligned} \tag{4.23}$$

bulunur. Yani

$$\frac{\partial e_k(n)}{\partial v_k(n)} = -\varphi_k'(v_k(n)) \tag{4.24}$$

olarak yazılabilir. Yine şekil (4.4)'e göre, aktivasyon fonksiyonunun girişi de denklem (4.25)'te görüldüğü gibi yazılır:

$$v_k(n) = \sum_{j=0}^{m} w_{kj}(n)\, y_j(n) \tag{4.25}$$

burada m, k nöronuna uygulanan toplam giriş sayısıdır (bias hariç). Denklem (4.25)'in $y_j(n)$'ye göre türevi alınırsa, denklem (4.26) bulunur.

$$\frac{\partial v_k(n)}{\partial y_j(n)} = w_{kj}(n) \tag{4.26}$$

İstenen kısmi türevi elde edebilmek için denklem (4.24), (4.26) ve (4.22) kullanılır.

$$\begin{aligned} \frac{\partial E(n)}{\partial y_j(n)} &= -\sum_k e_k(n)\varphi_k'(v_k(n))w_{kj}(n) \\ &= -\sum_k \delta_k(n) w_{kj}(n) \end{aligned} \tag{4.27}$$

Son olarak, denklem (4.27), denklem (4.19)'da kullanılırsa, geriye yayılım algoritmasının $\delta_j(n)$ yerel gradyen formülü bulunur.

$$\delta_j(n) = \varphi_j'(v_j(n))\sum_k \delta_k(n) w_{kj}(n), \qquad \text{j gizli katmanda} \tag{4.28}$$

1986 yılında Rumelhart, geriye yayılım algoritmasının ağırlık güncelleme denklemine, momentum terimi ekleyerek, ağın yerel minimuma takılması olasılığını azaltmıştır. Momentum terimi eklendikten sonra ağırlık güncelleme denklemi aşağıdaki hali alır:

$$w(n+1) = w(n) + \Delta w(n) \tag{4.29}$$

$$\Delta w_{ji}(n+1) = \eta\, \delta_j(n)\, y_i(n) + \alpha\, \Delta w_{ji}(n) \tag{4.30}$$

4.2.4 Geriye Yayılım Algoritmasında Gizli Katman Nöron Sayısının Belirlenmesi

Geriye yayılım algoritmasında gizli katman nöron sayısının belirli bir kurala göre belirlenmesi mümkün değildir. Deneme yanılma yöntemi kullanılarak en iyi nöron sayısına ulaşılmaya çalışılır.

Gizli katmandaki nöron sayısı, yakınsama açısından başarı sınırı sayılmaktadır. Giriş ve çıkış verilerinin artması durumunda, gizli katmanda kullanılması gereken nöron sayısının da artacağı düşünülmektedir. Ancak gizli katman nöron sayısını arttırmanın yakınsamada iyileşme göstermemesi de mümkündür. Ağ yapısı dikkate alınarak, gizli katmandaki nöron sayısını arttırmanın, ağın giriş ve çıkış nöronlarına bağlı olarak hesap yükünü arttıracağı söylenebilir.

Lippman, tek gizli katmanlı yapay sinir ağlarında, gizli katman nöron sayısının, M çıkış nöron sayısı, N giriş nöron sayısı olmak üzere, M(N+1) değerine eşit olması gerektiğini savunmuştur. Uygulamada Lippman'ın belirttiği M(N+1) değerinin en büyük gizli katman nöron sayısı olarak kabul edilmesi ve bu değerden daha az sayıda nöron kullanılmasıyla performansın arttığı gözlenmiştir [7,14].

4.3 Hızlı Eğitim

Bölüm (4.2)'de, iki geriye yayılım algoritmasından söz edilmiştir: dik iniş ve momentum terimli dik iniş. Bu iki yöntem, uygulamada genellikle çok yavaş kalmaktadır. Bu bölümde, standart geriye yayılım algoritmasında yüz kat daha hızlı, yüksek performanslı çeşitli algoritmalardan bahsedilecektir. Bu algoritmaların hepsi toplu eğitim (batch training mode) biçiminde işletilir. Hızlı algoritmalar genel olarak iki kategoriye ayrılabilir. İlk kategorideki algoritmalar, deneme yanılma tekniklerini kullanarak, standart dik iniş (steepest descent) yönteminden daha iyi sonuçlar verebilir. Deneme-yanılma işlemlerini kullanan geriye yayılım algoritmaları; momentum terimli geriye yayılım, öğrenme hızı değişen geriye yayılım ve esnek geriye yayılım algoritmalarıdır. Hızlı algoritmaların ikinci kategorisindeki algoritmalar, standart sayısal optimizasyon yöntemlerini kullanır. Bu algoritmalar; eşlenik gradyen öğrenme algoritması, Newton öğrenme algoritmaları ve Levenberg – Marquardt öğrenme algoritmasıdır.

4.3.1 Değişen Öğrenme Hızı

Standart geriye yayılım algoritmasında, öğrenme hızı, eğitim boyunca sabit kalır. Algoritmanın performansı, öğrenme hızının doğru ayarlarına çok duyarlıdır. Eğer öğrenme hızı çok yüksek ayarlanırsa, algoritma osilasyon yapılabilir ve kararsız hale gelir. Eğer öğrenme hızı çok küçük ayarlanırsa, algoritmanın yakınsaması çok uzun sürecektir. Öğrenme hızının eğitimden önce optimum ayarlamalarının hesaplanması çok uygulamada pratik bir çözüm değildir. Gerçekten de, algoritma performans yüzeyi boyunca ilerledikçe, optimum öğrenme hızı eğitim süreci boyunca değişir. Öğrenme hızının eğitim süreci boyunca değişmesi sağlanırsa, dik iniş algoritmasının performansı gelişir. Uyarlamalı öğrenme hızı, öğrenmeyi kararlı tutarken, öğrenme adımının boyutunu mümkün olduğunca büyük tutmaya çalışır. Öğrenme hızı, yerel hata yüzeyinin zorluğuna cevap verir.

Bir uyarlamalı öğrenme hızı, öğrenme prosedüründe bazı değişiklikler gerektirir. Önce, ağın başlangıç çıkış ve hata değerleri hesaplanır. Her iterasyonda, yeni ağırlıklar ve biaslar, güncel öğrenme hızına göre hesaplanır. Daha sonra yeni çıkışlar ve hatalar hesaplanır. Momentum terimi kullanılıyorsa, hesaplanan yeni hata, eski hatayı önceden belirlenmiş bir değeri aşarsa, yeni ağırlıklar ve biaslar atılır. Bunun yanında öğrenme hızı, bir azalma faktörüyle çarpılarak azaltılır. Tersi durumda, yeni ağırlıklar ve biaslar saklanır. Eğer yeni hata, eski hatadan küçükse, öğrenme hızı bir artma faktörüyle çarpılarak arttırılır. Bu işlem öğrenme hızını, ağ büyük hata artmaları olmadan öğrenebilecek şekilde arttırır. Böylece, yerel arazi için optimuma yakın öğrenme hızı bulunmuş olur. Daha büyük öğrenme hızı kararlı bir öğrenmeyle sonuçlandığı zaman, öğrenme hızı arttırılır.öğrenme hızı, hatada bir azalmayı sağlayacak kadar büyükse, öğrenme kararlı hale gelene kadar azaltılır.

Dik iniş yönteminde olduğu gibi, değişen öğrenme hızı algoritmasında da, eğitime momentum terimi eklenerek, osilasyon problemi azaltılabilir.

4.3.2 Esnek Geriye Yayılım (Resilient Back Propagation) Algoritması

Jervis ve Fitzgerald 1993'te esnek yayılım algoritmasını tasarlayarak, literatüre tanıttılar. Bu algoritma da, Delta – Bar – Delta algoritmasında olduğu gibi, her ağırlık değeri için farklı öğrenme katsayısı kullanır. Fakat esnek yayılım algoritmasında, ağırlık değeri wij güncellenirken, Delta – Bar – Delta algoritmasının aksine gradyenin ($\delta_{ij}(n) = \frac{\partial E(n)}{\partial w_{ij}(n-1)}$) işareti kullanılır.

Esnek geriye yayılım algoritmasının ağırlık güncellenmesi denklem (4.31)'de gösterildiği gibi yapılır:

$$\Delta w_{ij}(n) = \begin{cases} -(\delta_{ij})\,\eta_{ij}(n) & \bar{\delta}_{ij}(n-1)\,\delta_{ij}(n) > 0 \\ 0 & \text{aksi halde} \end{cases} \tag{4.31}$$

Kullanılan öğrenme hızı da, denklem (4.32)'ye göre hesaplanabilir:

$$\eta_{ij}(n) = \begin{cases} \min[1{,}2\,\eta_{ij}(n-1), \eta_{max}] & \bar{\delta}_{ij}(n-1)\,\delta_{ij}(n) > 0 \\ \max[0{,}5\,\eta_{ij}(n-1), \eta_{min}] & \bar{\delta}_{ij}(n-1)\,\delta_{ij}(n) < 0 \\ \eta_{ij}(n-1) & \text{aksi halde} \end{cases} \tag{4.32}$$

Denklem (4.32)'de görülen η_{min} ve η_{max}, adım aralığı sınırlarıdır.

Çok katmanlı ağlar, gizli katmanlarında genellikle sigmoid transfer fonksiyonu kullanılır. Bu fonksiyonlara, sonsuz giriş aralığını, sonlu sayıda çıkış aralığına sıkıştırdığı için, "sıkıştırma fonksiyonları" denir. Sigmoid fonksiyonları, girişler büyüdükçe eğimlerinin sıfıra yaklaşması gerçeğine dayanarak tanımlanır. Çok katmanlı ağlarda, sigmoid fonksiyonu dik iniş yöntemi kullanılarak eğitildiğinde bir probleme yol açar. Gradyen çok küçük bir değere sahip olabileceğinden, ağırlıklar ve biaslar optimal değerlerin çok uzakta bile olsa, ağırlıkların ve biasların değişimlerinin küçük olmasına neden olur. Esnek geriye yayılım algoritmasının amacı, kısmi türevlerin değerlerinin bozucu etkilerini ortadan kaldırır. Ağırlık güncellemesinin yönünü hesaplamak için, türevin sadece işareti kullanılır. Türevin değerinin ağırlık güncellemesinde bir etkisi yoktur. Ağırlık değişiminin boyutu ayrı bir güncelleme kullanılarak hesaplanır. Her ağırlık ve bias için güncelleme değeri, performans fonksiyonunun ağırlığa göre türevi iki başarılı iterasyon için de aynı ise, δ^+, artma faktörü tarafından arttırılır. Güncelleme değeri, performans fonksiyonunun ağırlığa göre türevi bir önceki iterasyondan farklı işarete sahipse, δ^-, azalma faktörü tarafından azaltılır. Eğer türev sıfırsa, güncelleme değeri aynı kalır.

Ağırlık osilasyon yaptığı zaman, ağırlık değişimi azaltılır. Çeşitli iterasyonlar için ağırlık aynı yönde değişmeye devam ederse, ağırlık değişiminin değeri arttırılır.

Esnek geriye yayılım algoritmasının performansı eğitim parametrelerinin ayarlarına çok duyarlı değildir ve bu algoritma, buraya kadar açıklanan algoritmalardan çok daha çabuk yakınsar. Esnek geriye yayılım algoritması, elektronik ortamda sadece

az miktarda bir hafıza artışına ihtiyaç duyar. Her ağırlık ve bias için yapılan güncellemeler depolanmalıdır bu da hafızada gradyen kadar bir yer kaplar.

4.3.3 Eşlenik Gradyen Algoritması

Temel geriye yayılım algoritması ağırlıkları gradyenin azalması yönünde ayarlar. Bu yön, genellikle performans fonksiyonunun azaldığı yöndür. Bu, fonksiyonun gradyenin negatif yönünde azalmasına rağmen, daha hızlı bir yakınsama sağlamamasına neden olur. Eşlenik gradyen algoritmaları, genellikle dik inişten daha hızlı yakınsama meydana getiren birleşme yönünde bir arama oluşturur.

Eşlenik gradyen algoritmasında, ağırlık uzayında hata yüzeyi, denklem (4.33)'ü sağlar.

$$E(t) = C^T w(t-1) + \frac{1}{2} w^T(t-1) H(t) w(t-1) \tag{4.33}$$

C terimi, sabit bir matris iken, H terimi Hessian matrisini ifade etmektedir. Eşlenik gradyen yöntemi ile, hata yüzeyinde yerel minimuma ulaşmak hedeflendiğinden, minimum arama doğrultusuna v(t) atanarak denklem (4.34) elde edilmiştir.

$$v(t) = -\frac{\partial E(t)}{\partial w(t-1)} + \beta(t) v(t-1) \tag{4.34}$$

Denklem (4.34)'te görülen $\beta(t)$, Polak – Ribiere kuralına göre denklem (4.35)'teki gibi ifade edilirken, Hestenes – Stiefel kuralına göre ise denklem (4.36)'da gösterildiği gibi tanımlanır.

$$\beta(t) = \frac{[\varphi(t) - \varphi(t-1)]^T [\varphi(t)]}{[\varphi(t-1)]^T [\varphi(t-1)]} \tag{4.35}$$

$$\beta(t) = \frac{[\varphi(t-1) - \varphi(t)]^T [\varphi(t)]}{v(t)^T [\varphi(t-1)]} \tag{4.36}$$

Eşlenik gradyen algoritması, adım boyutundan çok, minimum arama doğrultusunu belirleyen bir yöntemdir.

Bu bölümde eşlenik gradyen algoritmasının dört farklı çeşidi açıklanacaktır. Buraya kadar anlatılan öğrenme algoritmalarında öğrenme hızı, ağırlık güncellemesinin uzunluğunu yani adım boyutunu hesaplamak için kullanılmıştır. Eşlenik gradyen algoritmalarının çoğunda, adım boyutu her iterasyonda değiştirilir. Adım boyutunu hesaplamak için, eşlenik eğim yönünde arama yapılır, bu işlem hat boyunca performans fonksiyonunu minimize eder. Bu bölümde beş farklı arama fonksiyonu açıklanmıştır. Bu arama fonksiyonlarının hepsi, eşlenik gradyen algoritması çeşitlerinde kullanılabilir. Bazı arama fonksiyonları, optimum seçim uygulamaya göre değişse de, belirli eğitim fonksiyonlarıyla çok iyi uyum sağlarlar.

4.3.3.1 Fletcher – Reeves Güncellemesi

Bütün eşlenik gradyen algoritmaları, ilk iterasyonda gradyenin yönünü (negatif mi yoksa pozitif mi) olduğunu arayarak başlar.

$$x_i = - \delta_{ji} \tag{4.37}$$

(4.37) denkleminde görülen x_i girişi, δ_{ji} ise gradyeni ifade eder.

Arama yönü boyunca ilerleyen optimum uzaklığı hesaplamak için hat araması gerçekleştirilir.

$$W(n+1)=w(n)+\Delta w(n) \tag{4.38}$$

$$\Delta w_{ji}(n+1) = \eta\ \delta_{ji}(n)\ x_i(n) \tag{4.39}$$

Daha sonra, arama yönünün eşleniği olan sonraki arama yönü hesaplanır. Yeni arama yönünü hesaplayan genel prosedür, yeni dik iniş yönüyle önceki arama yönünün birleştirilmesidir.

$$x(n)= - \delta(n)+\beta(n)\ x(n-1) \tag{4.40}$$

(4.40) denklemindeki $\beta(n)$ sabiti, Fletcher – Reeves güncellemesi için aşağıdaki şekilde hesaplanır:

$$\beta(n) = \frac{\delta^T(n)\delta(n)}{\delta^T(n-1)\delta(n-1)} \tag{4.41}$$

Eşlenik gradyen algoritması, değişken öğrenme hızlı geriye yayılım algoritmasından çok daha hızlıdır. Bu algoritma, elektronik ortamda basit algoritmalardan sadece

biraz fazla belleğe ihtiyaç duyar. Bu yüzden ağırlık sayısı çok olan ağlar için iyi bir seçim olacaktır.

4.3.3.2 Polak-Ribiére Güncellemesi

Eşlenik gradyen algoritmasının başka bir çeşidi, Polak ve Ribiére tarafından geliştirilmiştir. Fletcher – Reeves algoritmasında olduğu gibi arama yönü her iterasyonda hesaplanır:

$$x(n)= - \delta(n)+\beta(n)\, x(n-1) \tag{4.42}$$

Polak ve Ribiére güncellemesinde $\beta(n)$ sabiti aşağıdaki şekilde hesaplanır.

$$\beta(n) = \frac{\Delta\delta^T(n-1)\,\delta(n)}{\delta^T(n-1)\,\delta(n-1)} \tag{4.43}$$

Δ terimi burada iç çarpımı ifade etmektedir.

Polak – Ribiére güncellemesinin performansı, Fletcher – Reeves algoritmasına benzer. Verilen bir problem için hangi algoritmanın en iyi sonuç vereceğini tahmin etmek zordur. Polak ve Ribiére için gereken bellek (4 vektör), Fletcher – Reeves için gerekenden (3 vektör) biraz fazladır.

4.3.3.3 Powell – Beale

Bütün eşlenik gradyen algoritmalarında, arama yönü periyodik olarak gradyenin negatifine ayarlanır. Ayarlama noktası, iterasyon sayısı ağ parametrelerinin (ağırlıklar, biaslar) sayısına eşit olduğu zaman gerçekleşir. Ancak eğitimin etkinliğini geliştirmek için farklı ayarlama yöntemleri de kullanılabilir. Bu ayarlama yöntemlerinden biri Powell tarafından geliştirilmiştir. Bu yöntem, daha önce Beale tarafından geliştirilen bir başka yöntem üzerine kurulmuştur. Bu teknikte, güncel gradyen ve önceki gradyen arasında küçük bir ortogonallik varsa, ayarlamalar baştan başlatılır.

$$\left|\delta^T(n-1)\,\delta(n)\right| \geq 0{,}2\left\|\delta(n)\right\|^2 \tag{4.44}$$

Denklem (4.44)'teki koşul sağlanırsa, arama yönü gradyenin negatifine ayarlanır.

Verilen herhangi bir problemin performansını tahmin etmek zor olsa da, Powell – Beale algoritması bazı durumlar için Polak – Ribiére algoritmasından daha iyi bir performansa sahip olur. Powell – Beale algoritması için gereken bellek (6 vektör), Polak – Ribiére için gerekenden (4 vektör) biraz daha fazladır.

4.3.3.4 Orantılı Eşlenik Gradyen

Buraya kadar anlatılan eşlenik gradyen algoritmalarında, her iterasyonda hat araması yapılaması gerekir. Bu hat aramasının hesaplanması zordur, çünkü ağın cevabının her iterasyon için çeşitli zamanlarda hesaplanması gerekir. Orantılı eşlenik gradyen Moller tarafından, hat aramasının zaman alıcılığından kaçınmak amacıyla geliştirilmiştir. Algoritmanın temeli, model – emniyet bölgesi yaklaşımını, eşlenik gradyen yaklaşımıyla birleştirmektir.

Bu algoritmada kullanılan parametreler, buraya kadar açıklanan algoritmaların parametrelere ek olarak σ ve λ 'dır. σ parametresi, ikinci türev yaklaşımı için, ağırlıkların değişimini hesaplar. λ parametresi ise, Hessian matrisinin belirsizliğini düzeltir.

Bu algoritma, diğer birleşik dik iniş algoritmalarına göre, daha fazla iterasyonda yakınsar. Ancak her iterasyonda yapılan hesaplama sayısı daha azdır çünkü hat araması hesaplaması yapılmaz. Bu algoritma için gereken bellek, yaklaşık Fletcher – Reeves için gereken bellek kadardır.

4.3.4 Newton Öğrenme Algoritmaları

Eeşlenik gradyen öğrenme algoritmasına alternatif olarak sunulan Newton yöntemlerinde, temel adım Hessian matrisini elde etmektir. Hessian matrisi, performans fonksiyonunun ağırlıklara göre ikinci dereceden türevlerinden oluşan bir matristir. Hessian matrisi, ağırlık uzayının farklı doğrultularındaki gradyen değişimini gösterir.

$$H(n) = \frac{\partial^2 E(n)}{\partial w^2 (n-1)} \tag{4.45}$$

Burada H Hessian matrisi, n iterasyon sayısı, E performans fonksiyonu, w ağın sinaptik ağırlığıdır. Performans fonksiyonu, duruma göre toplam ani hata veya ortalama karesel hata olarak alınabilir. Bu çalışmada, ileri beslemeli ağların çoğunda olduğu gibi, performans fonksiyonu olarak, ortalama karesel hata kullanılmıştır.

$$E(n) = E_{ort}(n) = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{2}\sum_{j\in C} e_j^2(n) \tag{4.46}$$

Burada N eğitim kümesindeki tüm örüntü sayısını, e_j hata işaretini, C kümesi, ağın çıkış katmanındaki bütün nöronları içerir. d_j istenen değer, y_j ağın çıkışı olmak üzere hata işareti,

$$e_j(n) = d_j(n) - y_j(n) \tag{4.47}$$

olarak bulunabilir. Hessian matrisi hesaplandıktan sonra, tersi bulunarak ağırlıklar yenilenebilir. Ancak Hessian matrisi çok karmaşık ve ileri beslemeli bir yapay sinir ağı için hesaplanması zor bir matristir. Newton yöntemlerinin içinde, ikinci dereceden türevlerin hesaplanmadan işlem yapılan bir sınıf vardır. Bu sınıftaki yöntemler, quasi – Newton yöntemleri olarak adlandırılırlar. Quasi – Newton yöntemleri, algoritmanın her iterasyonunda, Hessian matrisinin yaklaşık bir şeklini kullanır.

Hessian matrisinin yaklaşık değeri şu şekilde bulunabilir:

$$H(n) = J^T(n)J(n) \tag{4.48}$$

(4.48) denklemindeki J matrisi, Jakobien matrisi olarak adlandırılır ve ağ hatalarının ağırlıklara göre birinci türevlerinden oluşur:

$$J(n) = \frac{\partial e(n)}{\partial w(n-1)} \tag{4.49}$$

(4.49) denkleminde e, ağ hataları vektörüdür. Jakobien matrisi, hesaplamada Hessian matrisinden daha kolay olduğu için tercih edilir.

4.3.5 Levenberg – Marquardt Algoritması

Levenberg – Marquardt algoritması da quasi – Newton yöntemleri gibi, Hessian matrisinin bir yaklaşık değerini kullanırlar. Levenberg – Marquardt algoritması için Hessian matrisinin yaklaşık değeri şu şekilde bulunabilir:

$$H(n) = J^T(n)J(n) + \mu I \tag{4.50}$$

(4.50) denklemindeki μ Marquardt parametresi, I ise birim matristir. Buradaki J, Jakobien matrisi denklem (4.49)'daki gibi hesaplanır. Jakobien matrisi, hesaplamada Hessian matrisinden daha kolay olduğu için tercih edilir. Ağın gradyeni,

$$\delta(n) = J^T(n)e(n) \tag{4.51}$$

olarak hesaplanır ve ağırlıklar (4.52) denklemine göre güncelleştirilir:

$$w(n+1) = w(n) - [H(n)]^{-1}\delta(n) \tag{4.52}$$

Marquardt parametresi, μ, skaler bir sayıdır. Eğer μ sıfırsa, bu yöntem yaklaşık Hessian matrisini kullanan Newton algoritması; eğer μ büyük bir sayı ise, küçük adımlı dik iniş yöntemi haline gelir. Newton yöntemleri, en küçük hata yakınlarında daha hızlı ve kesindir. Her başarılı adımdan sonra, yani performans fonksiyonunun azalmasında μ azaltılır ve sadece deneme niteliğindeki bir adım performans fonksiyonunu yükseltecekse μ arttırılır. Bu yöntemle, algoritmanın her iterasyonunda, performans fonksiyonu daima azaltılır.

Genel olarak Levenberg – Marquardt algoritması yavaş yakınsama probleminden etkilenmez. Burada hedef, performans fonksiyonun en küçük yapacak ağırlık değerini bulmaktadır.

5 YAPAY SİNİR AĞI İLE YÜKSEK GERİLİM İZOLATÖRLERİNİN DARBE ATLAMA GERİLİMLERİNİN BULUNMASI

5.1 Giriş

Yüksek gerilim tekniğinde, iletim sistemlerinde, transformatör ve yüksek gerilim bağlama cihazlarında kullanılan izolatörler, iletkenlerin birbirine ve toprağa karşı yalıtımını sağlarlar. İzolatörler elektriksel bakımdan yalıtım sağlamanın yanında, mekanik bakımdan da gövde, destek görevlerini görür ve bağlantıların yapılmasını sağlarlar. Bu sebeple izolatörler, elektrik sistemlerinde önemli bir yer tutar. İzolatörün yapısal özellikleri, etkisi altında bulunduğu sıcaklık, nem, basınç, kirlenme gibi ortam koşulları, yükselti, gerilim türü gibi çalışma koşulları ve mekanik etkiler, izolatörlerin güvenilirliğini etkiler. Bu etkenlerin izolatörler üzerindeki etkisini incelemek, tasarımları geliştirmek, davranışlarını belirlemek ve olayları anlamak bakımından çok önemlidir. Bu amaçla pek çok araştırma yapılmaktadır [15-19]. Bu araştırmalardan biri de, farklı basınçlarda, negatif ve pozitif darbe gerilimi altında, değişik özellikli izolatörlerde, yüzeysel boşalma (atlama) olayıdır.

İzolatörler, uygulamada bulundukları yerin ortam koşullarına ve yükseltisine göre, farklı basınçlar altında bulunabilirler. Farklı basınçlarda izolatörlerin davranışlarının belirlenmesi ve açıklanması yönündeki çalışmalar devam etmektedir [16, 19]. İzolatörlerin normal atmosfer basıncı için verilmiş olan atlama verilerinden, bu basınç yakınındaki diğer basınçlar için, standartlarda verilen düzeltme katsayıları kullanılarak belirlenir. Ancak normal atmosfer basıncına yakın olmayan basınçlarda, bu belirlemenin nasıl yapılacağı konusu, çalışmaya açıktır.

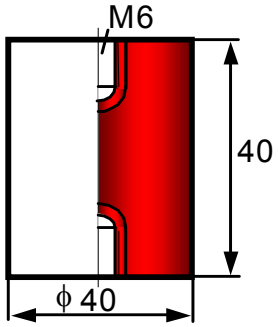
Basınç, nem, sıcaklık gibi koşullara bağlı olarak atlama gerilimi hesaplanmasında genellikle deneye dayalı olarak yazılmış (amprik) bağıntılar kullanılır. Gazlarda Townsend teorisinin geçerli olduğu durumlarda Paschen bağıntısı, düzgün olmayan alanda kanal teorisi bağıntısı kullanılır. Yüzeysel boşalma için genelde parametrelere göre tek tek deney yapılarak, elde edilen verilere uyan bağıntılar çıkarılır. Bu amaçla yapılan çalışmalarda, genel bir bağıntı bulmak zordur. Literatürü araştırıp duruma uyan veya yakın olan bağıntılardan yararlanılır veya onlarda

değişiklik yapılarak kullanılabilir. Ancak deney yapmak zaman alan pahalı bir yöntem olduğu için, daha çok sayısal yöntemler tercih edilmektedir. Bu çalışmalardan birisi de, sınırlı sayıdaki atlama verilerinden yararlanarak, bir yapay sinir ağı algoritması kullanmaktır. Böyle bir amaçla yapay sinir ağlarının kullanılması, deneysel çalışmaların zorluklarını, zaman alıcılığını ve pahalılığını gidermesi yanında hesaplamada hem daha kolay ve hızlı, hem de daha genel sonuç alınmasını sağlar [20].

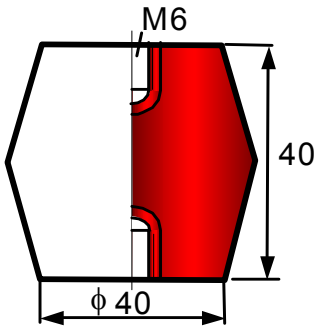
Bu çalışmada, farklı basınç ve ortam koşulları altında, izolatörlerin darbe atlama gerilimlerinin hesaplanmasında yapay sinir ağı kullanılarak, diğer yöntemlere göre farklı bir yaklaşım açıklanmıştır.

5.2 Atlama Gerilimi Deneyi

Yüksek gerilim izolatörlerinin alçak basınçta darbe atlama geriliminin sayısal belirlenmesinde veri olarak kullanılan gerilim değerleri, deneysel çalışma ile elde edilmiştir. Deneylerde düz ve mekik biçimi, bina içi türden epoksi reçineden yapılmış iki tür izolatör kullanılmıştır.



(a) Düz izolatör

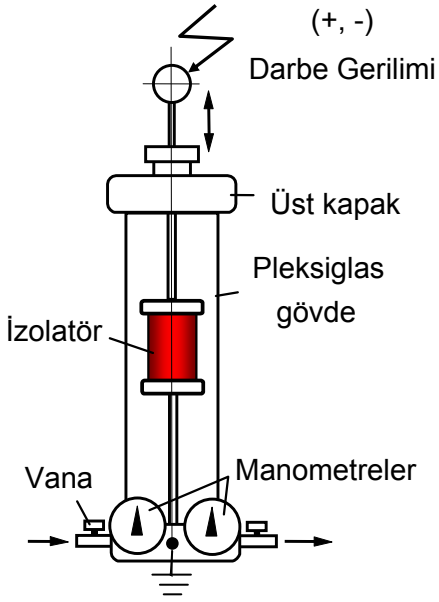


(b) Mekik izolatör

Şekil 5-1 Atlama deneyinde kullanılan yüksek gerilim izolatörleri

Her biri 40 mm boyunda olan izolatörler, biri gerilim kaynağına bağlı diğeri topraklı düzlem – düzlem elektrotlar arasına konmuş, bu düzen de pleksiglastan yapılmış olan 10^{-3} – 10^{-6} bar'lık bir basınç kabına yerleştirilmiştir. Kabın basıncı bir vakum pompası ile ayarlanmış ve basınç, kap üzerindeki manometreden okunmuştur.

Deneyler standart yıldırım darbe gerilimi ile yapılmıştır. Darbe gerilimleri, 280 kV, 200 J'luk bir darbe üretecinden elde edilmiştir. Elde edilen gerilimler, devrenin çıkışına bağlı bir kapasitif gerilim bölücü ve bir darbe tepe değer voltmetresi ile ölçülmüş ve dalga şekli bir osiloskoptan izlenmiştir. Deneyler hem pozitif hem de negatif kutuplu darbe gerilimleri ile yapılmıştır. Her durum için deneysel olarak %50 atlama gerilimi saptanmış ve bu değer atlama gerilimi olarak alınmıştır.



Şekil 5-2 Deney düzeni

Deneyler atmosfer basıncından (~1000 mbar) 100 mbar aralıklarla 100 mbar'a kadar azaltılarak tekrarlanmıştır. Her basınç basamağında kararlı basınç elde edildikten sonra pozitif ve negatif darbe atlama gerilimi deneysel olarak saptanmıştır. Basıncın değişimini eğitim kümesinde değişken parametre olarak kullanırken, atmosfer basıncı (760 mmHg = 1013 mbar ~ 1000 mbar = 1 bar) referans olarak alınıp, ölçüm yapılan basınç değerleri bu referans değerine göre yazılmıştır.

Benzer çalışma, elektrotlar arasında izolatör yokken 40 mm elektrot açıklığı için yinelenmiştir.

Deneyler, değişik alçak basınçlar yanında, farklı sıcaklık ve bağıl nem koşullarında gerçekleştirilmiştir [20].

5.3 Uygulamada Kullanılan Yapay Sinir Ağı

Çalışmada eğiticili öğrenmeli, ileri beslemeli, çok katmanlı bir yapay sinir ağı kullanılmıştır. Eğiticili öğrenmede, nöronları birleştiren ağırlıklar, eğitici tarafından ayrıntılı hata bilgileri ağa uygulanacak şekilde yerleştirilirler. Pek çok durumda ağ, giriş – çıkış çiftleri kullanılarak eğitilir. Bu çalışmada ele alınan atlama geriliminin bulunması probleminde, farklı izolatör tipleri (silindirsel, etekli, eteksiz izolatör, mekik izolatör) ve yalnızca atlama aralığı, ortam koşulları (bağıl nem, sıcaklık ve basınç) değerlerine karşılık, negatif ve pozitif darbe atlama gerilim değerleri bilinmektedir. Problemde giriş – çıkış eğitim kümesi bilindiğine göre, eğiticili öğrenmeli bir yapay sinir ağı kullanılması uygundur.

Eğiticili öğrenmeli en basit ağ yapısı, bir giriş ve bir çıkış katmanı bulunan, iki katmanlı, ileri beslemeli ağdır. Çıkış katmanındaki herbir nöron, düzeltilebilen ağırlıklar yoluyla, bütün giriş nöronlarından işaret alır. Ancak iki katmanlı ileri beslemeli yapay sinir ağları, sadece doğrusal ayrıştırılabilir fonksiyonları hesaplayabilir. Daha karmaşık problemlerin çözümünde, birden fazla uyarlamalı ağırlıklı katmana sahip olan bir ileri beslemeli ağ kullanılmalıdır. Atlama geriliminin bulunması problemi de, doğrusal olmayan bir problemdir. Bu sebeple çözümde kullanılacak ağ da, doğrusal olmayan bir yapıda olmalı, yani çok katmanlı olmalıdır.

Çok katmanlı yapay sinir ağlarında, birden çok gizli katman kullanılması, pekçok problem için hatanın değerini fazla etkilemediği gibi, sonuçların kesinliğinin artmasına da fazla katkıda bulunmaz. Bununla beraber, ikinci bir gizli katmanın varlığı, hesaplama süresinin uzamasına sebep olur. Ele alınan atlama geriliminin çözülmesi probleminde de, tek gizli katman bulunan bir yapay sinir ağı yapısı kullanılmıştır.

Problemin yapay sinir ağı ile çözümünde, basınç, sıcaklık ve bağıl nem gibi koşulların yanında, izolatörün varlığı ve izolatör türü de değişken parametre olarak kullanılmıştır. İzolatörlerin boyu ve arada izolatör olmayan durumdaki elektrot açıklığı eşit olduğu için, elektrot açıklığı giriş örüntülerinde kullanılmaz.

Yapay sinir ağı incelemesinde izolatör türü ve varlığının değişken parametre olarak ele alınabilmesi için bir kodlama yapılmıştır. Elektrotlar arasında izolatör olmaması

(1) durumu olarak, mekik izolatörün bulunduğu durum (2) durumu olarak ve düz izolatörün bulunduğu durum da (3) durumu olarak kodlanmıştır. Deneyler ve kodlama sonucunda eğitim kümesi elde edilmiştir (Tablo 5.1).

Tablo 5-1 Darbe atlama geriliminin YSA ile belirlenmesinde kullanılan eğitim kümesi

Giriş verileri				Çıkış verileri	
Kod	t (^oC)	Nem (%)	(1 – p) (bar)	+ U_a (kVtepe)	- U_a (kVtepe)
2	26,0	66,9	0,9	13,00	13,30
2	26,0	66,9	0,4	67,85	68,14
2	26,0	66,9	0,0	101,20	110,00
3	27,4	74,0	0,8	30,17	31,12
3	27,4	74,0	0,3	72,07	78,54
3	27,4	74,0	0,1	95,64	99,59
1	26,0	66,9	0,6	72,17	58,28
1	26,0	66,9	0,5	76,27	61,31
1	26,0	66,9	0,2	87,11	82,24
1	23,0	38,0	0,7	53,41	39,62
1	23,0	38,0	0,6	67,20	50,31
1	23,0	38,0	0,5	69,14	60,66
1	23,0	38,0	0,0	101,80	119,54

Tablo (5.1)'de görüldüğü gibi, izolatör tipi, basınç, sıcaklık ve bağıl nem, giriş vektörünü; pozitif ve negatif darbe atlama gerilimi değerleri ise, çıkış vektörünü oluşturmaktadır.

Yapay sinir ağı, 4 giriş nöronu ve 2 çıkış nöronundan meydana gelmektedir. Tek gizli katman kullanılan yapay sinir ağında öğrenme algoritması olarak, momentum terimli geriye yayılım algoritması seçilmiştir. Bu algoritma, Bölüm (4.2)'de anlatılan dik iniş (steepest descent) yöntemiyle ağırlıklarını hesaplar. Algoritma, Matlab 5.0 Neural Network Toolbox kullanılarak yazılan bir programla [Ek B1] uygulanmıştır (denklem (4.5) – denklem (4.30)).

Eğitimin tamamlanmasından sonra, ağın güvenilirliğini sınamak amacıyla, eğitim kümesinden bir örüntü kullanılarak bir test kümesi oluşturulmuş ve gerçek çıkış ile yapay sinir ağının hesaplamış olduğu çıkış karşılaştırılarak, hata bulunmuştur.

Test kümesi, arada izolatör yokken (kod 1) 23˚C sıcaklıkta, 300 mbar basınçta (0,7) ölçülen atlama gerilimi değerlerinden oluşmaktadır. Bu durumda test kümesinin giriş vektörü [1 23 38 0,7] ve istenen çıkış vektörü ise [53,41 39,62] olacaktır.

5.3.1 Eğitim Kümesinin Normalizasyonu

Geriye yayılım algoritmasında, yerel gradyenin hesaplanabilmesi için hataların türevinin alınması gerekmektedir. Bu sebeple, kullanılan aktivasyon fonksiyonunun da türevi alınabilir olması gerekir. Pek çok yapay sinir ağı yapısında olduğu gibi, bu çalışmada da logaritmik sigmoid fonksiyonu kullanılmıştır (Bölüm (2.5.1)). Sigmoid fonksiyonu, sonsuz giriş aralığını, sonlu sayıda çıkış aralığına (0 – 1) sıkıştırır.

$$\varphi(v) = \frac{1}{1 + e^{(-a\,v)}} \tag{5.1}$$

Sigmoid fonksiyonunun çıkışı 0 – 1 arasında olduğu için, eğitim kümesinin çıkışı da 0 – 1 arasında olmalıdır. Bununla birlikte, sigmoid fonksiyonun hemen doymaya girmemesi için, giriş değerlerinin de yeterince küçük olması gerekir. Sigmoid fonksiyonunun bu özelliklerinden dolayı, ağ eğitilmeden önce giriş ve çıkış verileri normalize edilerek, istenen sınırlar içinde olmaları sağlanmalıdır. Bu çalışmada normalizasyon, giriş ve çıkış verileri 0 – 1 aralığında kalacak şekilde yapılmıştır.

Giriş ve çıkış nöronları tek tek, o nörondaki en büyük değere bölünürse, giriş ve çıkış vektörleri 0-1 arasında değer alır.

5.3.2 Yapay Sinir Ağında Kullanılan Parametrelerin Bulunması

Yapay sinir ağlarında ağırlıkların hesaplanması için kullanılan parametrelerin hesaplanabilmesi için kesin bir yöntem yoktur. Tavsiye edilen yöntemler de, her problem için en uygun sonucu vermeyebilir. Geriye yayılım algoritmasıyla eğitilen bir yapay sinir ağında, öğrenme hızı, momentum katsayısı, gizli katmandaki nöron sayısı gibi parametrelerin bulunması için en etkili yöntem deneme yanılma yöntemidir.

Bu çalışmada çözülecek olan, yüksek gerilim izolatörlerinin darbe atlama gerilimlerinin yapay sinir ağı ile bulunması probleminde de parametreler deneme yanılma ile belirlenmiştir. Öğrenme hızı, momentum katsayısı ve gizli katmandaki nöron sayısı gibi ağ parametreleri, eğitim ve test sürecinde, ağ hatasını en küçük yapacak şekilde seçilmiştir.

Standart geriye yayılım algoritmasında, öğrenme hızı, sonuca ulaşma hızı ile doğrudan ilgilidir. Öğrenme hızı çok küçük seçilirse, ağırlıklardaki değişim bir iterasyondan diğerine geçerken az olacaktır. Bu sebeple sonuca ulaşmak uzun zaman alır, öğrenme yavaştır. Buna karşılık, öğrenme hızı çok büyük seçilirse,

öğrenme hızlı olmasına rağmen, çözüm civarında salınım yaparak yakınsamayı zorlaştırır. Hem hızlı bir öğrenme sağlamak, hem de osilasyonu engellemek için, standart geriye yayılım algoritmasına bir de momentum terimi eklenir. Öğrenme hızı ve momentum katsayısı, genellikle 0 – 1 arasında olacak şekilde seçilir.

Öğrenme hızı ve momentum katsayısını bulmak için yapılan hesaplamalar, 10 nöronlu tek katmanlı bir ağ için yapılmıştır. Ağ için en uygun parametreleri bulmak amacıyla, öğrenme hızı (η) ve momentum katsayısı (α) için denemeler yapılmıştır. Momentum katsayısı α = 0,1'de sabit tutulup, öğrenme hızı (η) 0,1'den 0,9'a kadar değiştirilerek en uygun öğrenme hızı aranmıştır (Tablo (5.2)). Benzer çalışma, öğrenme hızı η = 0,1'de sabit tutulup, momentum katsayısına 0,1 ile 0,9 arasında değerler verilmiş ve sonuçlar Tablo (5.3)'te gösterilmiştir. 5000 iterasyon için yapılan denemelerde, öğrenme hızı ve momentum katsayısı her deneme için sabit tutulmuştur.

Bu çalışmada eğitim için toplam karesel hata (sum squared error, sse) ve hatanın etkin değeri (root mean square, rms), test için ise ortalama mutlak hata (mean absolute error, mae) kullanılmıştır.

$$\text{sse} = \sum_{j=1}^{N}\sum_{k=1}^{n_o}(d_{jk} - y_{jk})^2 \tag{5.2}$$

$$\text{rms} = \sqrt{\frac{1}{N\,n_o}\sum_{j=1}^{N}\sum_{k=1}^{n_o}(d_{jk} - y_{jk})^2} \tag{5.3}$$

$$\text{mae} = \frac{1}{n_t\ n_o}\sum_{j=1}^{n_t}\sum_{k=1}^{n_o}\left[\frac{\left|d_{jk} - y_{jk}\right|}{d_{jk}}\right] 100 \tag{5.4}$$

Burada N eğitim kümesindeki tüm örüntü sayısını, n_o çıkış katmanındaki nöron sayısını, n_t test kümesindeki örüntü sayısını, d_{jk} istenen çıkışı, y_{jk} ise ağın çıkışını göstermektedir.

Tablo 5-2 Öğrenme hızının hatalara etkisi
[sse: toplam karesel hata (sum squared error);
mae: ortalama mutlak hata (mean absolute error)]

Gizli katman nöron sayısı = 10 İterasyon sayısı = 5000		
α = 0,1		
η	sse	mae (%)
0,1	0,0638	2,22
0,2	0,6590	2,03
0,3	0,0491	3,68
0,4	0,5100	1,88
0,5	0,0364	2,48
0,6	0,0372	4,55
0,7	1,1914	6,31

Tablo 5-3 Momentum katsayısının hatalara etkisi
[sse: toplam karesel hata (sum squared error);
mae: ortalama mutlak hata (mean absolute error)]

Gizli katman nöron sayısı = 10 İterasyon sayısı = 5000		
η = 0,1		
α	sse	mae (%)
0,9	0,0638	2,22
0,8	0,0674	2,22
0,7	0,0680	1,60
0,6	0,0679	2,38
0,5	0,0700	1,65
0,4	0,0657	1,53
0,3	0,0774	2,31
0,2	0,0665	1,65
0,1	0,0669	1,96

Tablo (5.2)'de ve Tablo (5.3)'te görüldüğü gibi, yapay sinir ağı için en uygun öğrenme hızı η = 0,5 ve en uygun momentum katsayısı α = 0,9 olarak belirlenir.

En uygun öğrenme hızı ve momentum katsayısı bulunduktan sonra, bu değerler gizli katmandaki nöron sayısını bulabilmek için kullanılmıştır. Öğrenme hızı η = 0,5 ve momentum katsayısı α = 0,9 iken, gizli katmandaki nöron sayısı, 4'ten 15'e kadar değiştirilerek, en iyi eğitim ve test hatasını veren değer aranmıştır. 5000 iterasyon için yapılan denemelerde, gizli katmandaki nöron sayısının 7 olmasının en uygun sonucu verdiği gözlenmiştir (Tablo (5.4)).

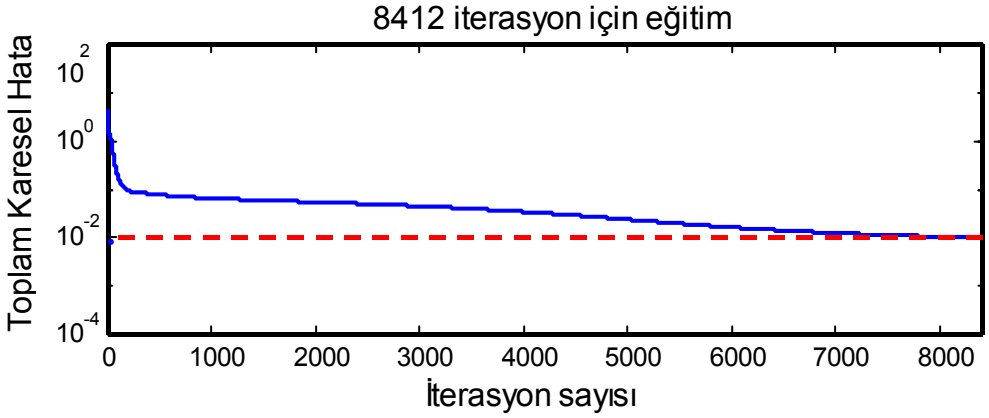
Tablo 5-4 Gizli katmandaki nöron sayısının hatalara etkisi

İterasyon sayısı = 5000		
Gizli katman nöron sayısı	sse	mae (%)
4	0,0347	4,56
5	0,0373	3,70
7	0,0303	1,66
8	0,0360	2,86
10	0,0364	2,48
15	0,0496	3,72

Tablo (5.2), Tablo (5.3) ve Tablo (5.4)'ten görüldüğü gibi, atlama gerilimi bulunması probleminin çözülmesi için kullanılacak yapay sinir ağının parametreleri α = 0,5; η = 0,9; gizli katman nöron sayısı = 7 olarak belirlenmiştir.

5.3.3 Algoritmanın Uygulanması

En uygun öğrenme hızı, momentum katsayısı ve gizli katmandaki nöron sayısı bulunduktan sonra, Şekil (5.3)'te görüldüğü gibi eğitim, toplam karesel hata 0,01'den küçük olana kadar sürdürülmüştür.



Şekil 5-3 Toplam karesel hatanın iterasyon sayısına göre değişimi

Öngörülen toplam karesel hataya 8412 iterasyonda ulaşmış ve bu iterasyon sonunda hatanın etkin değeri %1,96, ve ortalama mutlak hata %0,93 olarak bulunmuştur. Test kümesinin ağa uygulanması, Tablo (5.5)'te görüldüğü gibi, gerçek gerilim değerleri ile yapay sinir ağı çıkışlarını karşılaştırarak, ağın güvenilirliğini görmeye yardımcı olur.

Tablo 5-5 YSA sonuçları ile gerçek değerlerin karşılaştırılmasına bir örnek

Gizli katman nöron sayısı = 7; α = 0,9; η = 0,5 İterasyon sayısı = 8412		
Atlama gerilimi (kVtepe)	Gerçek değerler	YSA değerleri
Pozitif darbe	53,41	53,30
Negatif darbe	39,62	40,19

Test kümesi olarak eğitim kümesinden bir örüntü seçilmiş olmasına rağmen, gerçek değerler ile, yapay sinir ağı değerleri arasındaki fark, eğitimdeki hatadan kaynaklanmaktadır. Eğitim sonunda ağın 0,01 olan toplam karesel hatası, bulunan ağırlıkların optimum değerlerinden uzakta olmasına, dolayısıyla yapay sinir ağı çıkışları ile beklenen değerler arasında % 0,1 farka neden olur [20].

5.4 Sonuç

Bu çalışmada, yüksek gerilim izolatörlerinin darbe atlama gerilimleri, farklı bir yaklaşım olarak yapay sinir ağı algoritması ile belirlenmiştir. Değişik ortam koşulları altında, izolatörlerin darbe atlama gerilimlerinin hesaplanmasında geriye yayılım algoritması kullanan bir ileri beslemeli çok katmanlı bir yapay sinir ağı tasarlanmıştır. Öğrenme hızı, momentum katsayısı ve gizli katmandaki nöron sayısı gibi ağ parametrelerinin doğru sonuç verecek en uygun değerlerini belirlemek için detaylı çalışmalar yapılmıştır. Karar verilen parametrelerin uyumu, sonuçlarda kendini göstermiştir. Yapılan çalışmalardan görüldüğü gibi, pozitif ve negatif darbe gerilimlerinin deneysel sonuçları ile yapay sinir ağı çıkışları arasındaki ortalama mutlak hata değerleri, denemelerin çoğunda % 3'ten küçüktür.

Geriye yayılım algoritması, optimizasyon tekniği olarak, dik iniş yöntemini kullandığı için, ilk koşullara çok duyarlıdır. İlk atanan ağırlık vektörü, hata yüzeyinin minimumuna çok yakın ve eğimli bir bölgede ise, yapay sinir ağının öğrenmesi çabuk olacaktır. İlk atanan ağırlık vektörünün hata yüzeyinin minimumundan uzakta ve eğimsiz bir bölgede ise, sonuca ulaşmak zaman alır. Bu sebeple geriye yayılım algoritmasının yakınsaması genellikle uzun sürer. Geriye yayılım algoritmasının bu eksikliği, algoritmada değişiklikler yapılarak giderilmeye çalışılır. Ancak bu algoritma, eğitimde yavaş olmasına rağmen, eğitim kümesinden farklı bir girişin ağa uygulanması durumunda oldukça doğru sonuçlar verir.

Darbe atlama geriliminin bulunması problemi için yapay sinir ağı kullanılması gerçekten oldukça etkin ve kesin sonuçlar vermektedir. Bu problemde yapay sinir kullanılmasının üstünlüğü, sadece sınırlı sayıda deney yapılarak genel bir çözüme ulaşmaktır. Böyle bir problem için deneysel yöntemleri kullanmak, pahalı olmasının yanında, çok fazla zaman almaktadır. Yapay sinir ağlarının kullanılması ise 13 örüntüsü bulunan bu yapıda, günümüz bilgisayarlarında yaklaşık 150-200 saniyede sonuç vererek, deneysel yöntemlerin zaman alıcılığını ve pahalılığını giderir.

6 YAPAY SİNİR AĞI İLE ELEKTROT BİÇİM OPTİMİZASYONU

6.1 Giriş

Elektrot ve izolatörlerin yalıtımının ekonomik ve etkin olması için, yüzeydeki alan dağılımının düzgün olması gerekmektedir. Yüksek gerilim düzenlerinde kullanılan elektrot ve izolatörlerin şekli, yüzeydeki elektrik alanını etkileyen en önemli faktördür. Yüzeyde düzgün alan dağılımını sağlayacak en iyi elektrot ve izolatör biçimini bulabilmek için çeşitli yöntemler ve algoritmalar geliştirilmiştir.

H. Singer'in 1979'da yayımlanan makalesinde, optimum elektrot geometrisi sayısal bir yöntemle belirlenir [21]. Yöntemin temeli, yük benzetim yönteminin kullanımına dayanmaktadır. Elektrot biçimi, elektrotun yüzey üzerindeki, önceden hesaplanan alan dağılımı veya elektrot üzerindeki verilen bir başlangıç gerilimi, sağlanacak şekilde, iterasyonla belirlenir. 1988'de J. Liu ve J. Sheng, bu yöntemi biraz daha geliştirerek, optimizasyon yüklerinin ilk değerlerinin nasıl hesaplanacağını göstermiş ve önceki yönteme göre, yakınsaması daha hızlı bir algoritma açıklamışlardır [22]. Bu yöntemde, optimizasyon yüklerinin değerleri ve koordinatları, optimizasyon değişkeni olarak kullanılır.

Simetri ekseninde doğrusal gerilim dağılımı sağlanması da, elektrot biçiminin optimum olduğunun bir göstergesidir [23]. Bu yöntemle yapılan optimizasyonlar, sadece elektrot değil her türlü yüksek gerilim elemanına uygulanabilir. Yüksek gerilim elemanlarının, üç boyutlu optimizasyonu ile ilgili çalışmalar da yapılmıştır [24].

Düzgün alan dağılımı elde etmek için uygulanan algoritmalar, bilgisayar yazılımlarındaki ilerlemelerle gelişmiştir. 1995'te K. Kato ve arkadaşları tarafından açıklanan bir yöntemde (CADEF - Computer Aided Design With Electric Field Analysis), optimum elektrot biçimini bulmak için, önce optimum elektrik alan dağılımı hesaplanır [25]. Bu algoritmada, optimum elektrik alanı sağlayacak istenen elektrot biçimi, nicel olarak hesaplanır.

Gaz yalıtımlı enerji elemanlarında, en yüksek delinme gerilimine dayanan optimum elektrot biçimi, yalıtım performansını arttıracak şekilde bulunabilir [26]. Böylece elektrot sisteminde hem düzgün alan saplanır, hem de maksimum alan mümkün olduğu kadar küçük tutulur.

Optimum elektrot biçimini bulmak için, alan dışında başka etkenlerin de kullanıldığı çalışmalar da yapılmıştır [27-28, 30].

İstenen alan dağılımını elde etmek için, elektrot biçiminin iterasyonla değiştirildiği yöntemler [21-30], pek çok problem için etkin değildir. Her iterasyonda alan hesabı yapılması gerektiği için, hesaplama zamanı oldukça uzundur. Bu yöntemler zaman alıcılığının yanında, her zaman genel bir çözüm sunamazlar.

Bu çalışmada, böyle bir optimizasyon problemi, çok katmanlı ileri beslemeli bir yapay sinir ağı algoritması kullanılarak çözülecektir [32-34]. Yapay sinir ağı, elektrot yüzeyinde istenen alan dağılımını sağlayacak bir kenar profili elde etmek için eğitilir. Öğrenme işleminde çeşitli ağ parametrelerinin etkileri ve elektrot yüzeyindeki gerçek alan dağılımı ile beklenen alan dağılımı arasındaki fark incelenir.

Elektrot ve izolatör biçim optimizasyonu, elektrotun çevresi veya izolatörün yüzeyi boyunca düzgün bir alan dağılımı elde edecek biçimi bulmak ve böylece etkin ve yalıtım maliyeti daha düşük bir yalıtım sistemi tasarlamak amacıyla yapılmaktadır. Optimizasyonun yapay sinir ağı kullanılarak yapılması da, daha kısa sürede oldukça etkin sonuçlar elde edilmesini sağlar.

Bu çalışmada, elektrot sistemlerinin önemli bir elemanı, elektrot, en uygun yalıtım sistemini oluşturacak şekilde biçimlendirilmiştir. Optimizasyon için kullanılan çok katmanlı, ileri beslemeli yapay sinir ağının eğitiminde, Levenberg – Marquardt algoritmaları kullanılmıştır.

6.2 Elektrot Biçim Optimizasyonu

Elektrik alanı içinde bulunan yalıtkan malzemeler, bu alanın etkisiyle zorlanırlar. Elektriksel zorlanmalar, yüzeysel boşalmaya ya da yalıtkanın delinmesine yol açabilir. Yüzeysel boşalma veya delinme riskini en aza indirebilmek ve elektrot sistemlerinden en verimli sonucu alabilmek için, yalıtkan malzemenin her yerde eşit zorlanması ve elektrot yüzeyinde eşit bir elektrik alan dağılımı sağlanmalıdır. Düzgün olmayan alanlarda boşalma kısmi boşalma şeklinde başlayarak, gerilimin

yükseltilmesi durumunda tam delinmeyle sonuçlanır. Düzgün alanda ise, delinme koşulu alanın her noktasında aynı anda gerçekleştiğinden, elektrotlardan biri üzerinden başlayan boşalma olayı, tam delinmeyle sonuçlanır. Bu nedenle düzgün alanda boşalma başlangıç gerilimi ile, yalıtkan maddenin delinme dayanımı eşittir.

Yüksek gerilim düzenlerinde kullanılan elektrot ve izolatörlerin şekli, yüzeydeki elektrik alanını etkileyen en önemli faktördür. Yüzeyde düzgün alan elde edebilmek için, yüksek gerilim düzenindeki elektrotların ve izolatörlerin biçiminin optimize edilmesi gerekir. Optimum elektrot ve izolatör biçiminin kullanılması, düzgün alan sağlamanın yanında, elektrik alanının değerinin, kabul edilebilir sınırlar içinde tutulmasını da sağlar. Bu da, sistem ömrünün uzamasına ve yüksek gerilim araçlarının performansının artmasına yardımcı olur.

Yüksek gerilim düzenlerinde sıkça kullanılan çubuk – düzlem elektrot sisteminde alan dağılımı düzgün değildir. Bu çalışmada, bir yapay sinir ağı eğitilerek, çubuk elektrotun optimum biçimi bulunmuş, ve böylece çubuk elektrotun yüzeyinde düzgün alan elde edilmiştir.

6.3 Uygulamada Kullanılan Yapay Sinir Ağ Yapısı

Çubuk – düzlem elektrot sistemde, çubuk elektrotun yapay sinir ağı ile biçim optimizasyonu probleminde, ileri beslemeli, çok katmanlı, eğiticili öğrenmeli bir ağ kullanılmıştır.

Problem doğrusal ayrıştırılabilir bir yapıda olmadığı için, çok katmanlı bir ağ tasarlanmıştır. Tek gizli katmana sahip olan ağ için, eğiticili öğrenmeli bir öğrenme algoritması seçilmiştir.

Eğiticili öğrenmede, ortam dışından bir eğitici, bilinen çıkış örüntülerine göre, her çıkış nöronu için bir hata işareti üretir. Ağırlıklar da, ayrıntılı hata bilgileri ağa uygulanacak şekilde yerleştirilir. Bu öğrenme işleminin performansı, eğitim kümesi kullanılarak istenen sonuca ulaşılabilmesi ve eğitilen ağın genelleştirilebilmesiyle ölçülür.

Elektrot biçim optimizasyonu probleminde, eğitim kümesi bilgisi olarak optimizasyon yapılacak bölgedeki koordinatlar ve karşı düşen elektrik alan değerleri kullanılır. Problemde giriş – çıkış eğitim kümesi bilindiği için, eğiticili öğrenmeli bir yapay sinir ağı kullanılması uygundur [5-7].

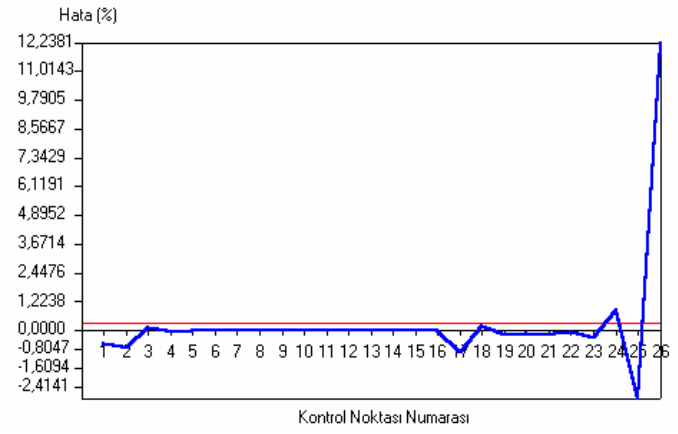
Öğrenme algoritması olarak, geriye yayılım algoritmasının hızlı eğitim yapan bir türü, Levenberg – Marquardt algoritması kullanılmıştır. Levenberg – Marquardt algoritması, standart geriye yayılım algoritmasından çok daha çabuk yakınsar. İleri beslemeli ağların eğitimi için, yakınsamada en hızlı algoritma sayılabilir. Bu algoritma, ilk koşullara ve eğitim parametrelerinin değerlerine çok duyarlı değildir. Bu sebeple bu çalışmada, eğitim parametrelerinin bulunması için, ayrıntılı bir araştırma yapılmayacaktır.

6.4 Eğitim Kümesinin Giriş – Çıkış Örüntülerinin Elde Edilmesi

Yüksek gerilim altında bulunan elektrotların biçim optimizasyonu problemi, elektrik alan hesaplamalarına dayanır. Elektrot yüzeyinde elektrik alanlar, sonlu elemanlar yöntemi, yük benzetim yöntemi gibi, sayısal yöntemlerle hesaplanırlar. Bu çalışmada optimizasyonu yapılacak elektrot yüzeyinin elektrik alan hesaplamaları, yük benzetim yöntemi (Charge Simulation Method) kullanılmıştır. Bu yöntem Ek A'da açıklanmıştır.

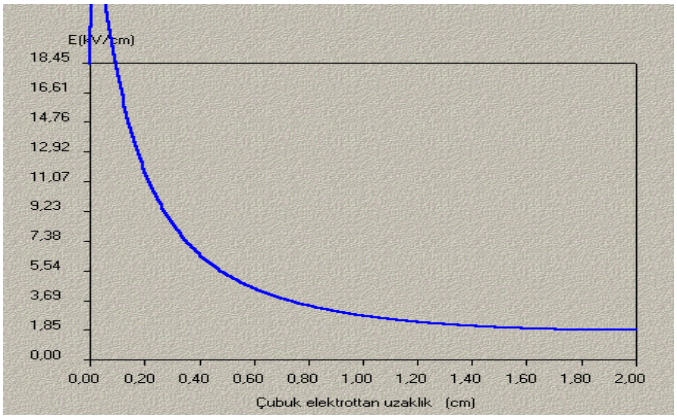
6.4.1 Eğitim Kümesi

Eksenel simetriye sahip, çubuk – düzlem elektrot düzeninde, biçim optimizasyonunun yapay sinir ağı ile yapılması, daha çabuk ve oldukça doğru sonuçlar vermektedir. Yapay sinir ağı ile optimizasyonun ilk aşamasında, eğitim için giriş – çıkış örüntüleri sağlanır. Elektrik alanların hesaplanmasında kullanılan yük benzetim yönteminde, çubuk elektrotun uç noktasına bir noktasal yük, küresel bölgede halkasal yükler ve silindirsel bölgede silindirsel yükler kullanılmıştır. Benzetimin hatası, şekil (6.1)'de gösterilmiştir.



Şekil 6-1 Yük benzetim yöntemiyle hesaplanan potansiyel hatası

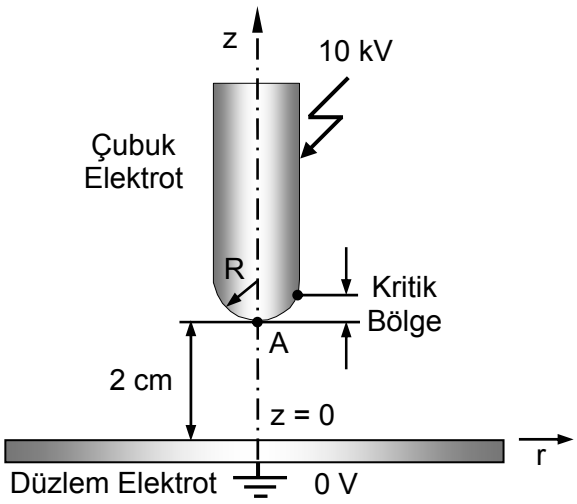
Şekil (6.2)'de benzetimin çubuk elektrot üzerinde oluşturduğu alan dağılımlarına bir örnek verilmiştir.



Şekil 6-2 Elektrik alanın uzaklığa göre değişimi

Eğitim kümesinin, doğruluğunun kabul edilebilir sınırlar içinde olması, yapay sinir ağının uygun sonuçlar vermesini sağlar.

Problemin çözüm amacı, belirli alan değerini sağlayacak olan elektrot biçimini elde etmektir. Bu amaçla uygulanacak yapay sinir ağının giriş örüntüleri elektrik alan değerleri, çıkış örüntüleri ise, elektrotun koordinatlarıdır.



Şekil 6-3 Çubuk – Düzlem Elektrot Sistemi

Eğitim aşamasında, elektrotun optimizasyonu yapılacak alt yüzeyi, küresel olarak düşünülmüştür. Boyu 10 cm olan çubuk elektrot ve düzlem elektrot arasındaki açıklık 2 cm ve elektrotlar arasındaki potansiyel farkı 10 kV olarak alınmıştır.

Alan hesaplamalarına bağlı olarak, farklı eğitim örnekleri elde etmek için, çubuk elektrotun alt yüzeyinin yarıçapına 10 farklı değer verilmiştir. Böylece 10 farklı elektrot sistemi elde edilmiştir. Bu elektrot sistemlerinin hepsi için elektrik alanları, çubuk elektrotun küresel yüzeyindeki 7 farklı noktada hesaplanmıştır. Bu noktalardan biri, çubuk elektrotun uç noktasında (Şekil (6.3)'te A noktası), diğer altısı ise, küresel yüzeyin kritik bölgesinde alınmıştır. Sistemin simetrisinden dolayı hesaplamalar sadece elektrotun bir yarısı için yapılmıştır. Eğitim kümesinin örüntüleri, bu alan hesaplamalarının sonucundan elde edilmiştir.

Eksenel simetriye sahip elektrot sistemi, silindirsel koordinatlarda (θ, r, z) koordinatları ile belirlenir. Elektrot sistemi simetrik olduğu için, θ her durumda sabittir. Alan hesaplamaları, 10 farklı elektrot sistemi için aynı z koordinatlarında yapılır. Bulunacak optimum biçime sahip elektrot da, kritik bölgede aynı z koordinatlarına sahip olacağından, elektrotun koordinatlarını belirlemek için z koordinatlarına ihtiyaç yoktur. Bu durumda optimum biçimi bulunacak olan elektrot, sadece r koordinatı ile belirlenir. Çubuk elektrotun uç noktasının koordinatları, her elektrot sistemi için r = 0, z = 20 mm alınmıştır. Kritik bölgede bulunan diğer 6 noktanın z koordinatları 20,002; 20,012; 20,109; 20,293; 20,546; 20,844 mm'dir. Bu 7 nokta için ölçülen elektrik alanları, 7 boyutlu giriş vektörleri olarak eğitim için kullanılır. Böylece, Tablo (6.1)'de görülen 7 boyutlu 10 giriş örüntüsü elde edilmiş olur.

Tablo 6-1 Çubuk – Düzlem Elektrot Sistemi İçin Giriş Örüntü Vektörleri

Örüntü numarası	Elektrot yarıçapı [mm]	Elektrik alan değerleri [kV/cm]						
		Noktaların silindirsel koordinatları (z, r) [mm]						
		z = 20 r = 0	z = 20,002	z = 20,012	z = 20,109	z = 20,293	z = 20,546	z = 20,844
1	1	82,7200	73,7078	69,1567	67,4438	64,8832	61,6029	57,0954
2	3	18,4550	28,3780	30,5665	30,9938	30,7906	29,8017	28,8979
3	5	35,7250	56,6942	47,5656	24,7823	21,6146	20,5484	19,9858
4	7	16,5350	15,6948	14,8061	16,4203	16,2711	16,0771	15,8690
5	9	12,9110	12,7880	12,8228	13,8403	13,7754	13,5705	13,4164
6	11	11,5610	11,4856	11,3028	12,0945	12,1761	11,9911	11,8726
7	13	10,4610	10,4060	10,2476	11,0027	11,0197	10,8723	10,7774
8	15	8,4280	10,3354	8,9749	10,4879	10,1991	10,0694	9,9131
9	17	11,1850	10,2946	9,1885	8,9817	9,4331	9,4276	9,3151
10	19	9,8030	9,3311	8,7711	8,6099	8,9432	8,9268	8,8174

Hesaplanan elektrik alanlarının koordinatları, biri çubuk elektrotun uç noktasında (A noktası), diğer altısı kritik bölgede bulunmaktadır. A noktasının r koordinatları her

durum için 0 olacağından, çıkış örüntüleri içinde bulunmaz. Kritik bölgede bulunan 6 noktanın silindirsel koordinatlardaki r koordinatları ise, Tablo (6.2)'de görüldüğü gibi, yapay sinir ağının çıkış vektörünü oluşturur.

Tablo 6-2 Çubuk – düzlem elektrot sistemi için çıkış örüntü vektörleri

Örüntü numarası	Elektrot yarıçapı [mm]	Kritik bölgedeki noktaların r – ordinatları [mm]					
		Noktaların z - koordinatları [mm]					
		z = 20,002	z = 20,012	z = 20,109	z = 20,293	z = 20,546	z = 20,844
1	1	0,0654	0,1564	0,1540	0,7071	0,8910	0,9877
2	3	0,1570	0,1962	0,7775	1,0751	1,6339	2,1213
3	5	0,1581	0,3280	1,0800	1,6937	2,2700	2,8104
4	7	0,1583	0,4578	1,4010	2,0132	2,6061	3,1779
5	9	0,1585	0,4710	1,4079	2,3294	3,2253	3,8036
6	11	0,1587	0,4721	1,4180	2,3382	3,5482	4,1368
7	13	0,1588	0,4821	1,7020	2,6516	3,8675	4,4627
8	15	0,9810	0,4812	1,7251	2,9649	3,8823	5,0811
9	17	0,4650	0,7841	2,0310	2,9693	4,1973	5,4025
10	19	0,4712	0,7752	2,0381	3,2821	4,5118	5,7217

Tablo (6.1) ve Tablo (6.2)'de görüldüğü gibi, yapay sinir ağı, 7 giriş ve 6 çıkış nöronundan oluşmaktadır. Giriş – çıkış vektörleri, eğitimi uygun yaparsa, çok katmanlı yapay sinir ağı, kritik bölgede düzgün dağılımlı elektrik alan elde edecek en iyi elektrot biçimini belirler.

Eğitimin tamamlanmasından sonra, test kümesi oluşturulur ve gerçek çıkış ile yapay sinir ağının hesaplamış olduğu çıkış karşılaştırılarak, ortalama mutlak hatası bulunur.

Eğitim ve test bittikten sonra, kritik bölgede istenen alan değerine karşı, çubuk elektrotun koordinatları belirlenir [32-33].

6.5 Algoritmanın Uygulanması

Önceki bölümde açıklandığı gibi, yapay sinir ağı, 7 giriş nöronu ve 6 çıkış nöronundan oluşmaktadır. Tek gizli katmandan oluşan yapıda, eğitimde kullanılan Marquardt parametresinin ilk değeri, μ_i = 0,001; artma faktörü μ^+ = 10; azalma faktörü μ^- = 0,1 olarak alındığında, ağın oldukça iyi sonuç verdiği görülmüştür. Levenberg – Marquardt algoritmasında ağın performansı, bu parametrelere çok fazla duyarlı değildir. Bu sebeple, parametreler için ayrıntılı bir araştırma yapılmamıştır.

1000 iterasyon için yapılan çalışmalarda, gizli katmandaki nöron sayısı 7'den 22'ye kadar arttırılarak incelenmiş ve en iyi sonucun 20 nöronda meydana geldiği gözlenmiştir.

Eğitimin tamamlanmasından sonra, ağın güvenilirliğini sınamak amacıyla, eğitim kümesinden bir örüntü kullanılarak bir test kümesi oluşturulmuş ve gerçek çıkış ile yapay sinir ağının hesaplamış olduğu çıkış karşılaştırılarak, hata hesaplanmıştır. Daha sonra, kritik bölgede istenen alan değerleri 10; 12 ve 24,4 olmak üzere, yapay sinir ağının hesapladığı ağırlıklar yardımıyla, bu alanları sağlayan çubuk elektrotun koordinatları bulunmuştur. Bu çalışmalar, Matlab 6.5 Neural Network Toolbox kullanılarak gerçekleştirilmiştir [Ek B2].

Bu çalışmada eğitim için, performans fonksiyonu olarak ortalama karesel hata (mean squared error, mse), test için ise ortalama mutlak hata (mean absolute error, mae) kullanılmıştır (denklem (6.1) ve (6.2)).

$$\text{mse} = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{2}\sum_{j\in C} e_j^2(n) \tag{6.1}$$

$$\text{mae} = \frac{1}{n_t\, n_o}\sum_{j=1}^{n_t}\sum_{k=1}^{n_o}\left[\frac{\left|d_{jk} - y_{jk}\right|}{d_{jk}}\right] 100 \tag{6.2}$$

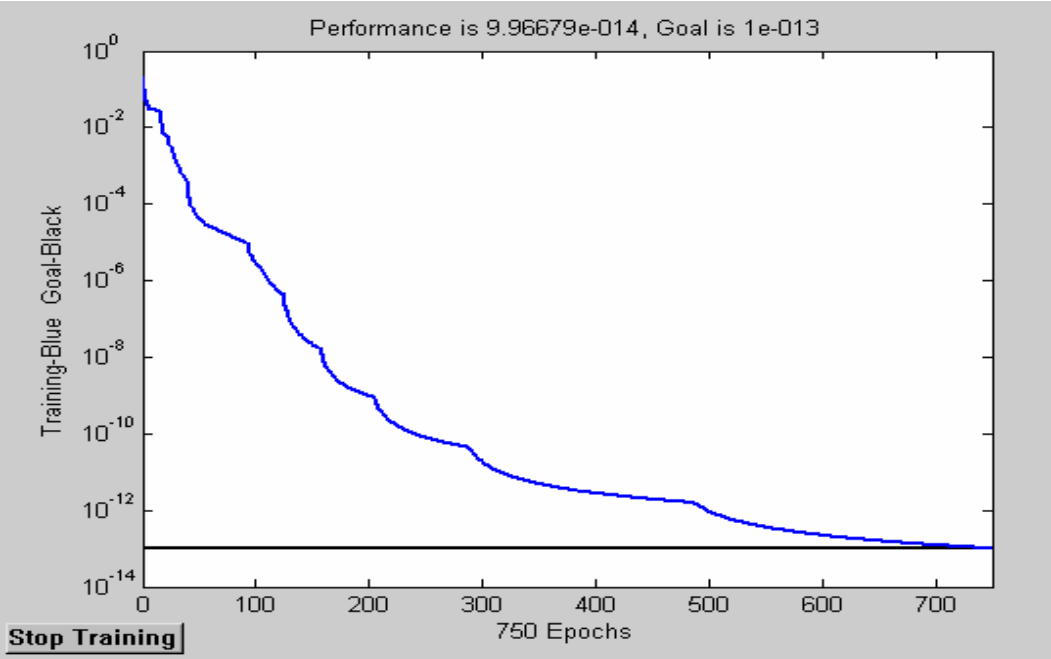
Burada n_t test kümesindeki örüntü sayısını, n_o çıkış katmanındaki nöron sayısını, N eğitim kümesindeki toplam örüntü sayısını, e_j hata işaretini, d_{jk} istenen değeri, y_{jk} ağın çıkışını göstermektedir.

Tablo 6-3 YSA sonuçları ile gerçek değerlerin karşılaştırılmasına bir örnek

Gizli katman nöron sayısı = 20; İterasyon sayısı = 750 μ_i = 0,001; μ^+ = 10; μ^- = 0,1						
r – ordinatları [mm]						
Gerçek değerler	0,1585	0,4710	1,4079	2,3294	3,2253	3,8036
YSA değerleri	0,1585	0,4712	1,4077	2,3288	3,2249	3,8030

En uygun ağ yapısıyla, eğitim, ortalama karesel hata 10^{-13}'den küçük olana kadar sürdürülmüştür (Şekil 6.4). Öngörülen ortalama karesel hataya 750 iterasyonda ulaşmıştır (Tablo (6.3)). Bu denemenin sonuçları Tablo (6.3)'te görülmektedir.

Bu örnek için ortalama karesel hata, mse = $9{,}96 \cdot 10^{-14}$ ve ortalama mutlak hata, mae = $1.4759 \cdot 10^{-4}$ olarak bulunmuştur (Şekil 6.4).



Şekil 6-4 Ortalama karesel hatanın iterasyon sayısına göre değişimi

Kullanılan ağ yapısının güvenilirliği, test kümesiyle gözlendikten sonra, istenen alan dağılımı için optimum elektrot biçimi, yapay sinir ağının hesapladığı ağırlıklar yardımıyla bulunur. Elde edilen yapay sinir ağı çıkışları (Tablo 6.4)'te verilmiştir.

Tablo 6-4 Kritik bölgenin koordinatları

Elektrik alan değerleri [kV/cm]	Kritik bölgedeki noktaların r – ordinatları [mm]					
	Noktaların z - koordinatları [mm]					
	z = 20,002	z = 20,012	z = 20,109	z = 20,293	z = 20,546	z = 20,844
10	0,0192	0,1389	1,1170	2,1817	3,3526	3,7427
12	0,0263	0,2542	0,4146	0,9929	2,2809	2,4718
24,4	0,0654	0,1564	0,7071	0,8910	0,9877	1,1540

6.6 Sonuç

Bu çalışmada, bir çubuk – düzlem elektrot sistemi için, Levenberg – Marquardt öğrenme algoritmasını kullanan, çok katmanlı, ileri beslemeli bir yapay sinir ağı tasarlanarak, çubuk elektrotun biçimi optimize edilmiştir. Yapay sinir ağında eğitim için Levenberg – Marquardt algoritması kullanmak, gradyen azalması ve Gauss – Newton yöntemlerine göre daha etkili sonuçlar vermektedir. Her iterasyonda

performans fonksiyonu azaltıldığı için, minimum hataya daha az iterasyonla ulaşılır. Bu algoritma, Hessian matrisinin yaklaşık bir değerini kullandığı için, diğer minimizasyon yöntemlerini kullanan algoritmalara göre, hesaplamalardaki karmaşıklığı da azaltır. Levenberg – Marquardt öğrenme algoritması, çok çabuk yakınsayan ve minimumlara takılmayan, pek çok uygulama için oldukça etkin bir algoritmadır.

Optimizasyonu yapılan kritik bölgede, ortalama mutlak hatanın, her durumda % 0,1'den daha küçük olduğu gözlenmiştir. Bu problemde yapay sinir ağı kullanılmasının üstünlüğü, az sayıda alan hesaplaması yapılarak, iteratif sürece girmeden, biçim optimizasyonunun yapılabilmesidir. Yapay sinir ağı kullanımı, iteratif yöntemlerin zaman alıcılığını büyük ölçüde gidermesinin yanında, daha genel çözümler elde etmeye de yardımcı olur.

Bu çalışmadan görüldüğü gibi, biçim optimizasyonu problemi için YSA kullanılması, çok kısa sürede istenen sonuçları verir. Böylece herhangi bir yalıtım sisteminin tasarlanması, daha ekonomik ve daha kolay hale gelir.

7 YAPAY SİNİR AĞI İLE İZOLATÖR KONUM AÇISI OPTİMİZASYONU

7.1 Giriş

Bölüm (6.1)'de anlatıldığı gibi, yüksek gerilim sistemlerinde, yalıtımın ekonomik ve etkin olması için, yüzeyde elektrik alanının düzgün dağılımlı olması gerekmektedir. Düzgün alan dağılımı, yalıtım malzemesinin her yerde eşit zorlanmasını sağlar. Böylece yalıtım sisteminde, yüzeysel boşalmalar ve delinmeler en aza inip, yalıtım sisteminden en verimli sonuç alınmış olur. Elektrik alanın dağılımını etkileyen en önemli etken, elektrot ve izolatörün şeklidir. Elektrot ve izolatörlerin optimum biçimde olması, yalıtım sisteminin alan dağılımının düzgün olmasını sağlar. Özellikle çok yüksek gerilim sistemlerinde, yalıtımın pahalı olmasından dolayı, optimum elektrot ve izolatör biçiminin kullanılması tercih edilir [21-42].

Optimum izolatör biçiminin elde edilebilmesi için, yalıtım sistemine bazı optimizasyon teknikleri uygulanması gerekir. İzolatör biçim optimizasyonu için farklı optimizasyon teknikleri kullanılan çeşitli çalışmalar yapılmıştır [35-42]. Bu çalışmaların hepsinde, izolatör biçimi adım adım değiştirilip, düzeltilerek belirlenir. Son yıllarda, izolatör biçim optimizasyonu problemlerinde, yapay sinir ağı da kullanılmaktadır [43-44].

İzolatörün optimum biçimi bulunurken, izolatör yüzeyinde düzgün alan dağılımı elde edilmeye çalışılır. Böylece etkin ve yalıtım maliyeti daha düşük bir yalıtım sistemi tasarlanmış olur. İzolatörün yüzeyinde birkaç noktada hesaplanan elektrik alanların teğetsel bileşenleri yapay sinir ağına giriş olarak, noktaların koordinatları da çıkış olarak yapay sinir ağına uygulanır. Bu şekilde eğitilen bir yapay sinir ağı için, eğitim bittikten sonra istenen düzgün dağılımlı elektrik alanını verecek izolatörün optimum biçimi bulunur [43].

Böyle bir problemin yapay sinir ağı ile çözülmesi, her adımda yapılması gereken hesaplamaların zaman alıcılığını giderir. Sınırlı sayıda yapılan hesaplamalar, yapay sinir ağını eğitmek için gerekli eğitim kümesinin elde edilmesine yeterlidir. Yapay

sinir ağı bir kere eğitildikten sonra, istenen giriş değerleri ağa uygulanarak sonuca ulaşılır. Böylece az sayıda hesaplama yaparak, etkin ve genel bir çözüm elde edilir.

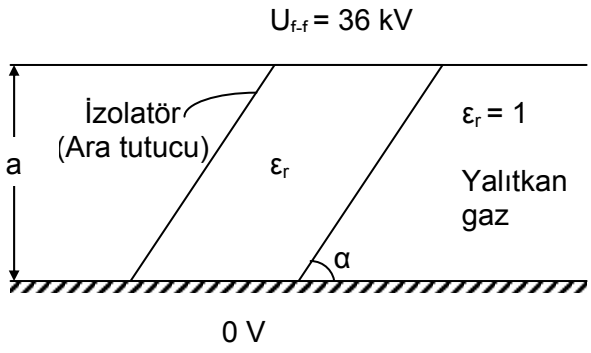
Bu çalışmada, özellikle basınçlı gaz yalıtımlı sistemlerde, yüksek gerilim iletkenlerini taşıyan, ara tutucu olarak kullanılan doğrusal eğimli izolatörlerin, metal mahfaza ve yüksek gerilim iletkeni arasına yerleştirilme açıları yapay sinir ağı ile belirlenecektir [45-47]. Bu amaçla kullanılan, çok katmanlı, ileri beslemeli yapay sinir ağı geriye yayılım algoritması ile eğitilmiştir.

7.2 Gaz Yalıtımlı Sistemlerde Ara Tutucular

Yerleşim sınırlamaları, kirlenme, çevre koşulları ve güvenlik gibi etkenler nedeniyle gaz yalıtımlı sistemlerin kullanımı yaygınlaşmaktadır. Basınçlı gaz yalıtımlı bu sistemlerde, metal mahfazaları ile yüksek gerilim iletkenleri arasındaki açıklığın sağlanması ve iletkenlerin mekanik olarak tutturulması amacıyla ara tutucu (spacer) adı verilen (yalıtkan) mesnet izolatörlerinden (Şekil (7.1)) yararlanılmaktadır.

Ara tutucuların kullanılması, gaz yalıtımlı sistemlerin elektriksel, mekanik ve ısıl özelliklerini dolayısıyla güvenilirliklerini etkiler. Ara tutucu – gaz ara yüzeyinde teğetsel alan şiddetinin değeri, ara tutucunun elektrot açıklığını köprülemesi nedeniyle basınçlı gaz yalıtımlı sistemin delinme dayanımına sınırlama getirir. Ara tutucu olarak kullanılan izolatörün biçimi, boyutları, yüzey durumu, yerleşimi, malzemesi, kirlenmesi ve iletken parçacıklar gaz yalıtımlı sistem içindeki elektriksel alan dağılımını değiştirir.

Ara tutucuların gerek yüzeylerinde, gerekse bulundukları ortam içinde elektrik alan dağılımındaki etkileri göz önünde bulundurulmalıdır. Bu amaçla deneysel ve kuramsal çalışmaları yapılmaktadır [47].



Şekil 7-1 Gaz yalıtımlı sistem içindeki ara tutucu olarak kullanılan izolatör

Bu çalışmada, elektrot açıklığı ve izolatörün gaz yalıtkan tarafındaki yüzeyinin elektrik alan değerlerine göre, izolatörün konum (yerleştirilme) açısı yapay sinir ağı ile belirlenecektir. Ara tutucu olarak porselen, cam ve epoksi reçineden yapılmış üç farklı izolatör kullanılmıştır.

7.3 Uygulamada Kullanılan Yapay Sinir Ağı

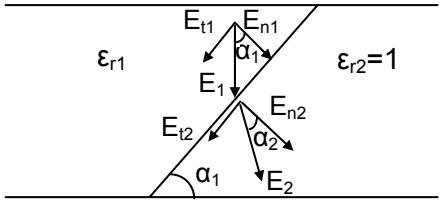
Gaz yalıtımlı sistemlerde ara tutucu olarak kullanılan izolatörlerin, yerleşim açılarının bulunması probleminde, bir çok katmanlı ileri beslemeli yapay sinir ağı kullanılmıştır.

Problem doğrusal ayrıştırılabilir bir yapıda olmadığı için, çok katmanlı bir ağ tasarlanmıştır. Giriş çıkış verileri bilindiği için problemin çözümünde eğiticili bir öğrenme algoritması seçilmiştir. Eğiticili öğrenmede, ortam dışından bir eğitici, bilinen çıkış örüntülerine göre, her çıkış nöronu için bir hata işareti üretir. Ağırlıklar da, ayrıntılı hata bilgileri ağa uygulanacak şekilde yerleştirilir. Bu öğrenme işleminin performansı, eğitim kümesi kullanılarak istenen sonuca ulaşılabilmesi ve eğitilen ağın genelleştirilebilmesiyle ölçülür.

Öğrenme algoritması olarak, yapay sinir ağlarında çok kullanılan geriye yayılım (back-propagation) algoritması kullanılmıştır. Geriye yayılım algoritması, ilk değerlere ve eğitim parametrelerine çok duyarlıdır. Bu sebeple, parametrelerin bulunması için ayrıntılı araştırmalar yapılmıştır [5-14].

7.4 Eğitim Kümesinin Giriş – Çıkış Örüntülerinin Elde Edilmesi

Şekil (7.1)'de görülen gaz yalıtımlı sistemlerde ara tutucu olarak kullanılan mesnet izolatörü, doğrusal eğimlidir. Elektrotlar arasındaki açıklık ve üst elektrota uygulanan gerilim değerleri belirlidir. Bu değerler kullanılarak oluşan elektrik alan hesaplanabilir.



Şekil 7-2 İzolatör yüzeyindeki elektriksel alanlar ve bileşenleri

Elektriksel alanlar, şekil (7.2)'de görüldüğü gibidir. Dielektrik katsayıları farklı olan izotrop, homojen iki yalıtkan ortamı ayıran sınır yüzeyde (izolatör yüzeyi) $\vec{E}$ ve $\vec{D}$ alan çizgileri kırılırlar. Kırılma olayı, kırılma açıları ile dielektrik katsayıları arasında belirli bir bağıntıya göre olur.

Şekil (7.2)'de görüldüğü gibi, dielektrik katsayısı ε_{r1}=1 olan gaz yalıtkan ve dielektrik katsayısı ε_{r2} olan izolatörü ayıran yüzeyde (izolatör yüzeyinde), $\vec{E}_1$ ve $\vec{E}_2$ elektrik alanı vektörlerinin sınır yüzeye dik normalle yaptıkları açılar α_1 ve α_2'dir. İzolatörün doğrusal eğimli olmasından ve elektrotların paralel düzende yerleşmesinden dolayı, α_1 aynı zamanda izolatörün yerleşim açısıdır.

Sınır yüzeyde, elektrik alanların teğetsel bileşenleri birbirlerine eşittir.

$$E_{t1} = E_{t2} \tag{7.1}$$

$$E_1 \sin\alpha_1 = E_2 \sin\alpha_2 \tag{7.2}$$

Ayrıca sınır yüzeyde deplasman vektörlerinin normal bileşenleri de eşittir.

$$D_{n1} = D_{n2} \tag{7.3}$$

$$D_1 \cos\alpha_1 = D_2 \cos\alpha_2 \tag{7.4}$$

$\vec{D}_1 = \varepsilon_1 \vec{E}_1$ ve $\vec{D}_2 = \varepsilon_2 \vec{E}_2$ olduğu göz önünde bulundurularak, denklem (7.4)'ten denklem (7.5) elde edilir.

$$\varepsilon_{r1} E_1 \cos\alpha_1 = \varepsilon_{r2} E_2 \cos\alpha_2 \tag{7.5}$$

Denklem (7.2) ve (7.5) birbirine oranlanırsa,

$$\varepsilon_{r1} \tan\alpha_1 = \varepsilon_{r2} \tan\alpha_2 \tag{7.6}$$

bağıntısı bulunur. Bu çalışmada, ikinci ortam gaz yalıtkan olduğu için dielektrik katsayısı 1'e eşittir.

$$\alpha_2 = \arctan(\varepsilon_{r1} \tan\alpha_1) \tag{7.7}$$

Bu çalışmada, üst elektrota faz – faz arası Um = 36 kV gerilim uygulanmaktadır. Ara tutucu olarak kullanılan izolatör, faz – toprak arasına yerleştirildiği için, izolatöre uygulanan gerilim $U = \frac{Um}{\sqrt{3}} = \frac{36}{\sqrt{3}}$ olarak alınır. Düzlemsel elektrotlarda, elektrik alan E = U/a olduğu için, yalıtkan ortamlardaki elektrik alan değerleri aşağıdaki gibi hesaplanır:

$$E_1 = \frac{U}{a}$$
$$E_2 = E_1 \frac{\sin\alpha_1}{\sin\alpha_2} \qquad (7.8)$$

7.4.1 Eğitim Kümeleri

Bu çalışmada, ara tutucu olarak kullanılan, doğrusal biçimli epoksi reçine, cam ve porselenden yapılmış üç farklı izolatör kullanılmıştır. Her izolatör durumu için, dielektrik katsayıları göz önünde bulundurularak, alan hesaplamaları yapılmıştır. Hesaplamalar bölüm (7.4)'te anlatıldığı gibi yapılarak eğitim kümeleri elde edilir. Elektrot açıklığı, her izolatör durumu için, izolatörün yerleşim açısı sabit olmak koşulu ile, 25 ve 35 cm arasında 2,5 cm aralıkla arttırılarak belirlenmiştir. Daha sonra, belirlenen elektrot açıklıkları için izolatörün yerleşim açısı 30º, 45º, ve 60º alınarak eğitim kümesi oluşturulur. Belirlenen elektrot açıklığı, izolatörün dielektrik katsayısı ve yerleşim açısı kullanılarak, gaz yalıtkandaki elektrik alan değerleri hesaplanır.

Bu çalışmada, elektrot açıklığı (a) ve gaz yalıtkandaki elektrik alan (E_2) değerleri giriş vektörünü, izolatörün yerleşim açısı (α_1) çıkış vektörünü oluşturmaktadır.

Eğitimin tamamlanmasında sonra, her izolatör için, eğitim kümesinin dışından bir test kümesi oluşturularak, eğitim ve test hataları hesaplanır. Eğitim ve test bittikten sonra, yapay sinir ağı istenen elektrot açıklığı ve gaz yalıtkandaki elektrik alanı değerlerine en uygun yerleşim açısını verir.

Tablo 7-1 Epoksi reçine izolatör için eğitim kümesi

Dielektrik katsayısı ε_{r1}	Örüntü numarası	Giriş vektörü		Çıkış vektörü
		Elektrot açıklığı, a (cm)	Gaz yalıtkandaki elektik alan (kV/cm)	İzolatörün yerleşim açısı, α_1 (º)
3	1	25,0	0,4800	30
3	2	27,5	0,4364	30
3	3	30,0	0,4000	30
3	4	32,5	0,3692	30
3	5	35,0	0,3429	30
3	6	25,0	0,6197	45
3	7	27,5	0,5633	45
3	8	30,0	0,5164	45
3	9	32,5	0,4767	45
3	10	35,0	0,4426	45
3	11	25,0	0,7332	60
3	12	27,5	0,6666	60
3	13	30,0	0,6110	60
3	14	32,5	0,5640	60
3	15	35,0	0,5237	60

Tablo 7-2 Cam izolatör için eğitim kümesi

Dielektrik katsayısı ε_{r1}	Örüntü numarası	Giriş vektörü		Çıkış vektörü
		Elektrot açıklığı, a (cm)	Gaz yalıtkandaki elektik alan (kV/cm)	İzolatörün yerleşim açısı, α_1 (º)
4	1	25,0	0.4530	30
4	2	27,5	0.4118	30
4	3	30,0	0.3775	30
4	4	32,5	0.3485	30
4	5	35,0	0.3236	30
4	6	25,0	0.6060	45
4	7	27,5	0.5509	45
4	8	30,0	0.5050	45
4	9	32,5	0.4661	45
4	10	35,0	0.4328	45
4	11	25,0	0.7275	60
4	12	27,5	0.6613	60
4	13	30,0	0.6062	60
4	14	32,5	0.5596	60
4	15	35,0	0.5196	60

Tablo 7-3 Porselen izolatörün eğitim kümesi

Dielektrik katsayısı ε_{r1}	Örüntü numarası	Giriş vektörü		Çıkış vektörü
		Elektrot açıklığı, a (cm)	Gaz yalıtkandaki elektik alan (kV/cm)	İzolatörün yerleşim açısı, α_1 (º)
5	1	25,0	0.4399	30
5	2	27,5	0.3999	30
5	3	30,0	0.3666	30
5	4	32,5	0.3384	30
5	5	35,0	0.3142	30
5	6	25,0	0.5995	45
5	7	27,5	0.5450	45
5	8	30,0	0.4996	45
5	9	32,5	0.4612	45
5	10	35,0	0.4282	45
5	11	25,0	0.7248	60
5	12	27,5	0.6589	60
5	13	30,0	0.6040	60
5	14	32,5	0.5575	60
5	15	35,0	0.5177	60

Eğitimin tamamlanmasında sonra, her izolatör için, eğitim kümesinin dışından bir test kümesi oluşturularak, eğitim ve test hataları hesaplanır. Eğitim ve test bittikten sonra, yapay sinir ağı istenen elektrot açıklığı ve gaz yalıtkandaki elektrik alanı değerlerine en uygun yerleşim açısını verir.

7.5 Algoritmanın Uygulanması

Bir önceki bölümde açıklandığı gibi, yapay sinir ağı iki giriş (elektrot açıklığı ve gaz yalıtkandaki elektrik alan) ve bir çıkış (izolatörün yerleşim açısı) nöronundan oluşmaktadır. Tek gizli katman kullanılan yapay sinir ağında, ağırlıkların belirlenmesi için momentum katsayılı geriye yayılım algoritması kullanılmıştır. Bu algoritma, Bölüm (4.2)'de anlatılan dik iniş (steepest descent) yöntemiyle ağırlıklarını hesaplar. Algoritmada aktivasyon fonksiyonu olarak birinci katmanda sigmoid fonksiyonu, ikinci katmanda parçalı lineer fonksiyon kullanılmıştır. Algoritma, Matlab 6.5 Neural Network Toolbox kullanılarak yazılan bir programla uygulanmıştır [Ek B3].

Geriye yayılım algoritmasının eğitim parametrelerinin belirlenmesi için ayrıntılı çalışmalar yapılmıştır (denklem (4.5) – denklem (4.30)).

Parametrelerin belirlenmesi için, deneme – yanılma yönteminden yararlanılmıştır. 1000 iterasyon için yapılan çalışmalarda, önce öğrenme hızı (η) sabit tutulup, momentum katsayısı (α) değiştirilmiş, daha sonra, momentum katsayısı sabit tutulup, öğrenme hızı değiştirilerek en uygun sonuç aranmıştır.

Eğitim kümesi hesaplandıktan sonra, bölüm (5.3.1)'de anlatıldığı gibi normalizasyonu yapılır. Parametrelerin bulunması için denemeler yapılırken, eğitim için hatanın etkin değeri, test için ise ortalama mutlak hata kullanılmıştır.

$$rms = \sqrt{\frac{1}{N\, n_o} \sum_{j=1}^{N} \sum_{k=1}^{n_o} (d_{jk} - y_{jk})^2} \tag{7.9}$$

$$mae = \frac{1}{n_t\, n_o} \sum_{j=1}^{n_t} \sum_{k=1}^{n_o} \left[\frac{\left| d_{jk} - y_{jk} \right|}{d_{jk}} \right] 100 \tag{7.10}$$

Burada N eğitim kümesindeki tüm örüntü sayısını, n_o çıkış katmanındaki nöron sayısını, n_t test kümesindeki örüntü sayısını, d_{jk} istenen çıkışı, y_{jk} ise ağın çıkışını göstermektedir [5-14].

7.5.1 Epoksi Reçine İzolatörün Konum Açısının Yapay Sinir Ağı İle Bulunması

Öğrenme hızını ve momentum katsayısını bulmak için yapılan çalışmalar, 10 nöronlu tek gizli katmanlı bir ağ kullanılarak 1000 iterasyon için yapılmıştır. Ağ için en uygun parametreleri bulmak amacıyla, öğrenme hızı (η) ve momentum katsayısı (α) için denemeler yapılmıştır. Öğrenme hızı η = 0,1'de sabit tutulup, momentum katsayısı (α) 0,1'den 0,9'a kadar değiştirilerek en uygun momentum katsayısı aranmıştır (Tablo (7.4)). Benzer çalışma, momentum katsayısı α = 0,1'de sabit tutulup, öğrenme hızına 0,1 ile 0,9 arasında değerler verilerek en uygun öğrenme hızı aranmış ve sonuçlar Tablo (7.5)'te gösterilmiştir. 1000 iterasyon için yapılan denemelerde, öğrenme hızı ve momentum katsayısı her deneme için sabit tutulmuştur.

Test kümesini oluşturmak için, elektrot açıklığı a = 26 cm iken, izolatörün yerleşim açısı 45º olarak alınarak, ε_{r1} = 3 için elektrik alan değerleri hesaplanmıştır. Bu değerlere göre, test kümesi girişi [26 0.5958] ve çıkışı [45] olarak bulunur.

Tablo 7-4 Momentum katsayısının hatalara etkisi
[rms: hatanın etkin değeri (root mean square error);
mae: ortalama mutlak hata (mean absolute error)]

η = 0,1; gizli katman nöron sayısı=10; iterasyon sayısı=1000		
α	rms	mae
0,9	0,00359	1,8339
0,8	0,0069	5,7497
0,7	0,0035	2,219
0,6	0,0028	4,8334
0,5	0,0015	0,6537
0,4	0,0115	4,6067
0,3	0,0235	9,492
0,2	0,0121	5,9563
0,1	0,0037	7,2558

Tablo 7-5 Öğrenme hızının hatalara etkisi
[rms: hatanın etkin değeri (root mean square error);
mae: ortalama mutlak hata (mean absolute error)]

α = 0,5; gizli katman nöron sayısı=10; iterasyon sayısı=1000		
η	rms	mae
0,1	0,0015	0,6537
0,2	0,0059	0,3063
0,3	0,0045	0,6203
0,4	0,0084	0,2801

Tablo (7.4) ve tablo (7.5)'ten görüldüğü gibi, epoksi reçine izolatör için en uygun öğrenme hızı ve momentum katsayısı, η=0,4 α=0,5 olarak belirlenir.

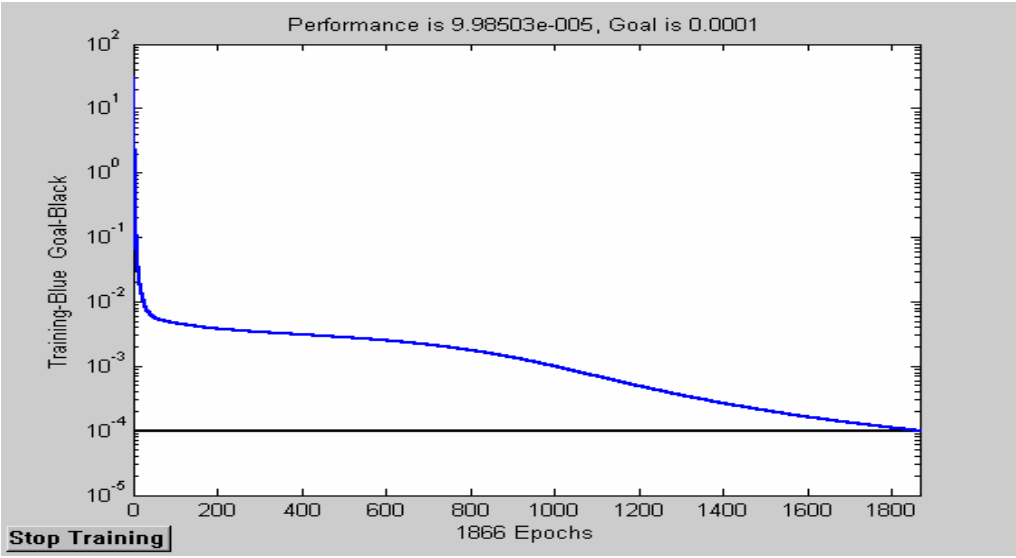
Tablo 7-6 Gizli katmandaki nöron sayısının hatalara etkisi

İterasyon sayısı:1000		
Gizli katman nöron sayısı	rms	mae
15	0,0040	1,3677
14	0,0026	4,1025
13	0,0030	4,8935
12	0,0009	0,5288
11	0,0009	2,7286
10	0,0084	0,2801
9	0,0012	9,7271
8	0,0002	1,4077
7	0,0083	6,4380
6	0,0003	5,2466
5	0,0035	0,5150

En uygun öğrenme hızı ve momentum katsayısı bulunduktan sonra, bu değerler gizli katmandaki nöron sayısını bulabilmek için kullanılmıştır. Öğrenme hızı η = 0,4 ve momentum katsayısı α = 0,5 iken, gizli katmandaki nöron sayısı, 5'ten 15'e kadar

değiştirilerek, en iyi eğitim ve test hatasını veren değer aranmıştır. 1000 iterasyon için yapılan denemelerde, gizli katmandaki nöron sayısının 10 olmasının en uygun sonucu verdiği gözlenmiştir (Tablo (7.6)).

Tablo (7.4), tablo (7.5) ve tablo (7.6)'dan görüldüğü gibi, epoksi reçine izolatörün yerleşim açısının probleminin yapay sinir ağı ile çözülmesi için kullanılacak parametreler η=0,4 α=0,5 ve gizli katmandaki nöron sayısı=10 olarak belirlenir. En uygun parametreler belirlendikten sonra eğitim, toplam karesel hata 0,0001'den küçük olana kadar sürdürülmüştür (Şekil (7.3)).



Şekil 7-3 Epoksi reçine izolatör için toplam karesel hatanın iterasyon sayısına göre değişimi

Şekil (7.3)'ten görüldüğü gibi, yapay sinir ağı öngörülen hataya, 1866 iterasyon sonunda ulaşmıştır. Eğitimin bittiği anda, hatanın etkin değeri rms = 0,0048, ve ortalama mutlak hata % 1,82 olarak bulunmuştur. Test kümesinin ağa uygulanması, ağın güvenilirliğini test etmeye yardımcı olur. Test kümesinin çıkışının gerçek değeri 45º iken, yapay sinir ağı bulduğu çıkış 45,0182'dir.

7.5.2 Cam İzolatörün Konum Açısının Yapay Sinir Ağı İle Bulunması

Cam izolatörün yerleşim açısının yapay sinir ağı ile belirlenmesi için kullanılacak uygun parametreleri bulmak için, bölüm (7.5.1)'de anlatılan çalışmaya benzer bir çalışma gerçekleştirilmiştir.

Test kümesini oluşturmak için, elektrot açıklığı a = 26 cm iken, izolatörün yerleşim açısı 45º olarak alınarak, ε_{r1} = 4 için elektrik alan değerleri hesaplanmıştır. Bu değerlere göre, test kümesi girişi [26 0.5827] ve çıkışı [45] olarak bulunur.

Tablo 7-7 Momentum katsayısının hatalara etkisi

η = 0,1; gizli katman nöron sayısı=10; iterasyon sayısı=1000		
α	rms	mae
0,9	0,0189	1,8181
0,8	0,0015	3,6108
0,7	0,0125	3,2912
0,6	0,0248	1,6418
0,5	0,0014	0,0379
0,4	0,0065	7,3186
0,3	0,0019	1,7935
0,2	0,0181	1,3464
0,1	0,0121	4,0914

Tablo 7-8 Öğrenme hızının hatalara etkisi

α = 0,5; gizli katman nöron sayısı=10; iterasyon sayısı=1000		
η	rms	mae
0,1	0,0014	0,0379
0,2	0,0054	3,2547
0,3	0,004	1,6985
0,4	0,0025	3,5431
0,5	0,0028	1,6233

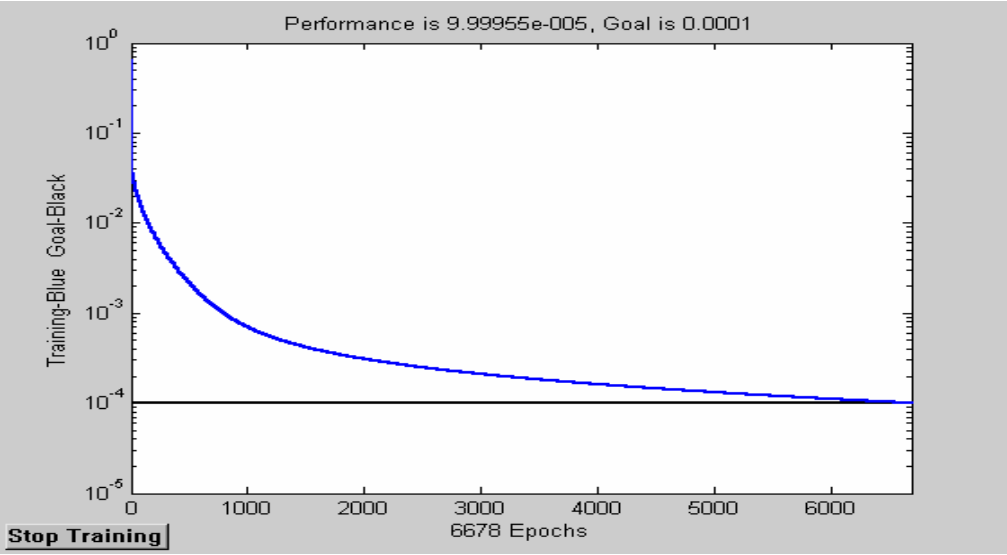
Tablo (7.7) ve tablo (7.8)'den görüldüğü gibi, cam izolatör için en uygun öğrenme hızı ve momentum katsayısı, η=0,1 α=0,5 olarak belirlenir.

Tablo 7-9 Gizli katman nöron sayısının hatalara etkisi

İterasyon sayısı:1000		
Gizli katman nöron sayısı	rms	mae
15	0,0021	2,7314
10	0,0014	0,0379
9	0,0063	5,7290
8	0,0027	0,3714
7	0,0013	0,7233
6	0,0147	2,0466
5	0,0046	4,7861

Bölüm (7.5.1)'de açıklandığı gibi, gizli katmandaki nöron sayısı, en uygun öğrenme hızı ve momentum katsayısı değerleri kullanılarak, deneme – yanılma yöntemi ile bulunur. Bunun için gizli katmandaki nöron sayısına, 5 – 15 arasına değerler verilir ve en küçük hatayı veren gizli katman nöron sayısı, eğitimde kullanılmak üzere seçilir (Tablo (7.9)).

Epoksi reçine izolatör için yapılan çalışmalara benzer şekilde, Tablo (7.7), tablo (7.8) ve tablo (7.9)'daki değerlerden yararlanarak, cam izolatörün yerleşim açısının probleminin yapay sinir ağı ile çözülmesi için kullanılacak parametreler η = 0,4, α = 0,5 ve gizli katmandaki nöron sayısı = 10 olarak belirlenir. En uygun parametreler belirlendikten sonra eğitim, toplam karesel hata 0,0001'den küçük olana kadar sürdürülmüştür (Şekil (7.4)).



Şekil 7-4 Cam izolatör için toplam karesel hatanın iterasyon sayısına göre değişimi

Şekil (7.4)'ten görüldüğü gibi, yapay sinir ağı öngörülen hataya, 6678 iterasyon sonunda ulaşmıştır. Eğitimin bittiği anda, hatanın etkin değeri rms = 0,0024, ve ortalama mutlak hata % 1,95 olarak bulunmuştur. Test kümesinin çıkışının gerçek değeri 45º iken, yapay sinir ağı çıkışı 44,9805 olarak bulunur.

7.5.3 Porselen İzolatörün Konum Açısının Yapay Sinir Ağı İle Bulunması

Uygun parametreleri bulmak için bölüm (7.5.1) ve (7.5.2)'de anlatılan çalışmalar, porselen izolatör için de yapılmıştır.

Porselen izolatörün yerleşim açısının yapay sinir ağı ile belirlenmesi probleminde, ağın güvenilirliğini sınamak için, elektrot açıklığı a = 26 cm iken, izolatörün yerleşim

açısı 45º olarak alınarak, ε_{r1} = 4 için elektrik alan değerleri hesaplanarak test kümesi oluşturulur. Test kümesi giriş vektörü [26 0.5765] ve çıkış vektörü [45] olarak belirlenir. Uygun parametrelerin belirlenmesi için yapılan çalışmalar tablo (7.10) ve (7.11)'de görülmektedir.

Tablo 7-10 Momentum katsayısının hatalara etkisi

η = 0,1; gizli katman nöron sayısı=10; iterasyon sayısı=1000		
α	rms	mae
0,9	0,0032	2,4849
0,8	0,0047	1,8562
0,7	0,0085	2,8997
0,6	0,0045	5,4120
0,5	0,0106	4,9450
0,4	0,0247	0,3295
0,3	0,0332	8,1272
0,2	0,0167	2,6379
0,1	0,0436	0,9941

Tablo 7-11 Öğrenme hızının hatalara etkisi

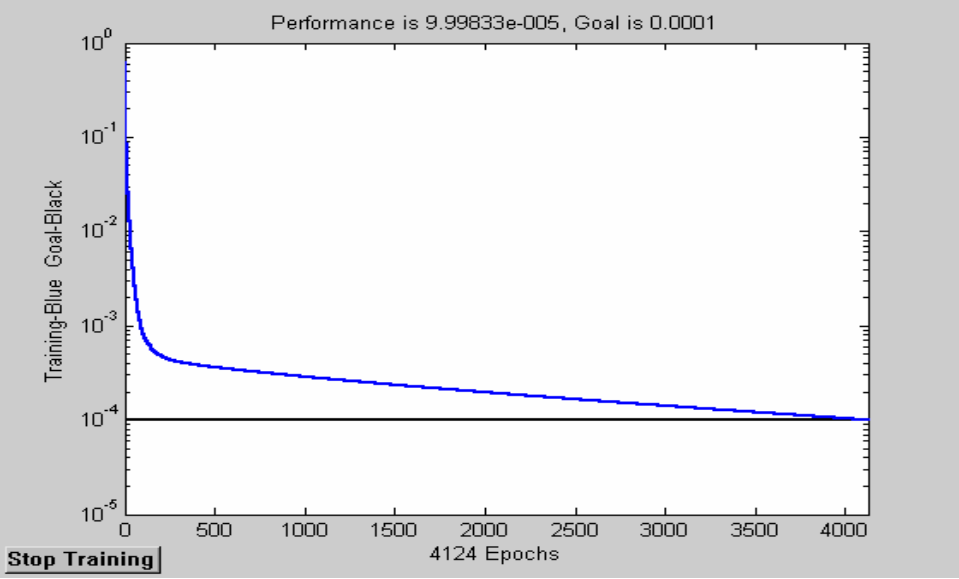
α = 0,8; gizli katman nöron sayısı=10; iterasyon sayısı=1000		
η	rms	mae
0,1	0,0017	1,8562
0,2	0,0117	0,5714
0,3	0,0103	0,5717
0,4	0,0049	0,1566
0,5	0,0011	1,6387
0,6	0,0082	0,7436
0,7	0,0076	3,097
0,8	0,0067	0,7925

Tablo 7-12 Gizli katman nöron sayısının hatalara etkisi

İterasyon sayısı:1000		
Gizli katman nöron sayısı	rms	mae
15	0,0020	3,9719
10	0,0011	0,1566
9	0,0010	1,1188
8	0,0041	0,7069
7	0,0046	3,799
6	0,0044	0,8535
5	0,0021	0,4253
4	0,0112	1,1494

En uygun öğrenme hızı α = 0,4 ve momentum katsayısı η = 0,8 olarak belirlendikten sonra, gizli katman nöron sayısı araştırılır (Tablo (7.12)).

En uygun yapay sinir ağı parametreleri öğrenme hızı α = 0,4 ve momentum katsayısı η = 0,8, gizli katman nöron sayısı=10 olarak belirlendikten sonra, toplam karesel hata 0,0001'den küçük olana kadar eğitim sürdürülerek, eğitim ve test hataları gözlenmiştir.



Şekil 7-5 Porselen izolatör için hatanın iterasyon sayısına göre değişimi

Şekil (7.5)'ten görüldüğü gibi, yapay sinir ağı öngörülen hataya, 4124 iterasyon sonunda ulaşmıştır. Eğitimin bittiği anda, hatanın etkin değeri rms = 0,0023, ve ortalama mutlak hata % 0,92 olarak bulunmuştur. Test kümesinin çıkışının gerçek değeri 45º iken, yapay sinir ağı çıkışı 45,0093 olarak bulunur.

7.6 Sonuç

Çalışmanın bu bölümünde, özellikle gaz yalıtımlı sistemlerde kullanılan, doğrusal biçimli üç farklı izolatörün (epoksi reçine, cam ve porselen) yerleşim açısı, yapay sinir ağı kullanılarak belirlenmiştir. Bu amaçla kullanılan yapay sinir ağı, ileri beslemeli, çok katmanlı bir yapay sinir ağıdır. Eğitim sürecinde ağırlıkların hesaplanması için dik iniş algoritmasıyla optimizasyon yapan, momentum katsayılı geriye yayılım algoritması kullanılmıştır. Ağın eğitim parametrelerinin belirlenmesi amacıyla, her izolatör durumu için ayrıntılı çalışmalar yapılmış ve en uygun ağ parametreleri belirlenmiştir. Eğitim sonunda ağın güvenilirliği, eğitim kümesinin dışından bir test kümesi belirlenerek sınanmıştır. Kullanılan eğitim parametrelerinin

uygunluğu, test hatasında kendini göstermektedir. Kullanılan ağ, eğitim kümesinden farklı bir giriş değeri için %2'den küçük hatayla cevap verebilir.

Bu problem için yapay sinir ağı kullanılması gerçekten oldukça etkin ve kesin sonuçlar vermektedir. Yapay sinir kullanılmasının üstünlüğü, sadece sınırlı sayıda hesaplama yapılarak genel bir çözüme ulaşmaktır. Böyle bir problem için kullanılan sayısal yöntemler, her iterasyon için pekçok hesaplama yapılmasını gerektirmektedir. Bu sebeple, çok fazla zaman almaktadır. Yapay sinir ağlarının kullanılması ise 15 örüntüsü bulunan bu yapıda, günümüz bilgisayarlarında yaklaşık 100-200 saniyede sonuç vererek, sayısal yöntemlerin zaman alıcılığını giderir ve genel çözümlere ulaşmamızı sağlar.

8 SONUÇLAR VE ÖNERİLER

Yapay sinir ağı ile izolatör ve elektrot optimizasyonu konulu tez çalışması, üç temel bölümden oluşmaktadır.

Birinci bölümde, basınç, bağıl nem sıcaklık gibi koşullara bağlı olarak, yüksek gerilim izolatörlerinin darbe atlama gerilimlerinin belirlenmesinde yeni bir yaklaşım açıklanmıştır. Bu amaçla yapılan çalışmada farklı basınçlar ve ortam koşulları altında bulunan farklı izolatör tipleri değerlerine karşılık, negatif ve pozitif darbe atlama gerilim değerleri, standart geriye yayılım algoritması ile yapay sinir ağına tanıtılmıştır. Kullanılan öğrenme algoritması, optimizasyon yöntemi olarak dik iniş (steepest descent) yöntemini kullanan, momentum terimli geriye yayılım algoritmasıdır.

Verilen bir problem için kullanılacak yapay sinir ağı yapısı seçimi, kurallara bağlı değildir, kullanılacak yapay sinir ağı, tasarımcının öngörüsü ve deneyimlerine göre seçilir. En uygun yapay sinir ağı yapısını oluşturan en uygun eğitim parametrelerini bulmak için, deneme yanılma yöntemi ile ayrıntılı çalışmalar yapılmıştır. Bu çalışmalar sonucunda seçilen ağ parametreleri ile yapay sinir ağı eğitilmiş ve ağın güvenilirliğini sınamak amacıyla bir test kümesi ağa uygulanmıştır. Test kümesinin giriş vektörü ağa giriş olarak uygulanarak, yapay sinir ağının çıkışı ile test kümesinin gerçek değeri arasındaki ortalama mutlak hata (mean absolute error-mae) incelenmiştir. parametrelerin belirlenmesi aşamasında, ortalama mutlak hatanın %3'ün üstüne hiç çıkmadığı gözlenmiştir. Bu da en kötü parametrelerin uygulanmasında bile, yapay sinir ağının oldukça doğru ve etkin sonuçlar verdiğini göstermektedir. En uygun eğitim parametreleri ile yapay sinir ağı eğitildiğinde ise, toplam karesel hata % 1, hatanın etkin değeri %1,96, ve ortalama mutlak hata %0,93 olarak hesaplanmıştır. Eğitim ve test hatalarının değerleri, karar verilen eğitim parametrelerinin uygun olduğunu göstermektedir.

Çalışmanın ikinci bölümünde, bir çubuk – düzlem elektrot sisteminde, çubuk elektrotun küresel bölgesi, yüzeyde düzgün elektrik alan sağlayacak şekilde optimize edilmiştir.

Bu bölümde, eğitim sürecinde optimizasyon tekniği olarak Levenberg – Marquardt yöntemini kullanan geriye yayılım algoritması kullanılmıştır. Levenberg – Marquardt algoritması, yavaş yakınsama probleminden etkilenmeyen, çok hızlı yanıt veren bir algoritmadır. Algoritmanın her iterasyonunda performans fonksiyonu daima azaltıldığı için, ağ çok kısa sürede, çok küçük hata değerlerine ulaşılır. Çubuk elektrotun kritik bölgesinin optimizasyonu için yapılan bir denemede, gerçek değerler ile yapay sinir sonuçları arasındaki ortalama karesel hata % 0,01'den daha küçük olarak hesaplanmıştır.

Üç farklı yalıtkan malzemeden (epoksi reçine, cam, porselen) yapılmış izolatörlerin elektrot açıklığı ve gaz yalıtkandaki elektrik alan değerlerine göre, yerleşim açılarının bulunduğu çalışmanın diğer bir bölümünde de standart geriye yayılım algoritması kullanılmıştır. Atlama geriliminin bulunması probleminde parametrelerin belirlenmesi için yapılan çalışmalara benzer çalışma, gaz yalıtımlı sistemlerde kullanılan doğrusal biçimli mesnet izolatörünün yerleşim açısının bulunması problemi için de uygulanmıştır. En uygun ağ yapısıyla eğitilen yapay sinir ağının bulduğu sonuçlar ile, uygulanan test kümesinin gerçek çıkışları arasındaki ortalama mutlak hata, her izolatör durumu için yaklaşık % 1 olarak bulunmuştur.

Algoritma uygulanırken yapay sinir ağına tanıtılan giriş örüntüleri ile yapay sinir ağı eğitimini tamamlar. Ancak geriye yayılım algoritması, eğitim parametreleri yanında ilk koşullara da çok duyarlı olduğu için, her zaman başarılı sonuç alınmayabilir. Bu durumda, farklı ilk koşullar denenerek en iyi sonuca ulaşılmaya çalışılır.

Yapay sinir ağına uygulanan giriş – çıkış örüntüleri yeterli sayıda olmalıdır. Çok az sayıda örüntü ile eğitilen ağ, kendisine tanıtılan giriş örüntüleri ve hedeflenen çıkış örüntüleri arasındaki ilişkiyi sağlıklı bir şekilde kuramaz. Bununla beraber, eğitim kümesinin giriş örüntülerinin birbirine çok yakın veya aynı giriş değerlerine karşılık, birbirinden çok farklı çıkış değerlerine sahip bir eğitim kümesi, yapay sinir ağında ezberleme problemine çok açabilir. Bu sebeple eğitim kümesi dikkatli bir biçimde belirlenmelidir.

Bu çalışmada uygulanan, yüksek gerilim izolatörlerinin darbe atlama gerilimlerinin bulunması probleminde, daha farklı izolatör tipleri kullanılarak, yalıtkan ortam değiştirilerek değişik çalışmalar yapılabilir.

Çalışmada göz önüne alınan doğrusal biçimli mesnet izolatörlerinin, farklı bir kaç noktasında yapılan teğetsel elektrik alan hesaplamalarına karşılık, noktaların r – z koordinatları kullanılarak yapay sinir ağı eğitildiğinde, yüzeyde istenen elektrik alan

dağılımını sağlayacak optimum izolatörün r –z koordinatları belirlenebilir. Bu amaçla, değişik yapılarda (etekli, eteksiz, doğrusal, eliptik, üstel,...) biçimli izolatörler kullanılarak, biçim optimizasyonu açısından geniş bir çalışma alanı elde edilebilir.

Ayrıca çalışmada anlatılan çubuk – düzlem elektrot sistemi yerine, herhangi bir elektrot sistemi kullanılarak, düzgün alan dağılımı sağlayan optimum elektrot biçimi aranabilir.

Bugüne kadar yapılan ve bundan sonra yapılacak olan biçim optimizasyonu çalışmaları, deneysel yöntemlerin pahalılığını ve sayısal yöntemleri zaman alıcılığını azaltacaktır.

Bu çalışmada ele alınan problemlerin yapay sinir ağı ile çözülmesi, çok etkili ve kesin sonuçlar vermektedir. Yapay sinir ağı kullanılarak, sadece sınırlı sayıda deney veya hesaplama yapılarak, genel bir çözüme ulaşılabilir. Deneysel yöntemlerin pahalılığı ve biçim optimizasyonu için kullanılan iteratif yöntemlerin zaman alıcılığı, yapay sinir ağı ile giderilmiş olur.

KAYNAKLAR

[1] **M. Özkaya**, 1996. Yüksek Gerilim Tekniği, Cilt 1, Birsen Yayınevi, İstanbul.

[2] **Salama, M. M. A. and Bartnikas, R.,** 2002. Determination Of Neural Network Topology for Partial Discharge Pulse Pattern Recognition, *IEEE Transactions on Neural Networks,* **Vol. 13, No. 2,** pp. 446- 456.

[3] **Ghosh, P. S., Chakravorti, S. and Chatterjee, N.,** 1995. Estimation of Time-to-flashover Characteristics of Contamined Electrolytic Surfaces Using a Neural Network, *IEEE Transactions on Neural Networks,* **Vol. 2, No. 6,** pp. 1064-1047.

[4] **Ghosh, P. S. and Kishore, N. K.,** 1999. Modeling PD Inception Voltage of Epoxy Resin Post Insulators Using an Adaptive Neural Network, *IEEE Transactions on Dielectrics and Electrical Insulation,* **Vol. 6, No. 1**, pp. 131-134.

[5] **Haykin, S.,** 1999. Neural Neworks: A Comprehensive Foundation Second Edition, Prentice Hall, New Jersey.

[6] **Şengör, N., S.,** 2001. Yapay Sinir Ağlarına Giriş, Ders Notları.

[7] **Oğuz, M.,** 2001. Yalıtkan Maddelerde Elektriksel Delinme Dayanımının Yapay Sinir Ağı İle Belirlenmesi, *Yüksek Lisans Tezi,* İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.

[8] **Anderson, D. and McNeill, G.,** 1992. Artificial neural networks technology, *A DACS State-of-the-Art Report,* **F30602-89-C-0082**, New York.

[9] **Şeker, S.,** 2001. Artificial Neural Networks for System Identification and Control, Ders Notları.

[10] **Baklavacı, S.,** 1994. Analysis of Learning Algorithms in Neural Networks, *Msc Thesis,* İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.

[11] **Bihsop, C. M.,** 1995. Neural Network for Pattern Recognition, Oxford University Press, Oxford.

[12] **Caudill, M. And Butler, C.,** 1992.Understanding Neural Network: Computer Explorations: a workbook in two volumes with software for the Macintosh and PC compatibilities, MIT Press, Cambridge.

[13] **Blum, A.,** 1992. Neural Networks in C ++: An Object-Oriented Framework For Building Connectionist System, Wiley, New York.

[14] **Sontag, D.,**1992. Feedback Stabilization Using Two-Hidden-Layer-Nets, *IEEE Transaction on Neural Networks,* **Vol. 3, No.6,** 981-990.

[15] **Aydınay Ş. ve Kalenderli Ö.,** 2000. Alçak Basınçta Havanın Delinme Dayanımı, *ELECO'2000, Elektrik Elektronik Bilgisayar Mühendisliği Sempozyumu,* Bursa, s. 141-145.

[16] **Aydınay Ş. ve Kalenderli Ö.,** 2000. Alçak Basınçta Reçine İzolatörlerde Atlama, *ELECO'2000, Elektrik Elektronik Bilgisayar Mühendisliği Sempozyumu, Bursa,* s. 146-149.

[17] **Türk L.,** 1989 Vakum Teknolojisi ve Vakum Kesicilerin Akım Kesme Özellikleri, *Elektrik Mühendisliği 3. Ulusal Kongresi,* İstanbul, s. 236-239.

[18] **Öztaş H., Yılmaz G. ve Kalenderli Ö.,** 1995. Darbe Geriliminde Kabloların Dielektrik Özelliklerine Sıcaklığın Etkisi, *Elektrik Mühendisliği 6. Ulusal Kongresi,* Bursa, s. 286-289.

[19] **İsmailoğlu H., Kalenderli Ö., Özkaya M. ve Gönenç İ.,** 1995. Darbe geriliminde havanın delinme dayanımı, *Elektrik Mühendisliği 6. Ulusal Kongresi,* Bursa, s. 257-260.

[20] **S. Bolat, Ö. Kalenderli, E. Önal ve A. Merev,** 2002. Yüksek Gerilim İzolatörlerinin Darbe Atlama Gerilimlerinin Yapay Sinir Ağı ile Belirlenmesi, *ELECO'2002, Elektrik Elektronik Bilgisayar Mühendisliği Sempozyumu,* Bursa, s. 154-158.

[21] **Singer, H.,** 1979. Computation Of Optimized Electrode Geometries, *The Third International Symposium on High Voltage Engineering,* Milan, Paper 11.06.

[22] **Liu, J. And Sheng, J.,** 1998. The Optimization of the High Voltage Axisymmetrical Electrode Contour, *IEEE Transactions on Magnetics*, **Vol.24, No.1,** pp.39-42.

[23] **Caminhas, W. M., Saldanha, R. R. and Mateus, G. R.,** 1990. Optimization Methods Used for Determining the Geometry of Shielding Electrodes, *IEEE Transactions on Magnetics,* **Vol.26, No.2,** pp. 642-645.

[24] **Trinities, C., Steinbigler, H., Spasojevic, M., Levin, P. L. and Andjelic, Z.,** 1995. Accelerated 3-D Optimization of High Voltage Apparatus, *The Ninth International Symposium on High Voltage Engineering,* Graz, Austria, paper 8867.

[25] **Kato, K., Hikita, M., Hayakawa, N. and Okubo, H.,** 1995. High Efficient Method for Determination of Electric Field Optimum Contour of High Voltage Electrode, *The Ninth International Symposium on High Voltage Engineering,* Graz, Austria, paper 8358.

[26] **Kato, K., and Okubo, H.,** 1997. Optimization of High Voltage Electrode Contour with the Highest Gaseous Insulation Performance, *IEEE Transaction on Dielectrics and Electrical Insulation*, **Vol.4, No.6,** pp.816-821.

[27] **Kato, K., Han, X. and Okubo, H.,** 1999. Optimization Technique of High Voltage Electrode Contour Considering V-t Characteristics Based on Volume-time Theory, *IEEE High Voltage Engineering Symposium,* Conference publication, No.2, pp.2.123-2.126.

[28] **Kato, K., Han, X. and Okubo, H.,** 2001. Insulation Optimization by Electrode Contour Based on Area/Volume Effect, *IEEE Transactions on Dielectrics and Electrical Insulation*, **Vol.8, No.2,** pp.162-167.

[29] **Stih, Z.,** 1986. High Voltage Insulating System Design by Application of Electrode and Insulator Contour Optimization, *IEEE Transactions on Electrical Insulation*, **Vol. EI-21, No. 4,** pp. 579-584.

[30] **Phillips, D. B., Olsen R. G. and Pedrow, P. D.,** 2000. Corona Onset as a Design Optimization Criterion for High Voltage Hardware, *IEEE Transactions on Dielectrics and Electrical Insulation,* **Vol. 7, No. 6,** pp. 744-751.

[31] **Chatterjee, A., Rakshit, A. and Mukherjee, P. K.,** 2001. A Self-Organizing Fuzzy Inference System for Electric Field Optimization of High Voltage Electrode Systems, *IEEE Transactions on Dielectrics and Electrical Insulation,* **Vol. 18, No. 6,** pp. 995-1001.

[32] **Chakravorti, S. and Mukherjee, P. K.,** 1994. Application of Artificial Neural Networks for Optimization of Electrode Contour, *IEEE Transactions on Dielectrics and Electrical Insulation,* **Vol. 1, No. 2,** pp. 254- 263.

[33] **Mukherjee, P. K., Trinities, C. and Steinbigler, H.,** 1996. Optimization of HV Electrode Systems by Neural Network using a New Learning Method, *IEEE Transactions on Dielectrics and Electrical Insulation,* **Vol. 3, No. 6,** pp. 737-742, 1996.

[34] **Okubo, H., Otsuka, T. Kato, K., Hayakawa, And Hikita, M.,** 1997. Electric Field Optimization of High Voltage Electrode Based on Neural Network, *IEEE Transactions on Power Systems,* **Vol. 12, No. 4,** pp. 1413-1418.

[35] **Pillai, A. S. and Hackam, R.,** 1984. Optimal Electrode – Solid Insulator Geometry with Accumulated Surface Charges, *IEEE Transactions Electrical Insulation,* **Vol. EI-19, No. 4,** pp. 321-331.

[36] **Moreno, G. and Portela, C. M.,** 1995. Electromagnetics Field Assesing of Transmission Lines for Design Optimization Considering the Effect of Insulating and Supporting Components, *The Ninth International Symposium on High Voltage Engineering,* Graz, Austria, paper 8322.

[37] **Garcia, J. Á. G. and Singer, H.,** 1995. Contour Optimization of High Voltage Insulator by Means of Smoothing Cubic Splines, *The Ninth International Symposium on High Voltage Engineering,* Graz, Austria, paper 8342.

[38] **Skopec, A., Wankowicz, J. G. and Sikorski, B.,** 1994. Electric Field Calculation for an Axially - Symmetric Insulator with Surface Contamination, *IEEE Transactions on Dielectrics and Electrical Insulation,* **Vol. 1, No. 2,** pp.332-339.

[39] **Abdel-Salam, M. and Stanek, E. K.,** 1987. Field Optimization of High Voltage Insulators, *IEEE Transactions on Electrical Insulation,* **Vol. 22, No. 1**, pp. 47-56.

[40] **Mila, J., Opydo, W. and Batura, R.,** 2000. Optimization of Insulator for Vacuum Insulated Devices, *IEEE Ninth Symposium on Discharges and Electrical Insulators in Vacuum,* pp. 116-118.

[41] **Pillai, S. A., Hackam, R. and Alexander, P. H.,** 1983. Influence of Radius of Curvature, Contact Angle and Material of Solid Insulator on the Electric field in Vacuum (and Gaseous) Gaps, *IEEE Transactions Electrical Insulation,* **Vol. EI-18, No. 1,** pp. 11-22.

[42] **Cao, Y. Liu, X., Wang, E., Jin, L. and Wang, G.,** 2002. Electric Field Optimization Design of a Vacuum Interrupter Based on the Tabu Search Algorithm, *IEEE Transactions on Dielectrics and Electrical Insulation,* **Vol. 9, No. 2**, pp. 169-172.

[43] **Bhattacharya, K., Chakravorti, S. and Mukherjee, P. K.,** 2001. Insulator Contour Optimization by a Neural Network, *IEEE Transactions on Dielectrics and Electrical Insulation,* **Vol. 8, No. 2,** pp. 157-161.

[44] **Bhattacharya, K., Chakravorti, S. and Mukherjee, P. K.,** 1995. Optimization of High Voltage Support Insulator Contour by Artificial Neural Network, *The Ninth International Symposium on High Voltage Engineering,* Graz, Austria, paper 8868.

[45] **Techaumnat, B., Hamada, S. and Takuma, T.,** 2002. Electric Field Behavior Near a Zero-Angle Contact Point in the Presence of Surface Conductivity, *IEEE Transactions on Dielectrics and Electrical Insulation,* **Vol. 9, No. 4**, pp. 537-543.

[46] **Chakravorti, S. and Siddanangappa, M. C.,** 1995. Effect of Charge Accumulation on the Electric Field Distribution Along Spacer Surfaces, *The Ninth International Symposium on High Voltage Engineering,* Graz, Austria, paper 8352.

[47] **Kalenderli, Ö. ve İsmailoğlu, H.,** 1991. Gaz Yalıtımlı Sistemlerde Ara Tutucuların Alan Dağılımına Etkisinin Sonlu Elemanlar Yöntemi Ile Incelenmesine Bir Uygulama, *Elektrik Mühendisliği IV. Ulusal Kongresi,* s. 272-275.

[48] **Takuma, T., Kawamoto, C. and Honda, C.,** 1995. Analysis of Electric Field in Uniform-Field Electrodes, *The Ninth International Symposium on High Voltage Engineering,* Graz, Austria, paper 8361.

[49] **Frankel, D., Schiller, I., Draper, J.S. and Barnes, A.A.,** 1991**.** Use of neural networks to predict lightning at Kennedy Space Center*, IJCNN-91-Seattle International Joint Conference on Neural Networks,* 8-14 July, vol.1, pp.319 - 324.

[50] **Bermudez, J.L., Piras, A. And Rubinstein, M.,** 1996. Artificial neural networks in lightning location systems, *AT'96., International Symposium on Neuro-Fuzzy Systems,* 29-31 Aug., pp. 177-178.

[51] **Singer, H., Steinbigler, H. and Weiss, P.,** 1974. A Charge Simulation Method For the Calculation of High Voltage Fields, *IEEE Transactions on Power Apparatus and Systems,* **Vol.PAS-93,** 1660-1668.

[52] **Khan, M. J. and Alexander, P. H.,** 1982. Charge Simulation Modeling of Practical Insulator Geometries, *IEEE Transaction on Electrical Insulation,* **Vol EI-17 No:4,** 325-332.

[53] **Mukherjee, P. K. and Roy, C. K.,** 1978. Computation of Fields in and Around Insulators by Fictitious Point Charges, *IEEE Transaction on Electrical Insulation,* **Vol EI-13, No:1**, 24-31.

[54] **Kalenderli, Ö.,** 2003. Elektrik Mühendisliğinde Sonlu Elemanlar Yöntemi, Ders Notları.

[55] **Malik, N. H.,** 1989. A Rewiev of The Charge Simulation Method and Its Applications, *IEEE Transaction on Electrical Insulation,* **Vol 24, No:1**, pp. 3-20.

[56] **Takuma, T. and Kawamoto, T.,** 1979. Field Calculation Including Surface Resistance by Charge Simulation Method, Third International Symposium on High Voltage Engineering, Milan, paper 12-01.

[57] **Yıldırım, H.,** 1992. Elektrostatik Alanların Yük Benzetim Yöntemiyle ile İncelenmesi, *Yüksek Lisans Tezi,* İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.

EK A

YÜK BENZETİM YÖNTEMİ – YBY (CHARGE SIMULATION METHOD – CSM)

A.1 Giriş

Elektrik alan, sınır koşullarında Laplace ve Poisson denklemlerinin çözülmesiyle hesaplanır. Bunun için çözümsel veya sayısal yöntemler kullanılabilir. Pek çok durumda fiziksel sistemlerin karmaşık yapısından dolayı, çözümsel yöntemlerin kullanılması çok zor veya imkansızdır. Bu sebeple, mühendislik uygulamalarında sayısal yöntemler kullanılır. Yük benzetim yöntemi de, çözümü sayısal olmakla beraber, elektrik alanın çözümsel olarak hesaplandığı bir yöntemdir. Bu durum uygulamada oldukça kolaylık sağlar.

Elektriksel alanların hesabında çok etkin ve doğru bir yöntem olan yük benzetim yöntemi, son yıllarda önemli gelişmeler kaydederek, elektrik mühendisliğinde üç boyutlu ve çok yalıtkanlı düzenlerin eksenel simetrisi olmayan problemlerin ve sınır optimizasyonu problemlerinin çözümünde ve gazlarda boşalma olaylarının incelenmesinde başarıyla kullanılır.

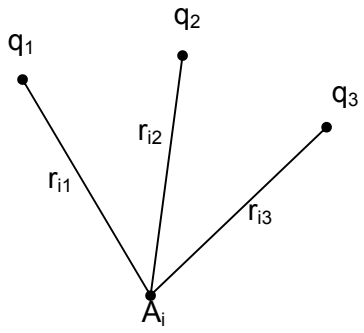
Yük benzetim yöntemiyle sayısal alan çözümleri için ilk çalışmalar Loeb ve arkadaşları ile Abou Seada ve Nasser tarafından yapılmıştır. Bu çalışmalarda çubuk – düzlem ve silindirsel iletken – düzlem elektrot sistemlerinin alan çözümleri bulunmuştur. Son olarak Singer, Steinbigler ve Weiss kapsamlı bir makalede yük benzetim yöntemini ve bu yöntemin dönel simetriye sahip iki ve üç boyutlu alanlara uygulanmasının ayrıntılarını vermiştir. O zamandan beri bu yöntem birçok yönden geliştirilerek, çeşitli alan problemlerinin çözümünde başarılı bir şekilde kullanılmıştır.

Yük benzetim yönteminin temel ilkesi elektroteknikteki toplama ilkesine dayanır. Elektrotlar üzerinde fiziksel olarak dağılmış elektrik yükleri yerine ayrık yükler konur. Bu yükler alan çözümü istenen yalıtkan bölgenin dışına, elektrot gibi bir eşpotansiyel yüzey arkasına yerleştirilir ve benzetim yükü adını alır. Yüklerin değerleri sınır koşullarından belirlenir. Sınır koşulu olarak elektrot sınırı üzerinde alınan m adet sınır noktasının potansiyeli kullanılır. Buna göre, eğer bir iletken bölge içersinde

herhangi bir tipten birçok ayrık yük (noktasal, çizgisel veya ayrık yük) varsa, alan çözümü istenen yalıtkan bölge içersindeki veya elektrot sınırındaki herhangi bir A noktasındaki elektrostatik potansiyel (A noktası herhangi bir yükün bulunduğu nokta olmadığı sürece), herbir yükün ayrı ayrı A noktasında oluşturduğu potansiyellerin toplamına eşittir. q_j, n adet ayrık yük ve V_i de bu bölge içinde herhangi bir A noktasındaki potansiyel olmak üzere, toplama ilkesine göre,

$$V_i = \sum_{j=1}^{n} p_{ij} q_j \qquad (A.1)$$

dir. Burada pij, potansiyel katsayısı adını alır. p_{ij}'nin değeri, her yük tipi için Laplace veya Poisson denkleminin çözümünden belirlenir.



Şekil –A.1 Üç adet noktasal yük ve potansiyeli hesaplanacak olan A_i noktası

Şekil (A.1)'de gösterilen, serbest uzaydaki üç adet noktasal yükün (q_1, q_2, q_3) A_i noktasında oluşturdukları V_i potansiyeli,

$$V_i = \frac{q_1}{4\pi\varepsilon r_{i1}} + \frac{q_2}{4\pi\varepsilon r_{i2}} + \frac{q_3}{4\pi\varepsilon r_{i3}} = p_{i1}.q_1 + p_{i2}.q_2 + p_{i3}.q_3 \qquad (A.2)$$

bağıntısıyla bulunabilir. Burada r_{ij} (j=1, 2, ..., n), j. yük ile A_i noktası arasındaki uzaklıktır, değeri ise yüklerin ve A_i noktasının koordinatlarından yararlanılarak hesaplanır. E ise yalıtkan ortamın dielektrik sabitidir. Eğer A_i noktası elektrot yüzeyi üzerinde alınmış bir nokta ise bu noktaya sınır noktası adı verilir ve bu noktanın potansiyeli elektrot potansiyeli olan V_e'ye eşittir. Böylece, elektrot potansiyeli bilindiğine göre, m adet sınır noktası için potansiyel bağıntıları yazılarak n adet bilinmeyen benzetim yükü için m adet denklemden oluşan bir lineer denklem sistemi elde edilir (Denklem A.3).

$$\begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \vdots & \vdots & & \vdots \\ p_{m1} & p_{m2} & \cdots & p_{mn} \end{bmatrix} \cdot \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{bmatrix} = \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_n \end{bmatrix} \tag{A.3}$$

Yük benzetim yönteminin temel bağıntısı olan denklem (A.3), genellikle simetrik bir denklem sistemi değildir. Bu denklem çözülerek q yüklerinin değerleri belirlenir. Bu yüklere denklem (A.1)'den yalıtkan bölge içindeki herhangi bir noktanın potansiyeli ve elektrik alan şiddeti kolaylıkla hesaplanabilir.

A.2 Benzetim Yükü Tipleri

Yük benzetim yönteminin başarısında, uygun tipte benzetim yükü seçilmesi önemli bir yer tutar. Bu yöntemin kullanıldığı ilk çalışmalarda noktasal yükler, sonsuz uzunlukta ve yarısonsuz uzunlukta çizgisel yükler kullanılmıştır. Singer ve arkadaşları, yaptıkları çalışmada yoğunluğu periyodik olarak değişen halkasal yükleri ve sonlu çizgisel yükleri kullanmışlardır. Son yıllarda daha başka yük tiplerinin kullanıldığı çalışmalara da rastlanmaktadır. Bunlar arasında eliptik silindirsel yükler, sabit yük yoğunluklu halkasal yükler, hacimsel yükler, tabaka ve halka şeklinde düzlemsel yükler ve değişken yoğunluklu çizgisel yükler sayılabilir.

Herhangi bir tip benzetim yükünü kullanmak için potansiyel ve alan katsayılarının bilinmesi gerekir. Bu katsayılara ilişkin bağıntılar, yukarda sayılan yük tipleri için literatürde verilmiştir. Genellikle noktasal ve çizgisel yüklerin potansiyel ve alan katsayıları, hesaplama süresi az olan basit çözümsel bağıntılarla verilebilir.

1. Noktasal Yük:

   Noktasal yük için p potansiyel katsayısı ve f alan katsayısı aşağıda verilmiştir. Denklem (A.4)'te r uzaklık, denklem (A.5)'te a, b, c ise yükün koordinatları ve ε dielektrik katsayısıdır.

$$p = \frac{1}{4\pi\varepsilon r} \tag{A.4}$$

$$f_x = \frac{x-a}{4\pi\varepsilon r^3}$$
$$f_y = \frac{y-a}{4\pi\varepsilon r^3} \qquad (A.5)$$
$$f_z = \frac{z-a}{4\pi\varepsilon r^3}$$

2. Sonsuz Çizgisel Yük

Sonsuz çizgisel yük için p potansiyel katsayısı ve f alan katsayısı aşağıdaki gibidir. Denklem (A.6)'da görülen r_0, sıfır potansiyelli en yakın nokta ile yük arasındaki uzaklıktır.

$$p = \frac{\ln(r_0 / r)}{2\pi\varepsilon r} \qquad (A.6)$$

$$f_x = \frac{x-a}{2\pi\varepsilon r^2}$$
$$f_y = \frac{y-a}{2\pi\varepsilon r^2} \qquad (A.7)$$
$$f_z = \frac{z-a}{2\pi\varepsilon r^2}$$

Noktasal ve sonsuz çizgisel yükler için yukarda verilen bağıntılarda, görüntü yüklerinin etkisi göz önüne alınmamıştır.

3. Halkasal Yük

Halkasal yükün potansiyel katsayısı, K(k), birinci türden komple eliptik integral olmak üzere, denklem (A.8)'deki gibi yazılabilir:

$$p = \frac{1}{4\pi^2\varepsilon}\left[\frac{K(k_1)}{\alpha_1} - \frac{K(k_2)}{\alpha_2}\right] \qquad (A.8)$$

Burada

$$\alpha_1 = \sqrt{(r+R)^2 + (z-Z)^2}$$
$$\alpha_1 = \sqrt{(r+R)^2 + (z+Z)^2} \qquad (A.9)$$

dır. Halkasal yükün alan bileşenleri ise,

$$f_r = \frac{1}{4\pi^2\varepsilon r}\left\{-\frac{\left[R^2 - r^2 + (Z-z)^2\right]E(k_1) - \beta_1^2 K(k_1)}{\alpha_1 \beta_1^2} + -\frac{\left[R^2 - r^2 + (Z+z)^2\right]E(k_2) - \beta_2^2 K(k_2)}{\alpha_2 \beta_2^2}\right\} \quad \text{(A.10)}$$

$$f_z = \frac{2}{4\pi^2 \varepsilon}\left\{\frac{(Z-z)E(k_1)}{\alpha_1 \beta_1^2} - \frac{(Z+z)E(k_2)}{\alpha_2 \beta_2^2}\right\} \quad \text{(A.11)}$$

şeklinde yazılabilir.

Daha karmaşık şekilli yük tipleri için potansiyel ve alan katsayılarının, zorunlu olarak sayısal bir yöntemle hesaplanması gerekebilir. Diğer taraftan elektrot sisteminin geometrik şekline uygun karmaşık şekilli yük tipleri kullanılırsa daha az sayıda yükle, verilen bir sistemin benzetimi yapılabilir. Bu yüzden, basit olanların yerine karmaşık yük tipleri kullanılırsa, bilgisayar için gerekli bellek ve hesaplama süresi daha az olabilir. Genellikle yüksek gerilim sistemlerinin çoğunun benzetimi, temel yük tipleri yani noktasal, çizgisel ve halkasal yükler veya bunların uygun bileşikleri kullanılarak başarılı bir şekilde yapılabilir.

A.3 Tek Yalıtkanlı Sistemler İçin Yük Benzetim Yöntemi

Tek yalıtkanlı elektrot sisteminde, potansiyelleri bilinen N adet elektrot (iletken) var iken, iletkenlerin içine veya alanı incelenecek bölgenin dışına n adet benzetim yükü yerleştirilir. Yüklerin tipi ve yerleri için kabul yapılır. Yük tipi, elektrot sisteminin geometrik şekline uygun seçilir.

Benzetim yüklerinin değerlerini belirlemek için elektrot yüzeyi üzerinde m adet sınır noktası seçilir. Sınır koşulu olarak, herhangi bir sınır noktasında bulunan bütün benzetim yüklerinin potansiyellerinin toplamından oluşan V_i potansiyelinin, iletkenin V_e potansiyeline eşit olması gerekir. Sınır noktalarının sayısı benzetim yüklerinin sayısına eşit (m=n) olması gerekir. Buna göre yük değerleri,

$$[p]_{n,n} \cdot [q]_{n,1} = [V]_{n,1} \quad \text{(A.8)}$$

denkleminden bulunur. Burada [p] potansiyel katsayıları matrisi, [q] yüklerden oluşan sütun matris ve [V] sınır noktalarının potansiyellerinden oluşan sütun matristir. Yüklerin değeri belirlendikten sonra bu yüklerin, elektrot yüzeyinin her yerinde sınır koşullarını sağlayıp sağlamadığını kontrol etmek gerekir. Çünkü denklem (A.8)'i çözmek için gerçek elektrot sistemi yerine, yalnızca n adet ayrık sınır noktası

kullanıldığından, elektrot yüzeyi üzerinde sınır noktaları dışındaki herhangi bir noktanın potansiyeli bu yüklerle hesaplandığında elde edilen potansiyel, gerçek elektrot yüzeyinden farklı olabilir. Bu nedenle, benzetimin doğruluğunu belirlemek için elektrot yüzeyinde sınır noktaları dışında, potansiyeli bilinen belirli sayıda sınır noktasında denklem (A.1) ile potansiyel hesabı yapılır. Elde edilen potansiyellerle bilinen sınır potansiyelleri (elektrot potansiyeli) arasındaki fark benzetimin doğruluğu için bir ölçüdür.

Hesaplanan sınır noktası potansiyelleri ile, o noktadaki elektrot potansiyelleri arasındaki fark kabul edilebilir. Değerden büyükse, aşağıdaki değişkenlerden bir veya birkaçının değiştirilmesiyle hesaplamalar yinelenerek yeni yük değerleri belirlenir. Değişkenler:

1. Benzetim yüklerinin sayısı,
2. Benzetim yüklerinin yerleri,
3. Benzetim yüklerinin tipi,
4. Sınır noktalarının yerleri.

Bu şekilde uygun yük değerleri elde edilir edilmez bu yüklerle elektrotlar dışındaki yalıtkan bölge içindeki herhangi bir noktanın potansiyeli ve elektriksel alanı toplama ilkesi ile çözümsel olarak kolayca hesaplanabilir. Potansiyel denklem (A.1)'in çözülmesiyle hesaplanırken elektriksel alan şiddeti, bu büyüklüğün çeşitli doğrultulardaki bileşenlerin toplanmasıyla hesaplanır. Örneğin üç boyutlu koordinat sisteminde herhangi bir C_i noktasındaki E_i elektriksel alanı,

$$\begin{aligned} E_i &= [\sum_{j=1}^{n} \frac{\partial p_{ij}}{\partial x} q_j]\,\vec{i}_x + [\sum_{j=1}^{n} \frac{\partial p_{ij}}{\partial y} q_j]\,\vec{i}_y + [\sum_{j=1}^{n} \frac{\partial p_{ij}}{\partial z} q_j]\,\vec{i}_z \\ &= [\sum_{j=1}^{n} (f_{ij})_x q_j]\,\vec{i}_x + [\sum_{j=1}^{n} (f_{ij})_y q_j]\,\vec{i}_y + [\sum_{j=1}^{n} (f_{ij})_z q_j]\,\vec{i}_z \end{aligned} \qquad (A.9)$$

bağıntısıyla hesaplanır. Buradaki $(f_{ij})_x$, $(f_{ij})_y$, $(f_{ij})_z$ katsayıları alan katsayıları olarak adlandırılır. $\vec{i}_x, \vec{i}_y, \vec{i}_z$ ise sırasıyla x, y, ve z doğrultularındaki birim vektörlerdir.

Birçok durumda bir iletken sistemi ile sıfır potansiyelli bir sonsuz düzlem arasındaki elektriksel alanın incelenmesi gerekir. Böyle durumlarda söz konusu düzlem görüntü yüklerinin başlangıcı olarak alınıp hesaba katılabilir. Eğer elektrot sisteminde potansiyeli bilinmeyen serbest elektrotlar varsa denklem (A.8) her bir serbest

elektrotun üzerindeki iç yüklerin toplamının sıfır olduğunu ifade eden ek bir koşulu içerecek şekilde değiştirilir. Eğer serbest elektrot bilinen bir yüke sahipse, iç yüklerin toplamı bu yük değerine eşit olur.

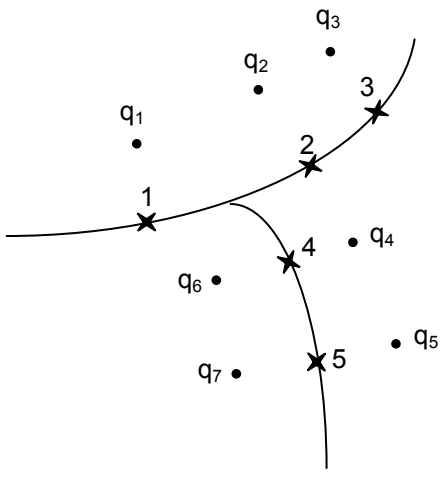
A.4 Çok Yalıtkanlı Sistemler İçin Yük Benzetim Yöntemi

Birden çok yalıtkandan oluşan bir sistem için yük benzetim yöntemi ile elektriksel alan hesapları, tek yalıtkanlı sistem için yapılan hesaplamalardan daha karmaşıktır. Bunun sebebi, gerilim altındaki bir yalıtkan içinde dipollerin ortaya çıkmasıdır. Bir yalıtkanda dipoller karşılıklı yük etkileri, yalıtkanın iç kısmında birbirini yok ederken yalıtkanın yüzeyinde bir yüzeysel yük oluşmasına sebep olurlar. Bu yüzden elektrotların benzetiminden başka, yalıtkanlar arasındaki sınırların da ayrık yüklerle benzetiminin yapılması gerekir. Bunun daha önce göz önüne alınan durumlardan iki önemli farkı vardır.

1. Genel olarak, yalıtkanlar arası sınır bir eşpotansiyel yüzey değildir.

2. Denklem sisteminin oluşturulabilmesi için yalıtkanlar arası sınırın her iki tarafında da elektriksel alan hesaplanabilmelidir.

Yöntemin uygulanmasının daha kolay açıklanabilmesi için, Şekil (A.2)'de gösterildiği gibi az sayıda benzetim yükünün kullanıldığı iki yalıtkanlı basit bir örnek seçilmiştir.



Şekil –A.2 Çok yalıtkanlı bir sistemin benzetimi

Elektrot üzerinde n_E adet sınır noktası ve elektrot için aynı sayıda yük alınmıştır. Bunlardan n_{EA} adedi A yalıtkanı tarafındadır (1 numaralı sınır noktası) ve n_E-n_{EA}

adedi ise B yalıtkanı tarafındadır (2 ve 3 numaralı sınır noktaları). Burada n_E adet yük, her iki ortamdaki yani hem A yalıtkanındaki, hem de B yalıtkanındaki potansiyellerin ve elektrik alan şiddetlerinin hesabında kullanılır. Yalıtkanların arakesitinde n_B adet sınır noktası (4 ve 5 numaralı sınır noktaları) ile B yalıtkanında n_B adet yük (4 ve 5 numaralı yükler, bunlar A yalıtkanı için geçerlidir) ve A yalıtkanında n_B adet yük (6 ve 7 numaralı yükler, bunlar da B yalıtkanı için geçerlidir) vardır. Toplam olarak $n_T=n_E+n_B$ (=5) adet sınır noktası ve $n_q=n_E+2n_B$ (=7) adet yük vardır.

Benzetim yüklerinin belirlenmesi için gerekli denklem sistemi, sağlanması gereken sınır koşullarından yararlanılarak oluşturulur:

1. Elektrot ile A yalıtkanı arasındaki sınır noktasının (1 numaralı noktanın) potansiyeli V_e olmalıdır (V_e elektrot potansiyeli).

$$\sum_{j=1}^{n_E} p_j q_j + \sum_{j=n_E+1}^{n_E+n_B} p_j q_j = V_e \qquad \text{(A.10)}$$

(1...3) (4...5)

2. Elektrot ile B yalıtkanı arasındaki sınır noktalarında da potansiyel V_e olmalıdır (2 ve 3 numaralı sınır noktaları).

$$\sum_{j=1}^{n_E} p_j q_j + \sum_{j=n_E+n_B+1}^{n_E+2n_B} p_j q_j = V_e \qquad \text{(A.11)}$$

(1...3) (6...7)

3. Yalıtkanların arakesiti üzerindeki sınır noktalarında potansiyeller bilinmemektedir, fakat herbir sınır noktası için A yalıtkanındaki (V_A) ve B yalıtkanındaki (V_B) potansiyellerin eşit ($V_A=V_B$) koşulundan:

$$\sum_{j=1}^{n_E} p_j q_j + \sum_{j=n_E+n_B+1}^{n_E+2n_B} p_j q_j = \sum_{j=1}^{n_E} p_j q_j + \sum_{j=n_E+1}^{n_E+n_B} p_j q_j \qquad \text{(A.12)}$$

(1...3) (6...7) (1...3) (4...5)

yazılabilir. Bu da basitleştirilerek,

$$-\sum_{j=n_E+1}^{n_E+n_B} p_j q_j + \sum_{j=n_E+n_B+1}^{n_E+2n_B} p_j q_j = 0 \qquad \text{(A.13)}$$

(4...5) (6...7)

elde edilir.

4. p_j, verilen bir sınır noktasındaki potansiyele j. yükün etkisini göz önüne alan bir katsayıdır. Benzer şekilde f_j de j. yükün verilen bir sınır noktasındaki alan şiddeti vektörünün yalıtkan sınırına dik (normal) bileşenine katkısını gösterir. Bu durumda, yalıtkanların arakesitindeki sınır noktalarında B yalıtkanındaki alan şiddetinin normal bileşeni, A yalıtkanındakinden ε_B defa daha büyük olmalıdır. Buna göre:

$D_{nA}=D_{nB}\rightarrow\varepsilon_B E_{nB}=\varepsilon_A E_{nA}$

$$\varepsilon_B\left[\sum_{j=1}^{n_E} f_j q_j + \sum_{j=n_E+1}^{n_E+n_B} f_j q_j\right] = \varepsilon_A\left[\sum_{j=1}^{n_E} f_j q_j + \sum_{j=n_E+n_B+1}^{n_E+2n_B} f_j q_j\right] \qquad \text{(A.14)}$$

(1...3) (4...5) (1...3) (6...7)

veya

$$(\varepsilon_B - \varepsilon_A)\sum_{j=1}^{n_E} f_j q_j + \varepsilon_B \sum_{j=n_E+1}^{n_E+n_B} f_j q_j - \varepsilon_A \sum_{j=n_E+n_B+1}^{n_E+2n_B} f_j q_j = 0 \qquad \text{(A.15)}$$

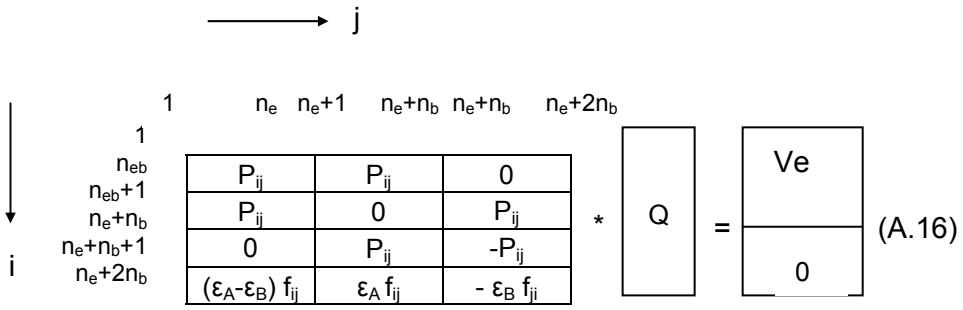
(1...3) (4...5) (6...7)

şeklide yazılabilir.

Bu örnekte, 4'ten 7'ye kadar olan yükler A yalıtkanı sınırı için kullanılmış iken 1'den 3'e kadar olanlar elektrotlar için kullanılmıştır. Yalıtkanların arakesitinde sadece iki sınır noktası (4 ve 5) seçilmişken, 1'den 3'e kadar olan sınır noktaları elektrot üzerinde seçilmiştir.

Verilen bir sınır noktasındaki denklemler yazılırken sınır noktasının üzerinde bulunduğu yalıtkan içindeki yükler göz önüne alınmaz. Örneğin, Şekil (A.2)'de 1 ile gösterilen sınır noktasındaki potansiyel sadece 1'den 5'e kadar olan yüklerin oluşturduğu potansiyellerin toplanmasıyla hesaplanır. Benzer şekilde, B yalıtkanı tarafından bakıldığında 5 ile gösterilen sınır noktasındaki potansiyel ve

elektriksel alan şiddeti de sadece 1'den 3'e ve 6'dan 7'ye kadar olan yüklerin oluşturduğu potansiyellerin toplamı ile hesaplanır. Böylece elde edilen n_E+2n_B (=7) adet lineer denklem aynı sayıda bilinmeyen yükün hesabı için kullanılır.



(A.16)

Bu örnekten de görüldüğü gibi, çok yalıtkanlı sistemlerin benzetimi tek yalıtkanlı sistemlere göre daha karmaşıktır. Bununla birlikte, yalıtkan sınırlar daha karmaşık bir yapıya sahip olduğu zaman böyle bir benzetimin doğruluğu da kötüleşmektedir.

A.5 Yük Benzetim Yönteminin Doğruluğu

A.5.1 Doğruluk Ölçütü

Yük benzetim yönteminde, benzetim yüklerinin sınır koşullarını yalnızca belirli sayıda sınır noktasında sağlaması gerekmektedir. Benzetimin doğruluğunu belirlemek için aşağıdaki ölçütlerden bir veya birkaçı kullanılabilir:

1. Herbir iletken sınırı için birkaç noktada potansiyel hatası hesaplanabilir. Bu hesaplamalarla, potansiyel hatasının ΔV_{ort} ortalama değeri, ΔV_{max} maksimum değeri veya $(\Delta V_{ort})^2$ karesel ortalama değeri bulunabilir.

2. Elektriksel alandaki hatanın genellikle potansiyel hatasından daha büyük olmasından dolayı, doğruluğu kontrol etmek için iletken sınırında (E_t/E_n) oranının çok küçük bir değerde olması önerilmektedir. Burada E_t ve E_n, elektriksel alan vektörünün sırasıyla teğetsel ve normal bileşenlerini göstermektedir. Bu bakımdan başka bir ölçüt, elektrot yüzeyinde elektriksel alan vektörünün normal doğrultusundan açısal sapmasıdır. Bu açıya "sapma açısı" denir. Sapma açısı, potansiyel hatasına göre benzetimin doğruluğunun daha duyarlı bir şekilde ortaya çıkmaktadır. Daha kuvvetli bir doğruluk ölçütü,

iletken yüzeyine dik elektriksel alanın türevinin yine alanın kendisine dik bölümünün, kontrol noktasındaki eğriliğe eşit olması gerektiğidir.

3. Çok yalıtkanlı sistemlerde, çözümün yapıldığı yalıtkan arakesitin her iki tarafındaki ortam için arakesitteki potansiyel çözümleri arasındaki fark olarak tanımlanan "potansiyel ayrılığı" küçük olmalıdır.

4. Yük benzetim yönteminin doğruluğunu kontrol etmek için kullanılan diğer bir yöntem, yük benzetim yöntemi sonuçları ile diğer sayısal, çözümsel veya örneksel yöntemlerle bulunan sonuçların karşılaştırılmasıdır.

5. Kapasite hesaplarında, C kapasitesinin değeri benzetim yükü sayısının işlevi olarak bulunabilir. Doğru bir benzetimde kapasitenin yük sayısından bağımsız olması gerekir.

A.5.2 Doğruluğu Etkileyen Etkenler

Yük benzetim yönteminde hata, benzetim yüklerinin ve sınır noktalarının yerleriyle, elektrotların ve yalıtkanların şekline bağlı olduğu kadar, benzetim yüklerinin tipine ve sayısına da bağlıdır.

Genellikle tek yalıtkanlı sistemlerde benzetim yükü sayısı arttırılarak potansiyel hatası azaltılabilir. Maksimum potansiyel hatası olarak en az % 10 – 3 mertebesinde hatalar elde edilebilmektedir. Bununla beraber, keskin köşeli elektrotlar ve ince elektrotlar için bu kadar az hatalı sonuçlar elde edebilmek zordur. Böyle durumlarda genel yaklaşım, potansiyel hatasını küçültmek ve keskin köşelerdeki sınır noktalarının sayısını arttırmaktır. Elektriksel alan hatasının potansiyel hatasından daha büyük olmasından dolayı, % 0,1'in altındaki potansiyel hatası değerleri kabul edilebilir.

Genellikle ardışık iki sınır notası arasındaki uzaklığın bir sınır noktası ve bu nokta ile ilgili yük arasındaki uzaklığı olarak tanımlanan f_a "uzaklık faktörü" benzetimin doğruluğunu oldukça etkiler. Bu faktörün 1 ile 2 arasında bir değere sahip olması gerektiği öne sürülmüştür. Pekçok çalışmada $1<f_a<2$ değeri kullanılarak tatmin edici doğrulukta benzetimler elde edilmiştir. Bazı çalışmalarda, $0,7<f_a<1,5$ değeri kullanılmıştır.

Çok yalıtkanlı sistemlerde eğer yalıtkan sınır karmaşık bir yapıya sahipse "potansiyel ayrılığı" genellikle büyüktür. Benzetim yüklerinin sayısını belirli bir

sınırdan öteye arttırmanın doğruluk üzerinde istenen bir etki yapmadığı öne sürülmüştür. Böyle durumlarda “Yüzeysel Yük Benzetim Yöntemi” nin veya “Birleşim Yöntemi”nin kullanılması daha yararlıdır.

EK B

EK B1

YÜKSEK GERİLİM İZOLATÖRLERİNDE DARBE ATLAMA GERİLİMİNİN BULUNMASINDA KULLANILAN YSA ALGORİTMASI

```
%%%%%%%%%%%GERİYE YAYILIM ALGORİTMASI%%%%%%%%%
%%MATLAB 5.0 NEURAL NETWORK TOOLBOX KULLANILARAK HAZIRLANMIŞTIR%%
%%%Aktivasyon fonksiyonu olarak her katmanda sigmoid fonksiyonu seçilmiştir%%%
clear;
ni=4;%giriş nöronlaarı sayısı
no=2;%çıkış nöronları sayısı
nts=13;%örüntü sayısı
load inp3.dat;%giriş örüntü vektörünün yüklenmesi
a=inp3;
load out3.dat;%çıkış örüntü vektörünün yüklenmesi
b=out3;

%%%%%%%%%%%%%%%normalizasyon%%%%%%%%%%%%%%%%
for p=1:ni
   input(:,p)=[a(:,p)/max(a(:,p))];
end

for q=1:no
   target(:,q)=[b(:,q)/max(b(:,q))];
end
x=input';%ysa giriş vektörü
t=target';%ysa beklenen değer vektörü
[r,q]=size(x);
s1=7;%gizli katman nöron sayısı
[s2,q]=size(t);
[w1,b1]=rands(s1,r);%birinci katman ağırlıklarının seçilmesi
[w2,b2]=rands(s2,s1);%ikinci katman ağırlıklarının seçilmesi
dispfreq=100;
maxepoch=5000;%iterasyon sayısı
errgoal=0.01;%sse bu değere ulaştığında eğitimi durdur
lr=0.5;%öğrenme hızı
mom=0.9;%momentum katsayısı
tp=[dispfreq maxepoch errgoal lr mom];
[w1,b1,w2,b2,epochs,tr]=trainbpm(w1,b1,'logsig',w2,b2,'logsig',x,t,tp);%eğitim
a1=logsig(w1*x,b1);%birinci katman (gizli katman) çıkışı
a2=logsig(w2*a1,b2);%ikinci katman (çıkış katmanı) çıkışı
ee= abs(t-a2);%ağ hatası
sse = sumsqr(ee)%toplam karesel hata (performans fonksiyonu)
rms=sqrt(sse/(no*nts))%hatanın etkin değeri
xnew=[0.3333 0.8394 0.5135 0.1]';%test kümesi (normalizasyonu yapılmış)
bc=logsig(w1*xnew,b1);
istdeg=logsig(w2*bc,b2);%%test kümesi uygulandığında ağın verdiği çıkış
for m=1:no
output(m,:)=[(istdeg(m,:)*(max(b(:,m))))];%%gerçek çıkış değeri (unnormalizing)
end
output
d=[out3(11,:)];
desiredoutput=d'
eee=output-desiredoutput;
abs=abs(eee);
mae=((abs(1)/output(1))+(abs(2)/output(2)))/no%test hatası ortalama mutlak hata
```

EK B2

ELEKTROT BİÇİM OPTİMİZASYONUNDA KULLANILAN YSA ALGORİTMASI

```
%%%%%%%%LEVENBERG-MARQUARDT ALGORİTMASI%%%%%%%%%
%%MATLAB 6.5 NEURAL NETWORK TOOLBOX KULLANILARAK HAZIRLANMIŞTIR%%
%%%Aktivasyon fonksiyonu olarak her katmanda sigmoid fonksiyonu seçilmiştir%%%
clear;
ni=7;%giriş nöronları sayısı
no=6;%çıkış nöronları sayısı
np=10;%%örüntü sayıdı
nhl=8;%%gizli katman nöron sayısı
load m2giris10.dat;%giriş vektörünün yüklenmesi
a=m2giris10;
load mikis10.dat;%çıkış vektörünün yüklenmesi
b=mikis10;

%%%%%%%%%%%%%%%normalizasyon%%%%%%%%%%%%%%%%
for pp=1:ni
   input(:,pp)=[a(:,pp)/max(a(:,pp))];
end

for q=1:no
   target(:,q)=[b(:,q)/max(b(:,q))];
end
x=input';
t=target';

net=newff([0 1;0 1;0 1;0 1;0 1;0 1;0 1], [nhl 6],{'logsig','logsig'},'trainlm');%eğitim
net.trainParam.epochs=1000;%iterasyon sayısı
net.trainParam.goal=0.0000000001;%hedeflenen ortalama karesel hata, bu değerde eğitim biter
net.trainParam.show=100;
net=train(net,x,t);

%%%%%%hatalar%%%%%%
output=sim(net,x);%ağ çıkışı
fark=t-output;
ee=abs(fark);
sse=sumsqr(ee);%toplam karesel hata
for k=1:no
   for l=1:np
      hata(k,l)=fark(k,l)^2;
      hata2(k)=sum(hata(:,l));
      rms=sqrt((sum(hata2))/(np*no));%hatanın etkin değeri
   end
end

%%%%%%%%%%test%%%%%%%%%%
xnew=[0.1561 0.1735 0.1854 0.2052 0.2123 0.2203 0.2350]'; %normalize edilmiş test kümesi
[u,w]=size(xnew);
nc=w;
output2=sim(net,xnew);
for m=1:no
realoutput(m,:)=[(output2(m,:)*(max(b(:,m))))];%%test kümesi uygulandığında ağın verdiği sonuç
end


desiredoutput=b(5,:)';
eee=realoutput-desiredoutput;%test kümesinin hata işareti

rms
maerror=mae(eee)%ortalama mutlak hata (test kümesi için)
desiredoutput%istenen sonuç
realoutput%ysa'nın verdiği sonuç
```

```
%%%%%%sonucunun bulunması istenen girişin ağa uygulanması%%%%%
alan=[10*ones(1,7); 12*ones(1,7); 24.4*ones(1,7)];
%%%istenen girişin normalizasyonu
for www=1:ni
   xtest(:,www)=[alan(:,www)/max(alan(:,www))];
end
y=sim(net,xtest');


for mm=1:no
realy(mm,:)=[(y(mm,:)*(max(b(:,mm))))];%% normalizasyonun tersi
end
realy%ağın verdiği sonucun gerçek değeri
```

EK B3

İZOLATÖR KONUM AÇISI OPTİMİZASYONUNDA KULLANILAN YSA ALGORİTMASI

```
%%%%%%%%%%%GERİYE YAYILIM ALGORİTMASI%%%%%%%%%
%%MATLAB 6.5 NEURAL NETWORK TOOLBOX KULLANILARAK HAZIRLANMIŞTIR%%
%%%Aktivasyon fonksiyonu olarak birinci katmanda sigmoid fonksiyonu, ikinci katmanda doğrusal
fonksiyon seçilmiştir%%%

clear;
ni=2;%number of the input neurons
no=1;%number of outputs
np=15;%%number pf the training patterns
nhl=10;%%number of hidden layer neurons
load epoksiavealan.dat;
load epoksiacilar.dat;
a=epoksiavealan;
b=epoksiacilar;
%%%%%%%%%%%%%%%%%normalization%%%%%%%%%%%%%%%%%
for p=1:ni
   input(:,p)=[a(:,p)/max(a(:,p))];
end
for q=1:no
   target(:,q)=[b(:,q)/max(b(:,q))];
end
x=input';
t=target';


net=newff(minmax(x), [nhl 1],{'logsig','purelin'},'traingdm');
net.trainParam.lr=0.4;
net.trainParam.mc=0.5;
net.trainParam.epochs=15000;
net.trainParam.goal=0.0001;
net.trainParam.show=1000;
net=train(net,x,t);
      Y = sim(net,x);

%%%%%eğitim hataları%%%%%%
fark=t-Y;
ee=abs(fark);
sse=sumsqr(ee);%toplam karesel hata
for k=1:no
   for l=1:np
      hata(k,l)=fark(k,l)^2;
      hata2(k)=sum(hata(:,l));
      rms=sqrt((sum(hata2))/(np*no));%hatanın etkin değeri
   end
end

%%%%%%%%%test%%%%%%%%%%
%a=26, e=u/a=0.7994, e2=0.5958, alfa1=45
x2=[26 0.5958];
%%%%%%%%%%%%%%%%%normalizasyon%%%%%%%%%%%%%%%%%
for pp=1:ni
   input2(:,pp)=[x2(:,pp)/max(a(:,pp))];
end
t2=[45];%beklenen çıkış
xnew=input2';
nc=1;%test kümesi sayısı
output=sim(net,xnew);
for m=1:no
realoutput(m,:)=[(output(m,:)*max(b(:,q)))];%%ysa'nın bulduğu açı
end
```

```
desiredoutput=45;
eee=realoutput-desiredoutput;
mutlak=abs(eee);

rms
maerror=mae(eee)
desiredoutput
realoutput
```

ÖZGEÇMİŞ

Suna Bolat, 1979 yılında İstanbul'da doğdu. İlk ve orta eğitimini sırasıyla Kazım Karabekir İlköğretim Okulu ve Kemal Hasoğlu Lisesi'nde tamamladı. 1996 – 2001 yılları arasında, bir yıl hazırlık sınıfı dahil, İstanbul Teknik Üniversitesi, Elektrik – Elektronik Fakültesi, Elektrik Mühendisliği Bülümü'nde tamamlayarak Elektrik Mühendisi ünvanını aldı. 2002 yılında, Bursa ELECO'2002, Elektrik Elektronik Bilgisayar Mühendisliği Sempozyumu'nda, "Yüksek Gerilim İzolatörlerinin Darbe Atlama Gerilimlerinin Yapay Sinir Ağı ile Belirlenmesi" başlıklı bildirisi yayınlandı. 2001 – 2002 döneminden beri, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Elektrik Mühendisliği Ana Bilim Dalı'nda lisans üstü eğitimini sürdürmektedir.