

93741

YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

KONİK KESİT FONKSİYONLU YAPAY SİNİR
AĞINDA ÖĞRENME ALGORİTMASININ
GELİŞTİRİLMESİ ve AĞIN ÇEŞİTLİ PROBLEMLER
İÇİN PERFORMANSI ile DUYARLIĞININ
İNCELENMESİ

Elektronik Yüksek Mühendisi Lale BAŞTÜRK ÖZYILMAZ

F.B.E. Elektronik ve Haberleşme Mühendisliği Anabilim Dalı Elektronik Programında Hazırlanan

DOKTORA TEZİ

Tez Savunma Tarihi : 7 Nisan 2000

Tez Danışmanı : Prof. B. Şefik SARIKAYALAR (YTÜ)

Jüri Üyeleri : Prof. Dr. Fikret GÜRGEN (BÜ)

Doç. Dr. Günhan DÜNDAR (BÜ)

[Handwritten signatures and initials]

İSTANBUL, 2000

KISALTMA LİSTESİ.....	i
ÖNSÖZ	ii
ÖZET	iii
ABSTRACT	iv
1. GİRİŞ	1
2. YAPAY SİNİR AĞLARININ GENEL YAPISI	5
2.1 Yapay Sinir Ağlarının Çalışması ve Çeşitli Özellikleri	5
2.1.1 Öğrenme	6
2.1.2 Genelleştirme	6
2.1.3 Çoklu işleme	7
2.2 İleri Beslemeli Ağlar	8
2.2.1 Tek katmanlı ileri beslemeli ağlar	8
2.2.1.1 Algılayıcı (Perceptron) öğrenme algoritması	8
2.2.2 Çok katmanlı ileri beslemeli ağlar	9
2.3 Çok Katmanlı Algılayıcı	10
2.3.1 Delta öğrenme kuralı	12
2.3.3 Genelleştirilmiş delta kuralı	12
2.3.4 Öğrenme faktörleri	15
2.3.4.1 Yakınsama	15
2.3.4.2 Ağırlıkların başlatılması	16
2.3.4.3 Öğrenme oranı	16
2.3.5 Eğitim zamanını azaltan yöntemler	16
2.3.5.1 Momentum yöntemi	16
2.3.5.2 Adaptif öğrenme oranı	17
2.4 Radyal Temelli Fonksiyon Ağları	17
2.4.1 RBF birimlerinin karakteristikleri	18
2.4.2 Temel fonksiyonların seçimi	19
2.4.3 RBF ağlarının eğitilmesi	20
2.4.3.1 RBF birim merkezlerinin belirlenmesi	20
2.4.3.2 Uzaklık ölçekleme parametresinin belirlenmesi	20
2.4.3.3 Birim ağırlıklarının belirlenmesi	20
2.4.4 RBF öğrenme algoritmaları	20
3. KONİK KESİT FONKSİYONLU YAPAY SİNİR AĞI	22
3.1 Konik Kesitler	23
3.2 Konik Kesit Fonksiyonlu Yapay Sinir Ağı	25
3.3 Konik Kesit Fonksiyonlu Yapay Sinir Ağının Eğitilmesi	27
3.4 Adaptif Öğrenmeli Konik Kesit Fonksiyonlu Yapay Sinir Ağı	28
3.4.1 Merkezlerin belirlenmesi için kullanılan Ortogonal En Az Kareler yöntemi	29
3.4.2 Ağırlıkların güncelleştirilmesi	33
3.4.3 Merkezlerin güncelleştirilmesi	36
3.4.4 Açılma açısının güncelleştirilmesi	36

3.4.5	Adaptif öğrenme oranı ve momentum	37
4.	KONİK KESİT FONKSİYONLU YAPAY SİNİR AĞININ ÇEŞİTLİ.....	39
	PROBLEMLER ÜZERİNDE İNCELENMESİ	
4.1.1	Elektromiyografik İşaretlerin Sınıflandırılması Problemi.....	39
4.1.2	Elektromiyografik işaret nedir?.....	39
4.1.3	EMG işaretlerin modellenmesi.....	40
4.1.4	AR katsayıları.....	40
4.1.5	Miyoelektrik işaretlerin yapay sinir ağlarına uygulanması.....	41
4.1.6	Miyoelektrik işaretleri sınıflamada kullanılan yöntem.....	44
4.1.7	Eğitme ve test sonuçları.....	44
4.2	Kontakt Lens Uydurma Problemi.....	48
4.3	İris Bitkisi Sınıflandırma Problemi.....	51
4.4	Fetus Gelişimi Problemi.....	58
4.5	Fonksiyon Yaklaşımı Problemi.....	62
4.6	Mikrodalga Transistörün YSA ile modellenmesi.....	65
4.7	Karakter Tanıma Problemi.....	66
5.	KONİK KESİT FONKSİYONLU YAPAY SİNİR AĞININ DUYARLIĞI.....	69
5.1	Yapay Sinir Ağlarında Duyarlık Analizi.....	69
5.2	Duyarlık Fonksiyonları.....	69
5.3	Konik Kesit Fonksiyonlu Yapay Sinir Ağı Çıkışlarının Girişlere Göre Duyarlığı.....	70
5.4	Duyarlık Analizinin Çeşitli Problemlere Uygulanması.....	72
5.4.1	Kontakt lens uydurma problemi için duyarlık analizi.....	72
5.4.2	İris bitkisi sınıflandırma problemi için duyarlık analizi.....	75
5.4.3	Fetus gelişimi problemi için duyarlık analizi.....	77
5.4.4	EMG işaretlerini sınıflandırma problemi için duyarlık analizi.....	78
5.4.5	Sonuçlar.....	78
6.	SONUÇLAR ve TARTIŞMA.....	80
6.1	Öğrenme Algoritmasının Geliştirilmesi.....	80
6.2	Ağın Performansının Değerlendirilmesi.....	80
6.3	Duyarlık Analizinin Değerlendirilmesi.....	83
6.4	Gelecek Çalışma.....	84
	YAYIN LİSTESİ.....	85
	KAYNAKLAR.....	86
Ek 1	EMG İşaretlerinin Sınıflandırılması Problemi İçin Veritabanı.....	90
Ek 2	Kontakt Lens Uydurma Problemi İçin Veritabanı.....	92
Ek 3	İris Bitkisi Sınıflandırma Problemi İçin Veritabanı.....	93
Ek 4	Fetus Gelişimi Problemi İçin Veritabanı.....	95
Ek 5	NE21900 Mikrodalga Transistörü İçin Veritabanı.....	99
Ek 6	Adaptif Öğrenmeli Konik Kesit Fonksiyonlu Yapay Sinir Ağı Eğitim Programı....	100
	ÖZGEÇMİŞ.....	107

KISALTMA LİSTESİ

AR	Auto Regressive
ARMA	Auto Regressive Moving Average
CSFNN	Conic Section Function Neural Network
EMG	Elektromiyografi
k-NN	k-Nearest Neighbor
k-NNFP	k-Nearest Neighbor on Feature Projections
MLP	Multilayer Perceptron
OLS	Orthogonal Least Squares
PI	Pulsatility Index
RBF	Radial Basis Function
RI	Resistance Index
S/D	Systolic/Diastolic ratio
UA	Umbilical Artery
WI	Week Index
YSA	Yapay Sinir Ağı



ÖNSÖZ

Bu tezin hazırlanmasında bana yol gösteren sayın hocam Prof. Şefik SARIKAYALAR'a, doktora çalışmamı tamamlamamda büyük katkısı olan, çok değerli görüş ve düşüncelerinden yararlandığım sayın Yrd. Doç. Dr. Tülay YILDIRIM'a en içten teşekkürlerimi sunarım.

Ayrıca moral destekleri için tüm çalışma arkadaşlarıma; gösterdikleri anlayış, sağladıkları ortam ve motivasyon için aileme sonsuz teşekkürler.

Bu çalışmamın benden sonraki lisansüstü öğrencilerine de yararlı olmasını temenni ederim.

İstanbul, 2000

Lale BAŞTÜRK ÖZYILMAZ

ÖZET

Konik Kesit Fonksiyonlu Yapay Sinir Ağında esas fikir, bir Çok Katmanlı Algılayıcı birimi ile bir Radyal Temelli Fonksiyonlu ağ birimi arasında ilişki sağlayarak bunlara ait karar bölgelerini içine alacak bir birimin fonksiyonunu genelleştirmektir. Bir yapay sinir ağının performansını belirleyen önemli parametreler vardır: Ağın gelecekteki performansı yani genelleştirme yeteneği, çok amaçlı kullanılıp kullanılamayacağı, ağ yapısının karmaşıklığı, elektronik tasarıma uygun olup olmadığı gibi. Bu çalışma üç ana konuda yoğunlaşmıştır: İlk olarak Konik Kesit Fonksiyonlu Yapay Sinir Ağı için daha önce kullanılan öğrenme algoritması geliştirilerek adaptif öğrenme sağlanmış, ayrıca momentum terimi eklenmiştir. Daha sonra ağın limitlerini ve farklı amaçlar için kullanımını belirleyebilmek için MATLAB kullanılarak geliştirilen algoritma çeşitli problemler üzerinde incelenmiştir. Bunlar elektromiyografik işaretlerin sınıflandırılması, kontakt lens uydurma, iris bitkisi sınıflandırma, fetus gelişimi, fonksiyon yaklaşımı, mikrodalga transistörün modellenmesi ve karakter tanıma problemleridir. Bu çalışma, Konik Kesit Fonksiyonlu Yapay Sinir Ağının değişik problemler üzerindeki performansının ve genelleştirme yeteneğinin, karşılaştırılan diğer ağ yapılarından daha üstün olduğunu göstermiştir.

Son olarak giriş verisinde meydana gelebilecek bir değişimden Konik Kesit Fonksiyonlu Yapay Sinir Ağı çıkışlarının nasıl etkileneceğini bulmak için ağ çıkışının girişlere göre duyarlılığının analizi yapılmıştır. Duyarlık analizi sonucu ağ çıkışında etkisi en az olan giriş, eğitime sırasında göz önüne alınmayabilir. Yani, az duyarlığa sahip olan giriş kanalı ağdan çıkartılabilir. Böylece ağ boyutu, dolayısıyla karmaşıklık ve eğitime zamanı azaltılmış olur. Bu da ağ performansının iyileşmesini sağlar. Konik Kesit Fonksiyonlu Yapay Sinir Ağı için duyarlık formülleri çıkartılarak değişik problemler için ağın duyarlılığı hesaplanmış ve sonuçlar verilmiştir. Duyarlık analizi ile çıkış üzerinde etkisi az olan girişin tespit edilip ağdan çıkartılabilmesi ile ağ boyutunun azaltılması elektronik ağ tasarımı açısından oldukça önemlidir.

ABSTRACT

The main idea of the Conic Section Function Neural Network is to provide a relationship between a Multilayer Perceptron unit and a Radial Basis Function Network unit and to generalize the function of a unit to include both of these units. There are important parameters which determine the performance of an artificial neural network: The future performance of the network; that is, the ability of generalization, whether or not it can be used for several purposes, the complexity of the network, the suitability for electronic implementations. This work is concentrated on three main subjects. First, the learning algorithm used for Conic Section Function Neural Network is developed and an adaptive learning is achieved and also momentum term is added. Then, an algorithm developed using MATLAB software is studied on various problems in order to determine the limits and the usability areas of network. These problems include the classification of electromyography signals, contact lens fitting, iris plant classification, fetal hypoxia, function approximation, modelling of microwave transistor and character recognition. This study showed that the performance and the generalization power of Conic Section Function Neural Network on several problems is more powerful than other network structures.

Finally, in order to determine how outputs of Conic Section Function Neural Network could be affected by the input data, a sensitivity analysis of network output depending on inputs is made. As a result, the input which has a minor effect on network output can be neglected during training. In other words, the input channels that produce low sensitivity can be removed from the network. This will reduce the size of the network which in turn reduces the complexity and the training time. This may also improve the network performance. The sensitivity analysis formulas for Conic Section Function Neural Network are obtained and the sensitivity of network is calculated and the results are given for various problems. Using sensitivity analysis, determination of the input which has minor effect on output and removal of this input from the network is very important to reduce the size of the network in electronic hardware implementations.

1. GİRİŞ

Genel anlamda yapay sinir ağı, beynin bir işi gerçekleştirebilme yolunu modelleyerek, bu modeli elektronik elemanlarla veya bilgisayarlarla gerçekleyen bir sistemdir. Yapay sinir ağının tanımı şöyle yapılabilir: Yapay sinir ağı, bilgiyi depolayabilen ve kullanıma uygun hale getirebilen, paralel işleyen işlemcilerden oluşmuş bir yapıdır.

Günümüzdeki bilgisayarlar kendileri için tanımlanmış komut dizilerini yani uzmanlık bilgilerini gerekli şartlar sağlandığı takdirde başarılı bir şekilde gerçekleştirirler. Belirsiz veya eksik bilgi verilmesi halinde sonuç büyük ölçüde etkilenir. Alt kademelerde oluşacak sorunlar sistemin çalışmamasına veya hatalı sonuçlar vermesine neden olur. Bilgisayarlar gerekli koşullar sağlandığında insanların çok uzun sürelerde yapabilecekleri işleri çok kısa sürelerde gerçekleştirmelerine ve insan beynine göre çok hızlı ve güvenilir olmasına rağmen insan beyninin rahatlıkla yaptığı görme, algılama, ses tanıma, şekil tanıma ve nicesel yargı (sağ duyu) gibi işleri yapamamaktadır. Söz konusu yeteneklere yapay zeka yöntemleriyle ulaşılmaya çalışılmış fakat doyurucu sonuçlar elde edilememiştir. Bu nedenle bilim adamları insan beynini incelemeye yönelmişler ve yapay sinir sistemleri adı altında bir bilim dalı ortaya çıkmıştır.

Yapay sinir ağı teknolojisi günümüzde endüstriden ekonomi dünyasına, tıp biliminden askeri eğitim çalışmalarına kadar bir çok alanda kullanılmaktadır. Bu teknoloji problemlerin çözümüne yeni bir boyut kazandırmıştır. Yapay sinir ağları insan vücudundaki sinir sisteminin bazı fonksiyonlarını modelleyen ve bazı yeteneklerini yakalamak isteyen basit hesapsal birimlerin (nöronlar) yoğun bir paralel dizisidir; başka bir deyişle, teorik hale getirilmiş zeka ve beyin faaliyetlerinin matematiksel modelleridir. Ancak biyolojik sistemler o kadar karmaşıktır ki yapay sinir ağı için kullanılan biyolojik modellerin fazlaca basite indirgenmiş biçimleri şeklindedir. Yapay sinir ağları, kesin kurallarla gösterimi zor olan, daha çok algılamaya yönelik bilgileri işlemekte kullanılırlar. Olayları genelleştirme yetenekleri ve eksik, belirsiz, bozulmuş bilgileri işleyebilme ve esnek olarak çalıştırabilmeleri önemli özelliklerindendir. Bu ağlarda kullanılan basit işleme elemanları insan beyninin işleme elemanı olan nöronların modelidir. İnsan sinir sisteminin problemleri çözebilmek için öğrenme özelliği olduğu gibi yapay sinir ağlarının da bu özelliği mevcut bulunmaktadır. Yapay sinir ağlarında bilgi saklama verilen eğitime özelliğini kullanarak eğitime örnekleriyle yapılır. Burada bilgisayarlardaki programlama ihtiyacının yerini öğrenme kuralları ve algoritmaları alır. Sinir ağı kullanıcıları klasik bilgisayar programcıları gibi belirli bir

algoritma seçmezler, onun yerine en iyi mimariyi ve öğrenme şeklini seçerler. Daha sonra uygun girişler sisteme verilir, sistemde bu bilgiyi işleyerek sonuca ulaşır.

Yapay sinir ağları konusundaki ilk çalışmalar 1940 yılına dayanmaktadır. Bu konudaki çalışmalar psikolojik teoriler ve nöropsikolojik araştırmalarla paralel olarak ilerlemiştir. Nöral hesaplama teknikleri hakkında ilk çalışmaları başlatan McCulloch ve Pitts (1943) ilk temel hesap yapan nöron modeli tasarlamayı başarmıştır. Geliştirdikleri model basit tipten sinir ağlarının aritmetik ya da lojik fonksiyonları gerçekleştirebilme yeteneğine sahipti. Fakat bu model vakum tüpleri çağı için uygulanabilir bir özellik taşııyordu dolayısıyla teknik olarak kesin bir sonuca ulaşamadı. D. Hebb 1949 yılında, "Hebbian öğrenme kuralı" adında ilk öğrenme şeklini geliştirdi. Hebb bilginin nöral ağlarda depolanabileceğini iddia etmiş ayrıca öğrenme tekniğı ile ilgili teoriler ortaya atmıştır. 1950'lerde ilk nörobilgisayarlar inşa edilmiş ve 1954'de Minsky tarafından test edilmiştir. 1958'de Rosenblatt, nöron benzeri "Perceptron" denen elemanları ortaya koymuştur. İlerki yıllarda Minsky'nin nöral ağların yeteneklerini gösteren bazı iyimser iddiaları ortaya atmasıyla yapay sinir ağlarına ilgi 1960'lı yıllarda hızla artmıştır. Bu yıllarda "ADALINE" ismi verilen bir cihaz tasarlandı (ADaptive LINEar combiner). 1960-1962 yıllarında Widrow-Hoff öğrenme kuralı geliştirildi. Bu kural toplam karesel hatayı daha aza indirgemıştır. Minsky'nin daha sonraki çalışmalarında yaklaşım ile ilgili umduğu sonuçları elde edememesi bu konudaki çalışmalara olumsuz yönde etkili olmuştur. O yıllarda henüz çok katmanlı ağlar konusunda çalışmalar yapılmıyordu. Tek katmanlı ağlarında bazı basit işlemleri gerçekleştirememesi yapay sinir ağlarına olan ilginin azalmasına neden olmuştur. 1969 yılında Minsky ve Papert, tek katmanlı ağlarla ayrıcalıklı-veya (XOR) işleminin yapılamayacağını ortaya koymuştur. 1965'ten 1984'e kadar olan yıllar arasında birçok araştırmacı bu konuda çalışmalarda bulunmuştur. Ağırlıklı toplama elemanlarının öğrenmeleri ile ilgili matematiksel teoremler Sun-Ichi Amari (1972-1977) tarafından geliştirilmiştir. Ayrıca Japonya 'da Kunihiro Fukushima neocognitrons adı verilen yeni bir sinir ağı mimari şekli geliştirmiştir. 1974-1982 yılları arasında Grossberg ve Carpenter, sinir ağları ile ilgili mimari şekilleri ve teoremler dizisi ortaya atmıştır. (Hecht, 1991; Harvey, 1994; Haykin, 1994; Zurada, 1995; Schalkoff, 1997)

Çok Katmanlı Algılayıcı (Multilayer Perceptrons-MLP) ve Radyal Temelli Fonksiyonlu (Radial Basis Function-RBF) yapay sinir ağları örüntü tanıma, sınıflandırma problemleri gibi birçok alanda kullanılabilir. Birçok araştırmacı bu ağların eğitme algoritmalarının geliştirilmesi için değişik çalışmalar yapmıştır. MLP için en popüler öğrenme algoritması olan hatanın geriye yayılması 1986 yılında McClelland ve Rumelhart tarafından geliştirilmiştir.

RBF ağlarının eğitilmesi için değişik metodlar kullanılmaktadır (Moody ve Darken, 1989; Chen vd., 1991). RBF ve MLP ağlarında en uygun eğitme algoritmasının hangisi olduğu konusunda araştırmalar devam etmektedir. Genel durumda birbirinden farklı propagasyon kuralı, aktivasyon fonksiyonu ve öğrenme kuralına sahip olan MLP ve RBF ağlarının çeşitli kombinasyonlarla birleştirilmesine çalışılmıştır. Maruyama, Girosi ve Poggio radyal temelli fonksiyon olarak sigmoid kullanarak genelleştirilmiş RBF ve MLP arasında bir bağlantı sağlamışlardır (Maruyama vd., 1992). (Tarassenko ve Roberts, 1994) bir RBF sınıflayıcıyı hatanın geriye yayılması ile eğitmişlerdir. (Hirahara ve Oka, 1993), MLP ve RBF için iki ayrı modül kullanmış ve bunları bir lineer birleştiriciden geçirerek hibrid bir model geliştirmişlerdir. 1992'de hem kapalı hem açık karar sınırlarını birleştiren projeksiyon sinir ağı önerilmiştir (Wilensky ve Manukian, 1992). Diğer araştırmacılar da (Tsoi, 1989; Geva ve Sitte, 1992; Smyth, 1992; Weymaere ve Martens, 1994) RBF/MLP hibrid ağları ve her iki ağ için değişik öğrenme algoritmaları geliştirmişlerdir. Dorffner MLP ve RBF'i bir ağda birleştiren Konik Kesit Fonksiyonlu Sinir Ağını tanıtmıştır (Dorffner, 1994; Yıldırım ve Marsland, 1997).

Bu tez Konik Kesit Fonksiyonlu Yapay Sinir Ağı'nın performansının belirlenmesi ve iyileştirilmesi konusunda yoğunlaşmıştır. Bu amaçla adaptif bir öğrenme algoritması geliştirilmiş ve momentum teriminin etkisi incelenmiştir. Algoritma farklı problemler üzerinde denenerek sonuçlar verilmiştir.

Bölüm 2'de yapay sinir ağlarının performansını belirleyen genel özelliklerden bahsedilmiş ve ileri beslemeli ağlara kısaca değinilerek MLP ve RBF ağlarının genel yapısı verilmiştir. Bu ağlara ait teorik eşitlikler ve öğrenme algoritmaları da tanımlanmıştır. Bölüm 3'de Konik Kesit Fonksiyonlu Yapay Sinir Ağı için geliştirilen adaptif öğrenme algoritması sunulmuştur.

Bölüm 4'de geliştirilen algoritma çeşitli problemler üzerinde incelenmiştir. Bunlar: Elektromiyografi (EMG) işaretlerinin sınıflandırılması, kontakt lens uydurma, iris bitkisi sınıflandırma, fetus gelişimi, fonksiyon yaklaşımı, mikrodalga transistörün modellenmesi ve karakter tanıma problemleridir. MATLAB kullanılarak her problem için simülasyon sonuçları sunulmuş ve başka ağ yapıları ile bu sonuçlar karşılaştırılmıştır.

Bölüm 5'de giriş verisinde meydana gelebilecek bir değişimden Konik Kesit Fonksiyonlu Yapay Sinir Ağı çıkışlarının nasıl etkileneceğini bulmak için ağ çıkışının girişe göre duyarlılığı incelenmiştir. Bu amaçla CSFNN için duyarlılık formülleri çıkartılarak, değişik problemler için

ağın duyarlılığı hesaplanmıştır. Farklı problemler için duyarlılığı az olan girişlerin ağdan çıkartılması ile yeniden eğitilen ağa ait sonuçlar verilmiştir.

Sonuç olarak Bölüm 6'da çalışma süresince elde edilen sonuçlar yorumlanarak, yapılan çalışmanın neler kazandırdığı vurgulanmış gelecekte yapılabilecekler hakkında öneriler sunulmuştur.



2. YAPAY SINİR AĞLARININ GENEL YAPISI

Bir yapay sinir ağının (YSA) temel elemanı nöronlar yani işlem birimleridir. Bir yapay sinir ağı işlem birimleri (elemanları) ile bağlantı geometrisinden oluşan eylem birimi ile bunlara ait bir graf biçimindedir. Her düğüm hücre denilen n. dereceden doğrusal olmayan (nonlinear) dinamik bir devredir.

YSA'larının elektrik devreleri bakımından tanımını özet olarak aşağıdaki gibi sıralanabilir:

- YSA'nın düğümleri birer işlem elemanıdır.
- Düğümler arasındaki bağlantılar tek yönlü işaret iletim yollarıdır.
- Her işlem elemanı istenildiği kadar girişe bağlanabilir.
- Her işlem elemanının tek bir çıkışı vardır. Fakat bu çıkışlar kopyalanabilir.
- İşlem elemanları yerel belleklere sahip olabilirler.
- Her işlem elemanının bir transfer (aktivasyon) fonksiyonu vardır. Bu fonksiyon, işlem elemanının çıkış değerini oluşturur. Transfer fonksiyonları sürekli veya ayrık olarak çalıştırılabilirler.
- YSA aynı transfer fonksiyonuna sahip alt gruplara ayrılabilir. Bunlara katman denir.
- Pek çok YSA giriş katmanı içerir ancak bu katmanın transfer fonksiyonu yoktur.

2.1 Yapay Sinir Ağlarının Çalışması ve Çeşitli Özellikleri

Yapay sinir ağlarında her işleme elemanının bir yerel belleği vardır, bu bellekte uyarlanan katsayıların önceki hesaplamalar ile belirlenen değerleri saklanır. Her bir işleme elemanının çıkışı, son gelen giriş işaretlerinin ve yerel bellekteki adaptif katsayıların bir fonksiyonu olarak belirlenir. Bu fonksiyon transfer fonksiyonu olarak adlandırılır. Nöral ağın tasarımına bağlı olarak, işleme elemanları ya sürekli olarak çalışırlar ya da ayrık zamanda güncelleştirilirler.

Her işleme elemanı komşularında yapılan işlemlerden bağımsız olarak çalışır. Ancak bir işleme elemanının çıkışı diğer elemanlara da giriş olduğu için, tüm işleme elemanları bütün ağın davranışını gizli bir şekilde etkiler. İşleme elemanlarının arasındaki bağlantıların topolojisi, her işleme elemanının hangi bilgiyi aldığını ve dolayısıyla üzerinde çalışabileceği bilgiyi belirlediği için, bir sinir ağının yürütebileceği tüm bilgi işleme fonksiyonlarını etkiler. (Hecht, 1991; Hertz vd., 1991)

2.1.1 Öğrenme

Öğrenme kuralı, giriş işaretlerine ve transfer fonksiyonu tarafından sağlanan değerlere cevap olarak yerel bellekteki ağırlıkların hepsini veya bazılarını değiştiren eşitlik olarak bilinir ve giriş işaretlerinin doğasına bağlı olarak, işleme elemanının cevabının değişmesini sağlar. Bu şekilde ağ kendisini istenen cevaplara göre ayarlayabilir ve kendi içinde bilgiyi düzenleyebilir yani kısaca öğrenebilir.

Yapay sinir ağları programlama yerine örneklerle eğitilir. Çocukların annelerini tanımaları için karşılaştırmalı fizyoloji hakkında hiçbir şey bilmelerine gerek olmadığı gibi, programlayıcılar da sinir ağlarına tanınacak cisimlerin nicel tanımlarını veya söz konusu cisimleri benzer cisimlerden ayırmak için lojik kriter kümeleri sağlamak zorunda değillerdir. Bunun yerine bir sinir ağına bazen tanımları ile beraber, cisim örnekleri de girilir. Ağ, ağırlık matrisindeki değerleri değiştirerek bunları öğrenir ve ağa bir giriş uygulandığı zaman o girişe uygun çıkış cevabı üretir.

Yapay sinir ağları eğiticili ve eğiticişiz olmak üzere iki şekilde eğitilmektedirler. Eğiticili öğrenmede ağa hem giriş hem de istenen çıkış bilgisi (hedef vektörü) girilir. Her denemeden sonra ağ kendi çıkışını doğru cevaplarla karşılaştırır ve çıkış hatası kabul edilebilecek seviyeye ininceye kadar ağırlıklarını değiştirerek tekrarlama yapar. Eğiticişiz öğrenmede hiçbir hedef vektörü yoktur. Giriş vektörü sisteme uygulanır ve sistem, girişin benzer veya ayrılan özelliklerinden yararlanarak uyumlu bir çıkış (muhtemelen eğitimden önce tahmin edilemeyen) üretecek şekilde kendisini organize eder. Böyle sistemler daha çok sınıflama ve kümeleme problemleri için kullanılmışlar ve özellikle Kohonen ile Grossberg tarafından geliştirilmişlerdir. (Lippman, 1987; Hinton, 1989)

2.1.2 Genelleştirme

Alışlagelmiş bilgisayarlar, istenen çıkışı üretmek için tam olarak doğru girişlere ihtiyaç gösterirler. Yapay sinir ağları ise girişlerinde değişimler olsa bile doğru çıkışı üretebilirler. Örneğin, basılı harfleri tanımak için eğitilmiş bir sistem, gürültü giriş karakterlerinin %40 oranında bozulduğu zaman bile doğru olarak çalışabilir. Yani, sistem daha önceden o tipten hiçbir şey görmemiş olsa da, aynen insanların tam olmayan veya kısmen hatalı girişlerde yaptığı gibi harfleri tanıyabilmektedir. Çalışmalar, insanların çoğunun bir metindeki harflerin yarısından fazlası silinmiş olsa bile, o metni okuyabildiklerini göstermiştir.

Gerçek dünya bir bilgisayar programının istediği doğrulukta bilgiyi çok ender olarak sağlayabilir. Bilgisayarlar gürültülü girişler göz önüne alınarak programlanabilirler, ancak getirilen ek hesapsal yük çoğu kez bu algoritmaların pratik uygulamalarda kullanılmasını engellemektedir. Yapay sinir ağları bu engellemeyi ayrıntılı programlamadan ziyade (ki bu yol uygulamalara bağlı olurdu), yapılarından dolayı başarırlar.

Bir yapay sinir ağının geliştirilmesindeki en kritik parametrelerden biri genelleştirme, yani ağın gelecekteki performansdır: Ağın, eğitim kümesinde mevcut olmayan durumlar için ne kadar iyi tahminlerde bulunabildiğinin belirlenmesidir. Öğrenme süresince, eğiticili bir sinir ağının çıkışları eğitim kümesindeki girişleri verilen hedef değerlere yaklaştırır. Bu yetenek tek başına yararlı olabilir; fakat, bir sinir ağını kullanmanın amaçlarından biri genelleştirme yapmaktır. Yani, ağın çıkışlarını, eğitim kümesinde verilmeyen girişler için de hedef değerlere yaklaştırmaktır. Genelleştirme her zaman mümkün olmayabilir. Tipik olarak iyi bir genelleştirme için üç koşul gereklidir (fakat yeterli olmayabilir): 1) Ağa uygulanan girişlerin hedefe ait yeterli bilgiyi içermesi. 2) Girişleri doğru çıkışlara bağlayan yani öğrenmeyi sağlayan fonksiyonun yumuşak geçişli olması. Başka bir deyişle girişlerdeki küçük bir değişik çıkışlarda da küçük bir değişiklik üretmelidir. 3) Eğitim durumlarının yeterince geniş ve kullanılan alt kümelerin (istatistiksel terminolojide "örnekler") genelleştirilmesi istenen (istatistiksel terminolojide "popülasyon") bütün durumları temsil etmesi. (Schalkoff, 1997; Sarle, 1999)

2.1.3 Çoklu işleme

Birçok bilgisayar tasarımcısı, bugünün en hızlı bilgisayarının bile tek bir işlemcinin teorik hız limitinin onda birinde çalıştığının farkındadır. Bu nedenle çoklu işlemeyi gerçekleştirmek amacı ile bir problemi her birini ayrı işlemcilere verebilecek alt problemlere bölme çabası gösterilmektedir. Bu yaklaşım iki temel zorluk getirmektedir. Birincisi, birçok problem tamamen paralel olarak çözülemez. Mantıksal kısıtlamalar programın bir kısmının veya hepsinin seri olarak hesaplanmasını gerektirmektedir ki bu durumlarda bin işlemcili bir sistemin, 999 tanesi zamanın çoğunda işsiz kalabilir. İkincisi, çoklu işlemciler için bir optimal plan bulmak amacıyla, en kötü durumda tüm olasılıkları denemeye dayalı bir plandan daha iyi bir sonuç verecek hiçbir algoritma bilinmemektedir.

Doğaları gereği, yapay sinir ağları kendi kendilerini düzenlerler; yani her düğüm, sistem içindeki diğer tüm işlemcilerden bağımsız olarak kendi girişleri üzerinde çalışan bir işlemci

olarak görülebilir. Ağ bir çözüme yakınsarken, tüm işlemciler çalışmaktadır. Buna ek olarak sistemin yapısını yeniden düzenlemeden problem boyutuna uyması için modüler bir şekilde yeni işlemciler yapıya eklenebilir. (Zurada, 1995)

2.2 İleri Beslemeli Ağlar

İleri beslemeli ağlar, işaretin ağ girişinden çıkışına doğru tek yönlü olarak iletildiği ağlardır. Bir katmandaki hücreler, ancak bir önceki katmanın çıkışlarını giriş olarak alabilir. İleri beslemeli ağlarda girişlerin ve çıkışların geciktirilmişleri bulunmaz. (Treleaven vd., 1989; Warwick vd., 1992; Ruth, 1994)

x_i girişine ilişkin ağırlık w_i ve b de eşik değeri olsun. s ağırlıklı toplam olmak üzere aktivasyon ifadesi ve aktivasyon fonksiyonu aşağıdaki gibi yazılabilir:

$$s = \sum_{i=1}^N w_i \cdot x_i + b \quad (2.1)$$

$$y = f(s) \quad (2.2)$$

$f(s)$ fonksiyonu olarak değişik fonksiyonlar kullanılmakta ve hücreler bu fonksiyonlara göre adlandırılmaktadır.

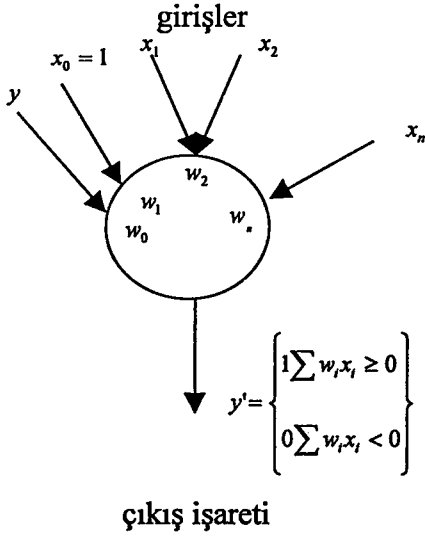
2.2.1 Tek katmanlı ileri beslemeli ağlar

Genelde tek bir hücre istenilen giriş çıkış işlevini yerine getiremez. Bunun için bağlantı ağırlıkları dışında aynı özelliklere sahip hücreler bir araya getirilerek katmanlar oluşturulur. Buradaki tek katman terimi, nöronların çıkış katmanına karşı düşmektedir. Herhangi bir hesaplamanın yapılmadığı giriş katmanı göz önüne alınmaz. Bu tür ağlar algılayıcı (perceptron) olarak bilinir. (Hertz vd., 1991; Haykin, 1994; Zurada, 1995)

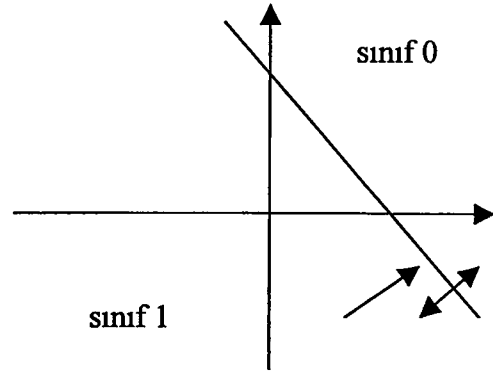
2.2.1.1 Algılayıcı (Perceptron) öğrenme algoritması

1958 yılında Rosenblatt tarafından önerilen algılayıcı öğrenme algoritmasında ağırlıkların değişimi, çıkış ile istenen çıkışın farkı ile orantılıdır. Dolayısıyla eğitici bir öğrenme algoritmasıdır. Algılayıcı, Şekil 2.1'de gösterilen bir veya daha çok işlem elemanından

meydana gelen basit bir yapay sinir ağıdır. Şekil 2.2’de ise algılayıcının iki sınıfı birbirinden nasıl ayrıldığı gösterilmektedir.



Şekil 2.1 Algılayıcı



Şekil 2.2 n boyutlu uzayda sınıfların dağılımı

Algılayıcıda kullanılan transfer fonksiyonunun ifadesi aşağıdaki gibidir:

$$y = \sum_{i=1}^n w_i x_i \quad (2.3)$$

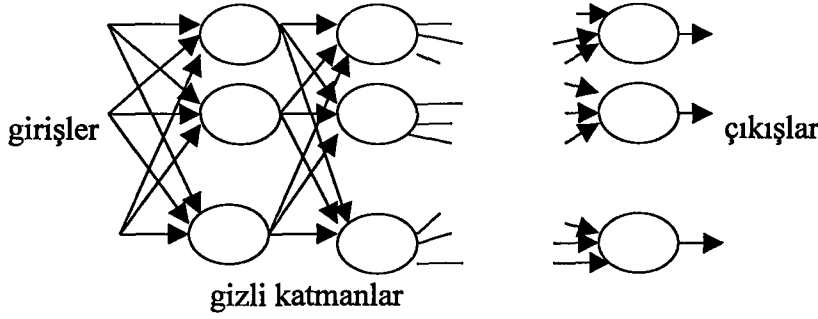
$$w_i(k+1) = w_i(k) + \eta [d_i(k) - \text{sgn}(w_i(k) x(k))] x(k) \quad (2.4)$$

(2.4) eşitliğinde yer alan x_i , giriş vektörünün i . elemanını temsil ederken w_i , yerel belleğin i . elemanını, d_i hedef çıkışı temsil etmektedir. η , öğrenme oranını belirleyen 0 ile 1 arasında pozitif sayıdır. Devrenin çıkışı, iki kutuplu ve ağırlıkların oluşturduğu bir hiper yüzey tarafından belirlenmektedir. Ağın ürettiği çıkış istenen çıkıştan farklı ise, hiper yüzey hatayı azaltacak yönde hareket etmektedir. Hata sıfır olduğunda w değişimi durmaktadır. (Haykin, 1994; Bose ve Liang, 1996; Gurney, 1996)

2.2.2 Çok katmanlı ileri beslemeli ağlar

Çok katmanlı ileri beslemeli ağlar, tek katmanlı ağların arka arkaya bağlanmasından oluşurlar. Bir katmanın girişi, bir önceki katmanın çıkışıdır. Şekil 2.3’de çok katmanlı ileri beslemeli ağın genel yapısı verilmiştir. Giriş katmanı sadece girişi çoğullamaktadır yani girişi ile çıkışları aynıdır. Giriş ile çıkış arasında kalan katmanlara gizli katmanlar adı verilmektedir. Genelde bir veya iki gizli katman kullanmak yeterli olmaktadır. Gizli katmanlardaki hücre

sayıları farklı olabilir. Çok katmanlı ağlar karmaşık karar yüzeylerine sahip sınıfların ayrılmasında kullanılırlar. (Haykin, 1994; Zurada 1995)



Şekil 2.3 Çok katmanlı ileri beslemeli ağ

2.3 Çok Katmanlı Algılayıcı

Bir Çok Katmanlı Algılayıcı (Multi Layer Perceptron - MLP) yapısı birçok birimin (perceptron) oluşturduğu bir kümedir. MLP'de algılama birimi olarak perceptron kullanılmıştır. Bu algılayıcı birimler bir araya gelerek ağdaki katmanları oluşturur. Bu katmanlar da bir araya gelerek ağı oluşturur. MLP'de üç temel katman vardır. Bunlar giriş katmanı, gizli katman ve çıkış katmanıdır. Giriş ve çıkış katmanları dışındaki bütün katmanlar gizli katman olarak adlandırılır. MLP birçok zor ve farklı problemin çözümünde başarıyla uygulanabilmektedir. Genelde ağın eğitiminde eğitici yöntem kullanılmaktadır. En yaygın kullanılan algoritma hatanın geriye yayılımı algoritmasıdır. Bu algoritma tespit edilen hatanın azaltılabilmesi için parametrelerde yapılacak olan değişikliğin bir artışı mı yoksa bir azaltmayı mı gerektirdiğinin belirlenmesi kuralına dayanır. Bu kural hata doğrulama öğrenme kuralı olarak da bilinir.

Temelde hatanın geriye yayılımı algoritması, ağdaki katmanlardan işaret akışının iki sefer sağlanmasına dayanır. İşaret akışı, önce bütün katmanlar üzerinden ileriye doğru ve sonra geriye doğrudur. İşaretin ileri akışında ağın giriş katmanına bir giriş vektörü uygulanır. Bu girişin etkisi katmandan katmana geçerek çıkışa ulaşır. Sonuçta bu giriş vektörü için bir çıkış vektörü elde edilir. İleri akışta sinaptik ağırlıklar sabit olup herhangi bir değişikliğe uğramazlar. İşaretin geriye akışında hata doğrulama kuralına göre ağırlıklar tekrar ayarlanır. Ağın giriş vektörüne olan cevabı istenen çıkış vektöründen çıkarılır. Böylece hata işareti elde edilmiş olur. Bu hata işareti ağ boyunca sinaptik ağırlıklarda yapılacak ayarlamayı belirlemek üzere geriye doğru yol alır. 'Hatanın geriye yayılımı' ifadesi buradan gelmektedir. Sinaptik ağırlıklarda yapılan ayarlamalar ağın cevabının istenen cevaba yaklaşmasını sağlayacak

yöndedir. Hatanın geriye yayılımı algoritması literatürde geriye yayılım algoritması (Back propagation) olarak adlandırılır. Bu algoritma yardımıyla gerçekleştirilen öğrenme de geriye yayılım öğrenmesi olarak adlandırılır.

Bir Çok Katmanlı Algılayıcı ağı üç ayrı özelliğe sahiptir:

1. Ağdaki her bir sinir hücresi girişlerin ağırlıklı toplamını doğrusal olmayan bir fonksiyona tabi tutarak çıkışına aktarır. Burada vurgulanması gereken önemli bir nokta bu fonksiyonun Rosenblatt'ın önerdiği algılayıcıdan farklı olarak her noktada türevi alınabilir bir fonksiyon olmasıdır. Genellikle, bu türevi alınabilirlik özelliğini sağlamak amacı ile eğrisel bir fonksiyon ile tanımlanan sigmoid fonksiyonu kullanılır.

$$y_j = \frac{1}{1 + \exp(-net_j)} \quad (2.5)$$

Burada net_j , j . sinir hücresine ait ağırlıklı toplamı yani aktivasyon değerini ve y_j , j . sinir hücresinin çıkışını ifade eder.

2. Ağ bir ya da daha fazla sayıda gizli katman içerir. Her bir gizli katman bir önceki katmandan, daha anlamlı sonuç alır ve bu sonuçları daha da anlamlandırarak bir sonraki katmana iletir. Böylece giriş vektörüne giriş katmanının cevabı katmandan katmana istenen çıkışa gittikçe yaklaştırılarak aktarılır.

3. Ağda sinir hücreleri çok yoğun bir şekilde birbirleriyle bağlanmışlardır. Ağ bağlantısındaki ya da bir hücrenin girişindeki herhangi bir ağırlıkta meydana gelecek en ufak bir değişim bütün sinaptik bağlantıların ya da ağırlıkların değişimini gerektirir.

Her bir gizli katmandaki ya da çıkış katmanındaki sinir hücresi şu iki hesaplamayı yapacak şekilde tasarlanmışlardır:

1. Bir sinir hücresinin çıkışında görülen fonksiyon işaretinin hesaplanması. Bu fonksiyon işareti sinir hücresindeki giriş işaretlerinin ağırlıklı toplamının sürekli bir fonksiyonudur.
2. Gradyen vektörünün o anki tahmini değerini hesaplamak. Gradyen vektörü, sinaptik ağırlıkların bir fonksiyonu olan hata performans yüzeyinin dip noktasına yani optimum çözümün bulunduğu noktaya ulaşmak için algoritmanın sinaptik ağırlıklarda yapılmasını belirlediği artırımların oluşturduğu vektördür. (Hush vd., 1992; Hush ve Horne, 1993; Haykin, 1994; Bose ve Liang, 1996)

2.3.1 Delta öğrenme kuralı

Delta öğrenme kuralı, Çok Katmanlı Algılayıcıların eğitilmesinde sıkça karşılaşılan geriye yayılım öğrenme algoritmasında kullanılan bir kuraldır. Herhangi bir katmandaki j . birime gelen net giriş, bir önceki katmandaki birimlerin y_i çıkışlarının ağırlıklı toplamıdır.

$$net_j = \sum_{i=1}^n w_{ji} y_i \quad (2.6)$$

İşlem elemanı, bu toplam girişi, türevi alınabilir, sürekli ve lineer olmayan bir fonksiyondan geçirir. Bu amaçla, en çok kullanılan fonksiyon sigmoid (lojistik) fonksiyonudur (Eşitlik 2.5). Bu fonksiyon, y çıkışını 0 ve 1 değerleri arasında sınırlar. Geriye yayılım öğrenme algoritmasında girişe uygulanan her bir giriş vektörüne karşı bir çıkış vektörü elde edilir. Bu çıkış vektörü, sisteme daha önceden verilen bir hedef vektörü ile karşılaştırılır. Bu iki vektör arasındaki fark yeterince küçük olacak şekilde ağırlıklar ayarlanır.

Eğitme iki adımda olur. Birinci adımda, ağa bir giriş vektörü uygulanır ve her işlem elemanının çıkışı hesaplanır. Elde edilen çıkışlar bir sonraki denemede ağırlıkların ayarlanmasında kullanılmak üzere saklanır. İkinci adımda, çıkış ve hedef vektörü arasındaki farka göre hesaplanan çıkış hatası geriye doğru iletilir. Ağ ağırlıkları bu hataya göre ayarlanır. Bu işlem, yeterince doğru cevaplar alınıncaya kadar tekrar edilir.

2.3.3 Genelleştirilmiş delta kuralı

Lineer fonksiyonlar için kullanılan delta kuralı lineer olmayan aktivasyon fonksiyonları için genelleştirilebilir. (Rumelhart vd, 1987; Zurada; 1995)

p giriş paterni için ağırlıklardaki değişim

$$\Delta_p w_{ji} = \eta (t_{pj} - O_{pj}) i_{pi} = \eta \delta_{pj} i_{pi} \quad (2.7)$$

ile verilir. Burada t_{pj} , p . patern için hedef çıkışın j . bileşenini, O_{pj} p . patern için aktif çıkışın j . bileşenini, i_{pi} giriş paterninin i . elemanının değerini ve δ_{pj} hata işaret terimini temsil eder. η orantı sabiti, öğrenme oranıdır. Genellikle 0 ile 1 arasında seçilir. Bu sabit ne kadar büyük ise

ağırlıktaki değişim o kadar büyüktür. Geriye yayılım algoritmasında kullanılan hata kriteri karesel ortalama hatadır. Buna göre p . patern için oluşan hata,

$$E_p = \frac{1}{2} \sum_j (t_{pj} - O_{pj})^2 \quad (2.8)$$

ve tüm paternler için toplam hata

$$E = \sum_p E_p \quad (2.9)$$

ile ifade edilir. Sistemde ağırlık değişimi, oluşan hataya göre yapıldığı için $\partial E_p / \partial w_{ji}$ ifadesi incelenmelidir. Bu inceleme ağız gizli katmana sahip olup olmamasına bağlı olarak değişik sonuçlar verir.

A-Yapay sinir ağızda gizli katman yoksa, zincir kuralına göre hatanın ağırlıklara göre değişimi;

$$\frac{\partial E_p}{\partial w_{ji}} = \frac{\partial E_p}{\partial O_{pj}} \cdot \frac{\partial O_{pj}}{\partial w_{ji}} \quad (2.10)$$

şeklinde iki ayrı türevin çarpımı olarak yazılabilir. Ağ çıkışı

$$O_{pj} = \sum_i w_{ji} i_{pi} \quad (2.11)$$

eşitliğinden görüldüğü gibi girişlerin ağırlıklı toplamıdır. Buna göre

$$\frac{\partial O_{pj}}{\partial w_{ji}} = i_{pi} \quad (2.12)$$

olarak bulunur. Eşitlik (2.10)'daki çarpımın birinci terimi (2.7) eşitliğinden yararlanarak

$$\frac{\partial E_p}{\partial O_{pj}} = -\delta_{pj} \quad (2.13)$$

şeklinde yazılabilir. Sonuç olarak p . patern için toplam karesel hatanın ağırlığa göre değişimi aşağıdaki gibi verilir:

$$\frac{\partial E_p}{\partial O_{pj}} = -\delta_{pj} i_{pi} \quad (2.14)$$

B-Yapay sinir ağında gizli katman varsa, yine zincir kuralı kullanılarak hatanın ağırlıklara göre değişimi;

$$\frac{\partial E_p}{\partial w_{ji}} = \frac{\partial E_p}{\partial net_{pj}} \cdot \frac{\partial net_{pj}}{\partial w_{ji}} \quad (2.15)$$

ile ifade edilir. Eşitliğin sağ tarafındaki kısmi türevler ayrı ayrı hesaplanacak olursa

$$net_{pj} = \sum_i w_{ji} O_{pi} \quad (2.16)$$

ifadesinden aktivasyonun ağırlıklara göre değişimi

$$\frac{\partial net_{pj}}{\partial w_{ji}} = O_{pi} \quad (2.17)$$

olarak bulunur.

$$O_{pj} = f_j(net_{pj}) \quad (2.18)$$

ve

$$\delta_{pj} = \frac{-\partial E_p}{\partial net_{pj}} \quad (2.19)$$

olmak üzere hata işaret terimleri:

a- O_{pj} yapay sinir ağı çıkışı ise

$$\delta_{pj} = (t_{pj} - O_{pj}) f'(net_{pj}) \quad (2.20)$$

b- O_{pj} gizli katman çıkışı ise

$$\delta_{pj} = f'(net_{pj}) \sum_k \delta_{pk} w_{pk} \quad (2.21)$$

ile ifade edilir. Sonuç olarak bu ifadeler (2.7) eşitliğinde yerine konularak gerekli ağırlık değişimi yapılır. Pratikte osilasyona yol açmayacak büyüklükte bir öğrenme oranı seçilir. Öğrenme oranının daha büyük değerlerine imkan sağlayarak, hata yakınsamasını hızlandırmak için genelleştirilmiş delta kuralı momentum terimi ile modifiye edilir.

$$\Delta w_{ji}(t+1) = \eta \delta_{pj} O_{pi} + \alpha \Delta w_{ji}(t) \quad (2.22)$$

2.3.4 Öğrenme faktörleri

Geriye yayılma algoritmasının bazı dezavantajları vardır. Bunlar; eğitimin yavaş olması, algoritmanın yerel minimuma yakalanabilmesi ve otonom olmamasıdır. Geriye yayılım algoritmasında ardarda iki katmandaki n tane birim için, n^2 tane ağırlık eğitilmektedir. Hatanın geriye yayılımı da bu ağırlıklar üzerinden yapılır. Bu işlemler oldukça zaman alır. Eğitim zamanını etkileyen faktörler ise ilk ağırlıklar, öğrenme oranı, gizli katman sayısı ve gizli katmandaki hücre sayısıdır. Hatayı minimum yapacak ağ ağırlıkları, bir gradyent arama tekniği kullanılarak bulunur. Eğitme sırasında yerel minimuma yakalanma sorunu vardır. Bu durumda ağ kabul edilemeyecek bir çözüme yakınsar. Bu sorundan kaçınmak için öğrenme oranı azaltılabilir. Ağa yeni bilgiler öğretildiğinde eskiden öğrenilen bilgilerin unutulması nedeniyle; ağ, değişen çevre şartlarına uyum sağlanmasının beklendiği durumlar için uygun değildir.

2.3.4.1 Yakınsama

$E=E(w)$ hata fonksiyonu, ağın tüm ağırlıklarını içeren bir vektör w olmak üzere $q+1$ boyutlu uzayda bir yüzey tanımlar. $E(w)$ 'nın tek global minimumu ve birçok yerel minimumu olabilir. Pratikte çok katmanlı ağların, eğitim sırasında yerel minimuma takılması durumu ile ender olarak karşılaşmaktadır. Eğer bir yerel minimuma ulaşırsa, eğitim, bir kaç yeni hücre ekleyerek yine rastgele başlatılmalı yada yerel minimum, sığ ise ağırlıklara rastgelelik eklenmelidir. İdeal durumda hatanın sıfır olması veya sıfırdan büyük bir global minimuma

ulaşılması beklenir. Ancak bu, pratikte sonsuz zaman gerektirebilir. Bunun yerine öğrenmenin kabul edilebilir bir $E_{\max}$ değerinin altındaki hatalar için başarılı olduğu kabul edilir.

2.3.4.2 Ağırlıkların başlatılması

Eğitimin başında seçilecek ağırlıklar, ağırlık istenen sonuca yakınsamasını önemli ölçüde etkiler. Başlangıçta, eğitilecek ağırlıklar, rastgele küçük değerler olarak seçilir. Eğer büyük değerlerde seçilirse ağırlıklı toplam da büyük değerler alır. Ancak çıkış fonksiyonunun bu noktadaki değeri çok küçük olur. Ağırlık değişiminde bu türev çarpım olarak geldiği için, her adımdaki değişimler de çok küçük olacaktır. Bu da eğitim zamanını arttıracaktır. Aynı sebepten dolayı girişler için küçük değerler alınır. Genelde ağırlıklar $[-1, +1]$ aralığında seçilir. Girişler ise $[-1, +1]$ aralığına ötelenerek kullanılır veya normalize edilir.

2.3.4.3 Öğrenme oranı

Geriye yayılma algoritmasının yakınsaması öğrenme oranına çok bağlıdır. Genel olarak, öğrenme oranının en iyi değeri çözülecek probleme bağlıdır. Teknik literatürde, değişik problemler için $[0, 1]$ aralığındaki öğrenme oranı değerlerinin başarıyla uygulanabildiğini gösteren çalışmalar vardır. Büyük öğrenme oranı değerleri, hata yüzeyinde büyük adımlar atılmasını sağlasa da ağırlık kararsız olmasına veya osilasyon yapmasına sebep olabilir. Bu da ağırlık bir minimuma yakınsamasına engel olabilir. Küçük değerler ise, bu problemi gidermesine rağmen eğitimin çok uzamasına sebep olur.

2.3.5 Eğitim zamanını azaltan yöntemler

Geriye yayılma algoritması ile eğitim uzun zaman almaktadır. Eğitim zamanını azaltabilmek için pek çok çalışmalar yapılmış ve bazı yöntemler üretilmiştir (Zurada, 1995; Bose ve Liang, 1996; Hagan vd., 1996). Bu yöntemlerden bazıları aşağıda verilmiştir.

2.3.5.1 Momentum yöntemi

Momentum yöntemi, yakınsama hızını artırmaya ek olarak, sık yerel minimumlardan kurtulmayı da sağlayabilmektedir. Bu yöntem, o anki ağırlık değişimine, daha önceki ağırlık değişiminin belli bir oranını eklemeyi içerir. Buna göre o anki ağırlık değişimi aşağıdaki gibi yazılabilir.

$$\Delta w(k) = -\alpha \nabla E(k) + \beta \Delta w(k-1) \quad (2.23)$$

Burada α öğrenme oranıdır. β ise momentum oranı olarak adlandırılan pozitif seçilen bir sabittir. Tipik olarak genelde 0.1 ile 0.9 arasında seçilir. (2.23) eşitliğindeki ikinci terim son ağırlık değişiminin bir oranını gösterir ve momentum terimi olarak adlandırılır. (Kröse ve Smagt, 1993; Moreira ve Fiesler, 1995; Zurada, 1995)

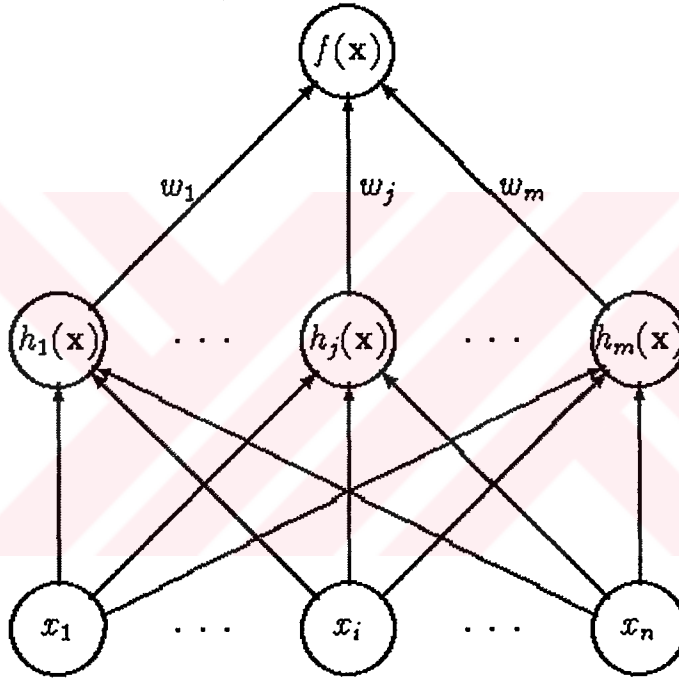
2.3.5.2 Adaptif öğrenme oranı

Adaptif öğrenme oranı, eğitim zamanını azaltan bir yöntemdir. Bir çevrimdeki hata $E(k)$, bir önceki çevrimdeki hata $E(k-1)$ 'den küçükse, α belirli bir sabitle (mesela 1.05 ile) çarpılarak artırılır. $E(k)$, $E(k-1)$ 'in belirli bir katından (mesela 1.04'den) büyükse yeni hesaplanan ağırlıklar ve hata iptal edilir ve α belirli bir sabitle (mesela 0.7 ile) çarpılarak azaltılır. Adaptif öğrenme oranı sayesinde ağ, makul olan en büyük α değerleri ile eğitilmiş olur. Bu sayede eğitim zamanı da azaltılır.

2.4 Radyal Temelli Fonksiyon Ağları

Biyolojik organizmaların sinir sistemlerinde, giriş uzayının bazı bölgelerine "lokalleşmiş" veya "ayarlanmış" cevap karakteristiklerine sahip nöronların olduğu gözlenmiştir. Örneğin görsel korteksin yöne duyarlı hücrelerinin cevabı retinadaki lokal bölgelere duyarlıdır; aynı şekilde kulaktaki *stereocilia* hücreleri frekansa cevap verecek şekilde lokal olarak ayarlıdır. Radyal Temelli Fonksiyon (Radial Basis Function-RBF) ağları biyolojik sinir ağlarının lokal olarak ayarlı hücrelerinden esinlenerek ortaya atılmıştır (Moody ve Darken, 1988; Moody ve Darken, 1989; Schalkoff, 1997). Bu ağlar, yüksek oranda lineer olmayan parametrelere sahip yapay sinir ağları için uygulanabilir bir alternatiftir. RBF ağları radyal olarak simetrik fonksiyonların lineer bir birleşimi olarak giriş-çıkış ilişkilerini modelleyerek fonksiyon yaklaşımı yaparlar (Broomhead ve Lowe, 1988; Poggio ve Girosi, 1989; Poggio ve Girosi, 1990). Fonksiyonel kestirim data analizi ve patern tanımda önemli bir problemidir. Ayrıca bu ağlar, ses bilgisi tanıma, el yazısı karakter tanıma, dijital sınıflama ve zaman serilerinin kestirimi gibi öğrenilmesi güç giriş-çıkış ilişkileri için başarılı bir şekilde kullanılırlar. Orijinal RBF yöntemi veri noktaları kadar RBF merkezinin varlığını da gerektirmektedir, bu ise kullanılan veri noktası sayısının genellikle büyük olmasından ötürü pratikte sinyal işleme uygulamalarında nadir olarak kullanılmaktadır.

Bir RBF ağı, bütün RBF merkezlerini ve lineer olmayan özellikleri gizli katmanda tutmak suretiyle parametre açısından lineer olan özel bir iki katmanlı ağ olarak ele alınabilir. Temel olarak, lokal olarak ayarlanmış veya lokal olarak duyarlı birimleri içeren tek bir gizli katman ve çoğu durumlarda lineer birimler içeren çıkış katmanından oluşur. Gizli katman ayarlanacak herhangi bir parametre olmaksızın sabit bir lineer olmayan dönüşümü gerçekleştirmektedir. Gizli katman birimlerinde, birim cevabı lokalleşmiştir ve girişlerin birimin alıcı alan merkezinden uzaklığının bir fonksiyonu olarak azalır. Çıkış katmanında ise lineer bir birleştirici işlem gerçekleştirmekte ve ayarlanacak yegane parametreler lineer birleştiricinin ağırlıkları olmaktadır. Şekil 2.4 bir RBF ağının genel yapısını göstermektedir. (Orr, 1996)



Şekil 2.4 RBF ağı

2.4.1 RBF birimlerinin karakteristikleri

Gizli katman, giriş uzayının belli bir bölgesi için cevap oluşturmaktadır. Yani giriş, giriş uzayının belli bir bölgesine düştüğünde çıkış sıfırdan farklıdır. Girişin bu küçük bölge dışında kalması durumunda çıkış sıfırdır. Bu nedenle gizli katman, giriş uyarısına yerel bir cevap veriyor denir. Girişten gizli katmana olan bağlantılar ağırlıksızdır; yani, her bir gizli düğüm her bir giriş değerini değişmeksizin alır. Gizli katmanın gerçeklediği fonksiyona temel fonksiyon denir. Temel fonksiyon olarak herhangi bir çekirdek (kernel) fonksiyon kullanılır.

Girişler belli bir merkez değere yaklaştıkça çıkışın değeri artmaktadır. Bu merkez değerden uzaklaştığında ise çıkışın değeri azalmaktadır. Başka bir deyişle bunlar tek maksimumlu fonksiyonlardır. Gauss dağılımı, çekirdek fonksiyonlara örnek olarak verilebilir. Çıkış katmanı, gizli katmandan gelen girişlerin lineer bir kombinasyonudur. En yaygın kullanılan temel fonksiyon Gauss fonksiyonudur.

$$u_j = \exp - \left[\frac{(x - c_j)^T (x - c_j)}{2\sigma_j^2} \right] \quad j = 1, 2, \dots, N_1 \quad (2.24)$$

Burada u_j birinci katmandaki j . düğüme karşılık gelen çıkış, x giriş paterni, c_j j . düğüm için Gauss fonksiyonunun merkezidir; σ_j^2 j . düğüm için uzaklık ölçekleme parametresi başka bir deyişle aktivasyon fonksiyonları için pozitif değerli şekillendirme parametresidir ve N_1 birinci katmandaki düğümlerin sayısıdır. Gizli katmandaki düğümlerin çıkışları, sıfırdan bire kadar değişen bir bölge içerisinde değer alır. Böylece giriş Gauss fonksiyonunun merkezine ne kadar yakın ise düğümünün çıkış değeri de o kadar büyüktür, başka bir deyişle 1'e o kadar yakındır. *Radyal temelli fonksiyon* ismi çekirdek fonksiyonlarının radyal olarak simetrik olmasından gelmektedir. Yani bu, her bir düğümün çekirdeğin c_j merkezinden sabit bir mesafedeki girişler için aynı sonucu üretmesi demektir. Çıkış katmanına ait düğümler gizli katmandan gelen çıkışların doğrusal bir kombinasyonudur. Çıkış düğümü eşitlikleri ise aşağıdaki formda verilebilir.

$$y_j = w_j^T u_1 \quad j = 1, 2, \dots, N_2 \quad (2.25)$$

Burada w_j , gizli düğümlerden çıkış düğümüne olan bağlantı ağırlığı ve u_1 'de birinci katmandan gelen çıkış vektörleridir. N_2 , çıkış katmanındaki düğümlerin sayısıdır.

2.4.2 Temel fonksiyonların seçimi

Bir RBF ağındaki lineer olmayan yapı birkaç adet lineer olmayan tipik temel fonksiyondan seçilebilir. Teorik araştırmalar ve pratik sonuçlar, bu yapı ile ilgili seçimin performans açısından kritik olmadığını göstermiştir. (Chen vd., 1991). Gizli katman düğümlerinin yeterli sayıda olması halinde parametrelerin ayarlanması ile lineer olmayan fonksiyonlara yaklaşım yapılabilir.

2.4.3 RBF ağlarının eğitilmesi

Bir RBF ağının eğitilmesi, RBF birim merkezlerinin, gizli katmandan çıkış katmanına olan ağırlıkların ve σ uzaklık ölçekleme parametresinin belirlenmesi ile yapılır.

2.4.3.1 RBF birim merkezlerinin belirlenmesi

RBF ağının performansı kritik olarak seçilen merkezlere bağlı bulunmaktadır. Alıcı alanların merkez koordinatlarının belirlenmesi için çeşitli metodlar kullanılmaktadır. Örneğin eğitme kümesindeki her bir giriş vektöründe bir merkez yerleştirilebilir. Fakat bu durumda gerekenden çok daha fazla küme ve gizli katman nöronu oluşabileceğinden bu yöntem pek uygun değildir. Pratikte merkezler verilerin bir alt kümesi olarak seçilir. Bu seçim yapılırken gizli düğümlerin sayısı bütün giriş uzayını kaplayacak yeterlikte olmalıdır. Küme merkezlerini bulmak için en iyi yaklaşımlardan biri K-ortalama kümeleme algoritmasıdır. Bu algoritmaya göre giriş bilgilerinin yoğun olduğu yerlerde merkezlerde yoğun bir şekilde dağıtılır. (Haykin, 1994; Middleton, 1997)

2.4.3.2 Uzaklık ölçekleme parametresinin belirlenmesi

Uzaklık ölçekleme (veya genişlik) parametresi σ , alıcı bölgelerin çapını belirleyen bir büyüklüktür. Merkezler birbirine yakınsa σ 'nın değeri küçük olmalı, birbirinden uzaksa boşluğu kapatacak kadar büyük olmalıdır. Genelde kullanılan yöntem, kümeleme merkezleri ve eğitme kümesindeki örnekler arasındaki ortalama mesafeyi almaktır. Bu parametre P-en yakın komşuluk yönteminden yararlanarak belirlenebilir (Cichocki ve Unbehauen, 1993).

2.4.3.3 Birim ağırlıklarının belirlenmesi

Eğitmenin bu son adımında Lineer En Küçük Kareler (Linear Least Squares) yöntemi kullanılarak hatayı istenen bir değere azaltacak şekilde gizli katmandan çıkışa olan ağırlıklar belirlenir.

2.4.4 RBF öğrenme algoritmaları

RBF ağlarında öğrenmeye ilişkin bir çok yaklaşım mevcuttur. Bunlardan bir çoğu öğrenme işini iki kısma ayırır. Buna göre ilk öğrenme işlemi gizli katmanda gerçekleşir. Daha sonra

öğrenme çıkış katında devam eder. Gizli katmandaki öğrenme eğiticişiz öğrenme algoritmalarından biri kullanılarak yapılır. Çıkış katmanındaki öğrenme ise eğiticişli öğrenmedir. Bazı uygulamalarda eğiticişli öğrenme algoritması çıkış katmanışyla birlikte gizli katmana da uygulanır. Böylece ağı parametrelerinde daha optimal bir uyum sağlanır. Gizli katmana uygulanabilecek çeşitli kümeleme algoritmaları vardır. Bunlardan en popüleri K-ortalamaşlı kümelemedir. Bu algoritma basit ve iyi sonuçlar veren bir algoritmadır. Uzaklık ölçekleme parametresi, σ , kümeleme algoritması son bulunduğunda hesaplanır. Çıkış katmanında öğrenme temel fonksiyonlardaki parametreler belirlendikten sonra yani gizli katmandaki öğrenme tamamlandıktan sonra sağlanır. Çıkış katında öğrenme Lineer Ortalamaşlı En Küçük Kareler (Linear Means Squares- LMS) algoritması kullanılarak yapılır.

RBF ağı için geliştirilen çeşitli öğrenme algoritmaları aşığıda kısaca özetlenmiştir (Chen vd., 1993) :

- 1) Sabit merkezlerle En Küçük Kareler yöntemi: RBF merkezleri giriş bilgisinden rasgele seçilir. Eğitim seti problemi iyi temsil edecek şekilde seçilirse bu yöntem iyi sonuç vermektedir. Merkezler belirlendikten sonra en küçük kareler yöntemi ile ağırlıklar eğiticişli modda belirlenir.
- 2) Ortogonal En Küçük Kareler yöntemi: Bu algoritmada uygun RBF merkezlerinin ve ağırlıkların belirlenmesi eş zamanlı olarak yapılır. Bu prosedür radyal temelli fonksiyon merkezlerini uygun bir ağı ortaya çıkana kadar rasyonel bir biçimde tek tek seçmektir.
- 3) İteratif kümeleme ve En Küçük Kareler yöntemi: Bu algoritmada RBF merkezleri bir iteratif kümeleme algoritması kullanılarak ayarlanır ve ağırlıklar iteratif en küçük kareler yöntemi ile güncelleştirilir. Burada merkezlerin belirlenmesi eğiticişiz olarak yapılır.
- 4) Dinamik komplekslik öğrenme algoritması: Bu iteratif öğrenme yönteminde, her yeni bir temel fonksiyonla önceki arasında oluşturulan bir açığı değerine ve kestirim hatasına bağılı olarak ağı yeni bir temel fonksiyon eklenir.

3. KONİK KESİT FONKSİYONLU YAPAY SİNİR AĞI

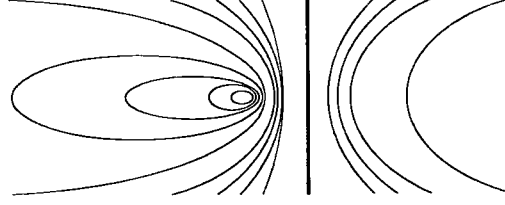
Yapay sinir ağlarının sınıflandırılması birçok yol izlenerek yapılabilir. Örneğin, sınıflama ve nörokontrol söz konusu ise lokal ve global genellemenin bir ayrım kriteri olarak kullanılması faydalı olacaktır. Ağın ağırlıklarından bir veya daha fazlası giriş uzayındaki bütün noktalar için ağ çıkışını etkileyebiliyorsa global genelleme söz konusudur. Lokal genellemede ise ağın ağırlıklarından birkaçı giriş uzayının lokal bir bölgesindeki noktalar için ağ çıkışını etkiler. MLP ağları, global olarak genelleme yapan ağlardır. Global genellemede öğrenme yavaştır. RBF ağları ise lokal olarak genelleme yapan ağlara örnektir.

Yapay sinir ağları kullanılarak yapılacak sınıflama işi, geometrik terimler kullanılarak tarif edilebilir. Eğer, girişler ayrılabilir iki sınıfa ait ise, yani girişler bir hiperdüzlemin iki tarafında yer alıyorsa, bu durumda algılayıcı yaklaşımı yöntemi başarılı olacak ve hiperdüzlemin bu iki sınıf arasında yerleşmesini sağlayacaktır. Eğer ayrılacak girişler, giriş uzayının belli bölgelerinde kümelenmişlerse bu durumda bir veya daha fazla gizli katmana sahip bir MLP ağı problemi çözecektir. Sınıflama işleminde karşılaşılabilecek diğer karar bölgeleri ise lokal olanlardır. Buna örnek olarak hiperküresel karar bölgeleri verilebilir. Giriş uzayının sonsuz bir kısmını içeren hiperdüzlemsel (açık) karar bölgelerinden farklı olarak, hiperküresel (kapalı) karar bölgelerinde her bir birimin alıcı (receptive) alanı lokaldir ve giriş uzayının küçük bir bölümü ile sınırlıdır. Bir RBF ağı ile hiperküresel karar sınırlarını elde etmek mümkündür.

Konik Kesit Fonksiyonlu Yapay Sinir Ağı (Conic Section Function Neural Network-CSFNN) ilk önce (Dorffner, 1994; Dorffner ve Porenta, 1994) tarafından tanımlanmıştır. Bu ağı, hiperdüzlem (düz çizgi) ve hiperkürelerin (daire) konik kesit fonksiyonlarının özel durumları olmasını esas alan bir sinir ağı yapısıdır. Bunlar, sırasıyla, Çok Katmanlı Algılayıcı (MLP) ve Radyal Temelli Fonksiyonlu (RBF) ağların karar sınırlarıdır. Bu iki durumun arasında yine karar bölgesi olarak geçerli olan, bazıları sınırsız ve sonsuz, bazıları sınırlı ve lokal olan elipsler, hiperboller ve paraboller gibi karar sınırları da vardır. Konik Kesit Fonksiyonlu Yapay Sinir Ağında esas fikir, bir Çok Katmanlı Algılayıcı birimi ile bir Radyal Temelli Fonksiyonlu ağı birimi arasında ilişki sağlayarak bütün bu karar bölgelerini içine alacak bir birimin fonksiyonunu genelleştirmektir. Konik Kesit Fonksiyonlu Yapay Sinir Ağı, giriş uzayını, verilen girişleri farklı sınıflara kolayca ayırabilecek lokalleşmiş bölgelere ayırır ve kapalı (hiperküre) karar sınırlarını açık (hiperdüzlem) olanlara veya açıkları kapalıya çevirebilir ve bu bölgeleri uygun olduğu yerlerde kullanabilir. Bölgenin tipi verilen uygulamadaki veri dağılımına bağlıdır.

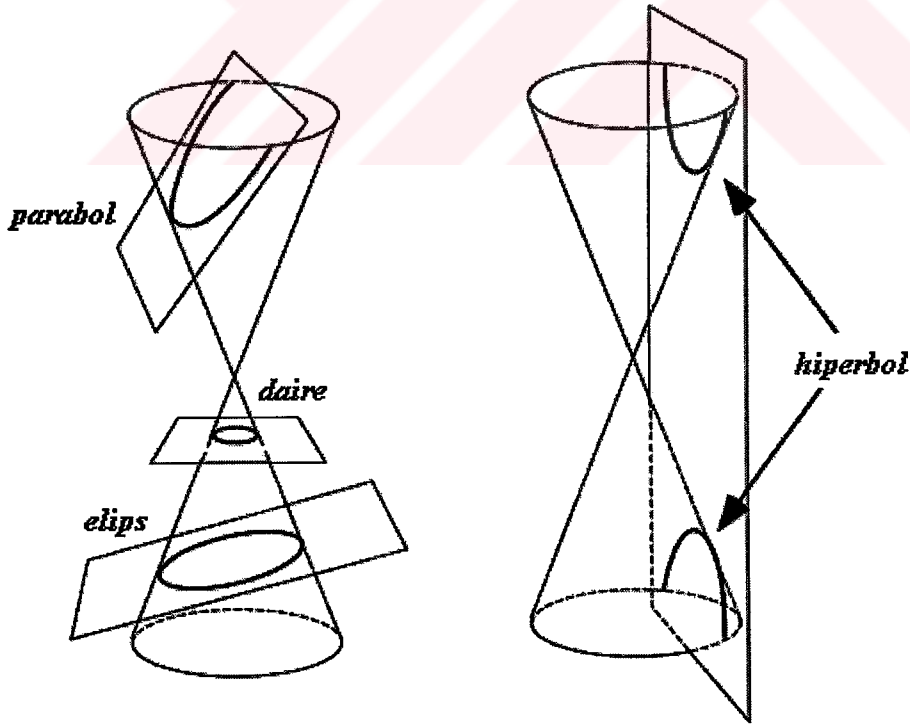
3.1 Konik Kesitler

Matematiksel olarak konik kesitler, bir düzlemde sabit bir noktaya (odak) olan mesafeleri ile bir doğruya (doğrultman) olan mesafelerinin oranlarının birbirine eşit olduğu noktaların yer eğrisidir. Eğrinin şekli bu oran ile belirlenir. Bu oran koninin dış merkezliği olarak adlandırılır ve e ile gösterilir. $0 < e < 1$ ise konik kesit bir elips, $e = 1$ ise parabol ve $e > 1$ ise hiperboldür. e 'nin aldığı değerlere göre karşılık gelen eğriler Şekil 3.1'de görülmektedir. (Ogilvy, 1969)



Şekil 3.1 Değişik dış merkezliklere göre konik kesitler

Konik kesitler, bir düzlem ve bir koninin kesiştirilmesi ile oluşan arakesitlerdir. Bir koninin bir düzlemle kesişmesi sonucunda oluşan ara kesit eğrileri Şekil 3.2'de verilmiştir. Bu şekiller koninin eksenleri ile düzlemin yaptığı değişik açılara bağlı olarak elde edilirler.

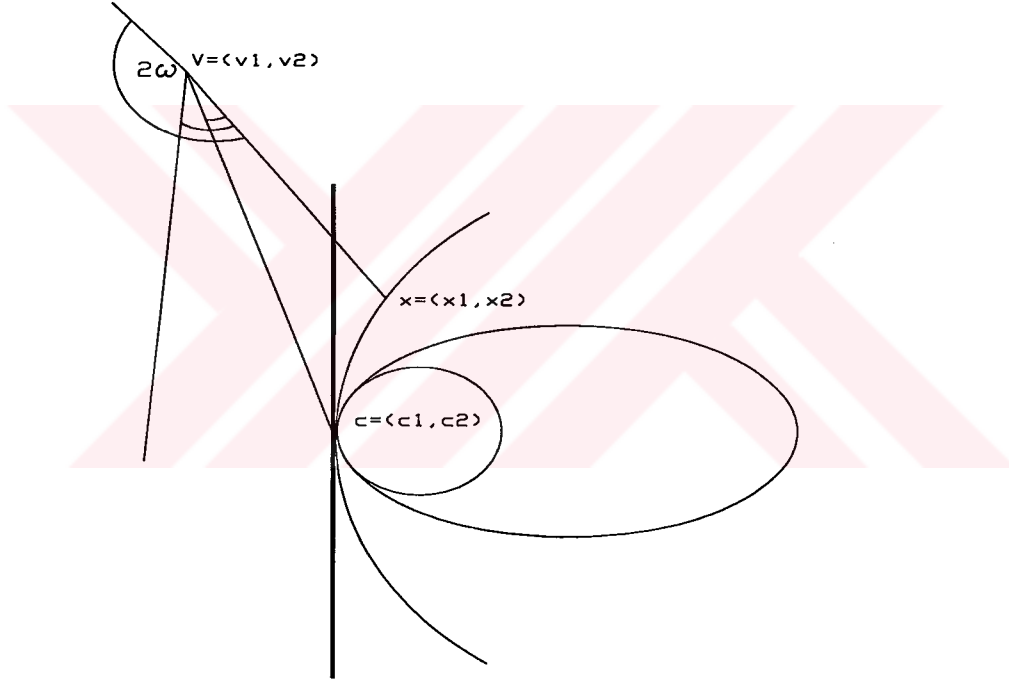


Şekil 3.2 Konik kesit örnekleri

Bir koni üzerinde herhangi bir nokta alınsın ve x noktası olarak isimlendirilsin. Koninin tepe açısı 2ω olsun. Tepe açısı, $[-\pi/2, \pi/2]$ aralığında bir değer olabilir. v koninin tepe noktası ve a da koninin eksenini tanımlayan birim vektör olsun. Bu durumda analitik olarak koni şu şekilde tanımlanabilir.

$$(x - v)a = \cos \omega \|x - v\| \quad (3.1)$$

Şekil 3.3'de tepe noktası V ve tepe açısı ω olan üç-boyutlu bir koninin bir daire, bir parabol ve bir düzlem oluşturacak şekilde bir düzlemle yaptığı arakesitler iki-boyutlu uzayda görülmektedir. Bu şekil, doğrusal (hiperdüzlem) ve dairesel (hiperküre) olmak üzere sırasıyla MLP ve RBF için karar sınırlarını göstermektedir. Diğer karar sınırları elips ve paraboldür.



Şekil 3.3 Bir giriş düzlemi ile bir koninin farklı karar sınırları (bir daire, bir parabol ve bir düz çizgi) oluşturacak şekilde kesişimi

Eşitlik (3.1)'deki nokta ve vektörlerin koordinatları iki-boyutlu uzay için aşağıdaki gibi tanımlanır.

$$(x_1 - v_1)a_1 + (x_2 - v_2)a_2 = \cos \omega \sqrt{(x_1 - v_1)^2 + (x_2 - v_2)^2} \quad (3.2)$$

Eşitlik (3.2), n-boyutlu giriş uzayı için şu şekilde geliştirilebilir.

$$\sum_{i=1}^{n+1} (x_i - v_i) a_i = \cos \omega \sqrt{\sum_{i=1}^{n+1} (x_i - v_i)^2} \quad (3.3)$$

Eşitliğin bu şekli n-boyutlu giriş uzayı için hiperkoni ve hiperdüzlem arasındaki kesişimi ifade etmektedir. Koninin tepe noktası koordinatı olan v yerine daire merkezi c kullanılabilir. Çünkü x noktası ve v arasındaki mesafe, 2ω ile verilen koninin tepe açısı 90° olduğunda dairenin yarı çapına eşittir. (Dorffner, 1994)

3.2 Konik Kesit Fonksiyonlu Yapay Sinir Ağı

n-katmanlı ağ için bir Çok Katmanlı Algılayıcı'nın çıkış fonksiyonu aşağıdaki eşitlikle verilir.

$$y_j = f(\sum w_{ij} x_j - \theta_j) \quad (3.4)$$

Burada w_{ij} önceki katmana olan ağırlıklar, x_j önceki katmanın çıkışı, θ_j eşik ve f sigmoid fonksiyonu gibi bir eşik fonksiyonudur.

Radyal Temelli Fonksiyonlu (RBF) ağlar, sadece bir gizli katmana sahiptir. Gizli katmandaki temel fonksiyonlar alıcı alanlarla giriş vektörleri arasındaki Öklid mesafesini hesaplayarak giriş vektörünün lokalleşmiş bir cevabını üretirler. Bir Radyal Temelli Fonksiyonlu (RBF) ağın genel formu;

$$y_j = \sum w_{ij} \phi_j(\|x - c_j\|) \quad (3.5)$$

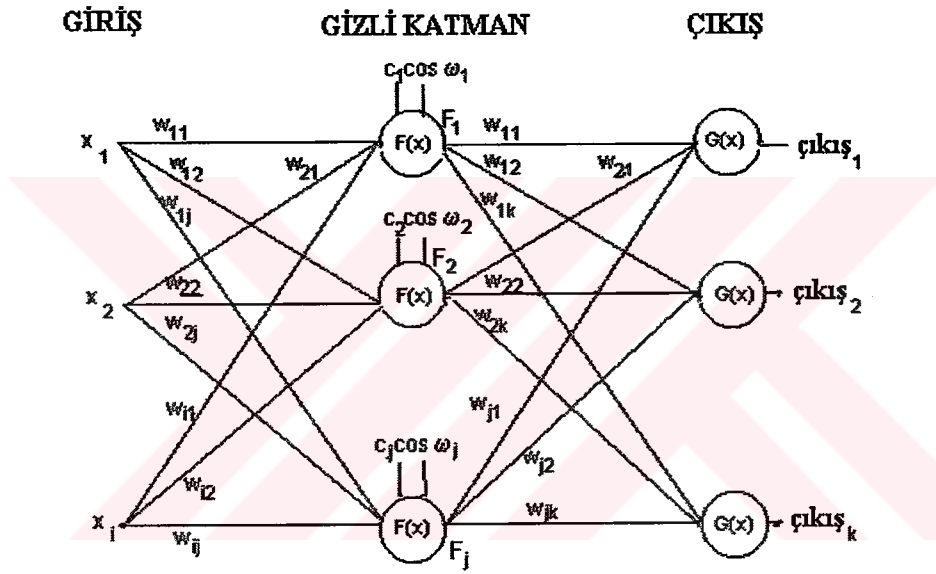
şeklinde verilir. Burada y_j ağın çıkışı, w_{ij} ikinci katmanın ağırlıkları, ϕ_j nonlineerlik, c_j merkezler ve x ağın girişidir.

Eşitlik (3.3)'ün sağ tarafı sol tarafından çıkarılarak

$$y_j = \sum_{i=1}^{n+1} (x_i - c_{ij}) w_{ij} - \cos \omega_j \sqrt{\sum_{i=1}^{n+1} (x_i - c_{ij})^2} \quad (3.6)$$

$$x_{n+1} \equiv 0$$

eşitliği elde edilir. Bu eşitlik CSFNN için genel aktivasyon ifadesini vermektedir. Burada, i ve j sırasıyla giriş katmanındaki ve gizli katmandaki birimlere karşı gelen indislerdir. Kolaylıkla fark edilebileceği gibi bu eşitlikte MLP ve RBF'tekine benzeyen temel iki kısım vardır. Burada, w_{ij} faktörleri Çok Katmanlı Algılayıcı'daki ağırlıkları ve c_j faktörleri de Radyal Temelli Fonksiyonlu ağlardaki merkez koordinatlarını temsil eder. ω_j Radyal Temelli Fonksiyonlu ağ yapısından Çok Katmanlı Algılayıcı ağ yapısına geçişi sağlayan koninin açılma açısıdır. (3.3) eşitliği dikkatle incelenirse şu sonuçlara varılabilir: ω açısı $\pi/2$ 'ye eşit olduğu zaman eşitliğin ikinci kısmı sıfır olur ve sadece birinci kısım etkindir. Yani, ağ bir MLP ağı gibi davranır. İkinci kısım ise bir RBF ağındaki merkezler ile girişler arasındaki Öklid mesafesini vermektedir. Şekil 3.4'de bir CSFNN'nin genel yapısı görülmektedir.



Şekil 3.4 Konik Kesit Fonksiyonlu Yapay Sinir Ağı

Konik Kesit Fonksiyonlu Yapay Sinir Ağı birçok avantaja sahiptir. Bu ağ yapısında öğrenme daha hızlıdır. Burada RBF'deki hız ve MLP'deki hata minimizasyonu birleştirilmiştir. Bu da karışık problemler ve standart RBF'le karşılaştırıldığında yüksek boyutlu girişler için etkili çözümler verir. RBF, MLP den daha hızlı yakınsamasına karşın, radyal temelli fonksiyonlar giriş uzayının boyutuyla expansiyel olarak arttığı için giriş verisinin boyutu yüksek olduğunda RBF ağları zaman tüketicidir. Bu nedenle yüksek boyutlu girişlere sahip problemlerde MLP tercih edilir. Dolayısıyla iki ağ yapısı arasında geçişi sağlayan CSFNN verilen uygulamaya göre optimum ağ çözümüne izin verir. (Dorffner, 1994; Yıldırım, 1997)

Konik Kesit Fonksiyonlu Yapay Sinir Ağlarının avantajları özetlenecek olursa:

- Kapalı karar sınırlarını açık sınırlara veya benzerlerine çevirme
- Giriş uzayını belirli bir yerle sınırlanmış bölgelere bölme
- Hızlı öğrenme
- Gizli katmanları daha etkili kullanma
- Hiperküresel ağların hızı ile geriye yayılma algoritmasının hata minimizasyonunu birleştirme
- Daha karmaşık problemler ve daha çok boyutlu girişler için daha iyi verimlilik
- Verilen uygulamaya göre en iyi sinir ağı çözümlerine izin verme

3.3 Konik Kesit Fonksiyonlu Yapay Sinir Ağının Eğitilmesi

Konik Kesit Fonksiyonlu Yapay Sinir Ağının eğitilmesi, birim merkezlerinin, bağlantı ağırlıklarının ve açılma açısının belirlenerek belli bir hata kriterine veya sınıflama oranına ulaşıncaya kadar bu parametrelerin güncelleştirilmesi ile yapılır. CSFNN'nin eğitime fazları ve öğrenme yöntemleri aşağıda özetlenmiştir.

Eğitime fazları:

- Başlangıç fazı
 - Merkezlerin belirlenmesi
 - Ağırlık ve açılma açısının başlatılması
- Öğrenme fazı

Öğrenme yöntemleri:

- Koni katlama yöntemi ile öğrenme: Bu yöntemde CSFNN, MLP, önceden eğitilmiş MLP veya RBF ağları gibi başlatılır. Bir gizli birimin eşik ve ağırlık değerleri rasgele veya bir kümeleme işlemi ile belirlenir. Gizli katmandan çıkışa olan ağırlıklar delta kuralı ile bulunabilir veya tüm ağ önceden belirlenmiş ağırlıklar kullanılarak standart geriye yayılma algoritması ile eğitilebilir. (Dorffner, 1994)
- Geriye yayılmalı öğrenme: Bu yöntemde ağ, RBF ağı gibi başlatılır. Ortogonal En Küçük Kareler yöntemi ile merkezler belirlendikten sonra ağ, standart geriye yayılma algoritması ile eğitilir. Burada sadece ağırlıklar değil aynı zamanda merkezler ve açılma açısı da güncelleştirilir (Yıldırım ve Marsland, 1997). Güncelleştirmede iki yol izlenir:
 - Aynı iterasyonda ω açılma açısı, merkezler ve ağırlıkların güncelleştirilmesi
 - Parametrelerin farklı iterasyonlarda güncelleştirilmesi

3.4 Adaptif Öğrenmeli Konik Kesit Fonksiyonlu Yapay Sinir Ağı

Adaptif öğrenmeli Konik Kesit Fonksiyonlu Yapay Sinir Ağı, başlangıç (parametrelerin başlatılması) ve öğrenme (parametrelerin güncelleştirilmesi) fazı olmak üzere iki fazda eğitilir:

Başlangıç fazında merkezlerin yerleri belirlenir; ayrıca ω ile verilen koninin açılma açısı ve katmanlar arası bağlantı ağırlıkları başlatılır. Bu fazda merkezler belirlenirken iki farklı yol izlenmiştir: 1) Önceden belirlenen bir merkez sayısına ulaşınca kadar Ortogonal En Küçük Kareler (Orthogonal Least Squares-OLS) yöntemi kullanılarak merkez değerleri belirlenir. 2) Merkez değerleri sıfıra atanır. Koninin açılma açısı, ağıın başlangıçta Radyal Temelli Fonksiyonlu ağ gibi davranmasını sağlayacak şekilde seçilir. Ağırlıklar belirlenirken, girişten gizli katmana olan ağırlıklar sıfıra atanır. Gizli katmandan çıkışa olan bağlantı ağırlıkları ise rasgele başlatılır.

Öğrenme fazı merkezlerin belirlenme yöntemine göre iki farklı şekilde yapılmaktadır. Eğer merkez değerleri sıfıra atanmış ise Çok Katmanlı Algılayıcılarda kullanılan hatanın geriye yayılımı yöntemi ile tüm parametreler (ağırlıklar, merkezler, açılma açısı) farklı iterasyonlarda sırayla güncelleştirilir. Merkez değerlerinin OLS ile belirlendiği durumda ise ağ, önceden belirlenen bir merkez sayısına ulaşınca kadar RBF ağırları gibi eğitilir. Daha sonra geriye yayılma yöntemi ile dokuz iterasyon boyunca ağırlıklar değiştirilir, sonraki bir iterasyonda da açılma açısı güncelleştirilir. Her iki durumda da güncelleştirme işlemi sırasında, geriye yayılma yönteminde kullanılan öğrenme oranı algoritmanın performansına göre değiştirilmektedir. Ayrıca algoritmaya momentum terimi eklenerek öğrenmenin iyileştirilmesine çalışılmıştır. Eğitim işlemi ya önceden belirlenen kabul edilebilir bir hata değerine ulaşıldığında ya da sınıflandırma işleminin tamamlanması halinde sonlandırılır.

Eğitim algoritması adım adım verilecek olursa;

Adım 1: Merkez sayısı ve toplam karesel hata veya sınıflanacak örnek sayısı belirlenir. Birinci katman ağırlıkları sıfıra atanır ve ω açılma açısı $\pi/4$ 'den başlar.

Adım 2: Merkezler sıfıra atanır veya OLS algoritması kullanılarak girişlerden üretilir. Gizli katman ve ikinci katman çıkışları hesaplanır.

Adım 3: Toplam karesel hata ve sınıflanan örnek sayısı hesaplanır.

Adım 4: Katman ağırlıklarını, duruma bağlı olarak merkezleri ve gizli katmandaki ω açılma açısını güncellemek için adaptif geriye yayılma algoritması çalıştırılır. Her bir katmandaki δ hata işaret vektörleri hesaplanır.

Adım 5: Çıkış katmanlarının ağırlıkları ayarlanır.

Adım 6: Gizli katman ağırlıkları, (merkezler) ve açılma açısı güncelleştirilir.

Adım 7: Katmanların çıkışları yeniden hesaplanır.

Adım 8: Yeni toplam karesel hata ve sınıflanan örnek sayısı hesaplanır. Eğer bu değer, hata hedefinden büyükse veya istenen sayıda örnek sınıflanamadıysa 4. Adıma gidilir. Aksi takdirde, eğitmeye son verilir.

3.4.1 Merkezlerin belirlenmesi için kullanılan Ortogonal En Az Kareler yöntemi

Ortogonal En Az Kareler (OLS) yöntemi minimum sayıda merkez seçimi için kullanılmaktadır (Chen vd., 1991). Merkezler, RBF ağlarının eğitiminde olduğu gibi veri tabanından seçilir.

$$d(n) = \sum_{i=1}^M p_i(n)\theta_i + e(n), \quad n = 1, 2, \dots, N \quad (3.7)$$

Burada $d(n)$, arzu edilen çıkıştır ve aynı zamanda bağımlı değişken olarak adlandırılır. θ_i , model parametreleri ve $p_i(n)$, regresörlerdir ve aynı zamanda $x(n)$ 'nin sabit fonksiyonlarıdır.

$$p_i(n) = p_i(x(n)) \quad (3.8)$$

Hata işareti $e(n)$, regresörlerle ilişkisiz varsayılır. (3.7) eşitliği matris formunda yazılırsa;

$$d = P\theta + E \quad (3.9)$$

burada

$$d = [d(1), d(2), \dots, d(N)]^T$$

$$P = [p_1, p_2, \dots, p_M]^T$$

$$\Theta = [\theta_1, \theta_2, \dots, \theta_M]^T$$

$$p_i = [p_i(1), p_i(2), \dots, p_i(N)]^T, \quad 1 \leq i \leq M$$

$$E = [e(1), e(2), \dots, e(N)]^T$$

Bir temel vektör kümesi içinden p_i regresör vektörleri alınır ve en küçük kareler çözümü θ , bu vektörler tarafından d 'nin ortogonal temelli vektörlerle taranan uzay üzerindeki izdüşümünün $P\theta$ olması koşulunu sağlar. Diğer bir deyişle, izdüşümün karesi, istenen çıkış enerjisinin bir parçasıdır. Farklı regresörler genel olarak ilişkilidir. Tek bir regresörün çıkış enerjisini nasıl etkileyeceği açık değildir. OLS metodu, p_i dizilerinin bunlara karşı gelen bir dizi ortogonal temel vektör kümesine dönüşümünü içerir ve böylelikle her bir temel vektörden istenen çıkış enerjisine bireysel olarak p_i 'nin katkısının hesaplanabilmesi mümkün olur. P regresör matrisi,

$$P = WA \quad (3.10)$$

şeklinde ayrıştırılır. Burada A , $M \times M$ boyutunda, köşegenlerinde 1 olan ve 1'lerin altında da sıfır bulunan bir kare matristir. W , $N \times M$ boyutunda, köşegen sütunları w_i olan bir matristir.

$$W^T W = H \quad (3.11)$$

Burada H , h_i elemanlarından oluşur.

$$h_i = w_i^T w_i = \sum_{n=1}^N w_i(n)w_i(t), \quad 1 \leq i \leq M \quad (3.12)$$

Ortogonal temel w_i vektörlerinin bir kümesi ile taranan uzay, p_i dizisinde taranan uzayla aynıdır ve (3.9) eşitliği şu şekilde tekrar yazılabilir:

$$d = Wg + E \quad (3.13)$$

Ortogonal en az kareler çözümü aşağıdaki gibi verilebilir.

$$g = H^{-1} W^T d \quad (3.14)$$

$$g_i = w_i^T d / (w_i^T w_i), \quad 1 \leq i \leq M \quad (3.15)$$

Klasik Gram-Schmidt yöntemi bir anda A'nın bir sütununu hesaplar ve P'yi aşağıdaki gibi ortogonalleştirir:

$$\begin{aligned} w_1 &= p_1 \\ \alpha_{ik} &= w_i^T p_k / (w_i^T w_i), \quad 1 \leq i < k \\ w_k &= p_k - \sum_{i=1}^{k-1} \alpha_{ik} w_i \end{aligned} \quad (3.16)$$

burada $k=2, \dots, M$ dir. Bu yöntemle k. sütun, k. adımda önceden ortogonalleştirilmiş $k-1$ tane sütunun her birine ortogonal yapılır ve işlem $k=2, \dots, M$ için tekrarlanır. OLS metodu normal LS metoduna göre daha üstün sayısal özelliklere sahiptir. RBF ağlarında $x(n)$ veri noktalarının sayısı çoğunlukla çok büyüktür ve veri kümesinde merkezler alt küme olarak seçilme durumundadır. Genel olarak M ile gösterilen bütün aday regresörlerin sayısı çok büyük olabilir ve uygun modelleme için sadece önemli regresörlere ihtiyaç duyulabilir. Bu önemli regresörler, OLS algoritması kullanılarak seçilebilir. $d(n)$ 'nin enerjisi veya karelerinin toplamı

$$d^T d = \sum_{i=1}^M g_i^2 w_i^T w_i + E^T E \quad (3.17)$$

ile ifade edilir. Eğer d istenen çıkış vektörü ise $d(n)$ 'nin değişimi şöyle verilebilir.

$$N^{-1} d^T d = N^{-1} \sum_{i=1}^M g_i^2 w_i^T w_i + N^{-1} E^T E \quad (3.18)$$

Hatadaki azalma aşağıdaki gibi tanımlanabilir:

$$e_i = g_i^2 w_i^T w_i / (d^T d), \quad 1 \leq i \leq M \quad (3.19)$$

Regresör seçimini özetlersek:

- Birinci adımda $1 < i < M$ için

$$\begin{aligned} w_1^{(i)} &= p_i \\ g_1^{(i)} &= (w_1^{(i)})^T d / ((w_1^{(i)})^T w_1^{(i)}), \quad 1 \leq i < k \\ [err] &= (g_1^{(i)})^2 (w_1^{(i)})^T w_1^{(i)} / (d^T d) \end{aligned} \quad (3.20)$$

hesaplanır. Hesaplanan hata değerleri içindeki en büyük hatanın indisi tespit edilir ve buna göre

$$w_1 = w_1^{(i)} = p_i \quad (3.21)$$

$$[err]_1^{(i)} = \max\{[err]_i^{(i)}, \quad 1 \leq i \leq M\} \quad (3.22)$$

seçilir.

- k. adımda $k \geq 2$, ve $1 \leq i \leq M$, $i \neq i_1, \dots, i_{k-1}$ için

$$\begin{aligned} \alpha_{jk}^{(i)} &= w_j^T p_i / (w_j^T w_j), \quad 1 \leq j < k \\ w_k^{(i)} &= p_i - \sum_{j=1}^{k-1} \alpha_{jk}^{(i)} w_j \\ g_k^{(i)} &= (w_k^{(i)})^T d / ((w_k^{(i)})^T w_k^{(i)}) \\ [err]_k^{(i)} &= (g_k^{(i)})^2 (w_k^{(i)})^T w_k^{(i)} / (d^T d) \end{aligned} \quad (3.23)$$

hesaplanır ve aynı şekilde her i. adımda hesaplanan hataların içinden en büyük olanının indisi tespit edilip w' 'ya atamalar yapılır.

$$[err]_k^{(i_k)} = \max\{[err]_k^{(i)}\}, \quad 1 \leq i \leq M, \quad i \neq i_1, \dots, i_{k-1} \quad (3.24)$$

$$w_k = w_k^{(i_k)} = p_{i_k} - \sum_{j=1}^{k-1} \alpha_{jk} w_j \quad (3.25)$$

$$\alpha_{jk} = \alpha_{jk}^{(i_k)}, \quad 1 \leq j \leq k \quad (3.26)$$

Bu işlem M. adımda

$$1 - \sum_{j=1}^m [err]_j < \rho \quad (3.27)$$

tamamlanır. Tolerans, $0 < \rho < 1$ arasında tanımlanmıştır. Birinci adımda ilk merkez yerleştirilir ve hata terimi hesaplanır. k. adımda, yeterli bir ağ yapısı elde edinceye kadar merkezler birer birer eklenir. Bu çalışmada $p_i(n)$ ile verilen regresörler olarak, lojistik fonksiyon kullanılmıştır.

3.4.2 Ağırlıkların güncelleştirilmesi

İleri beslemeli ağlarda aktivasyon fonksiyonu Eşitlik (3.28)'deki gibi tanımlanabilir:

$$out_{pj} = f_j(y_{pj}) \quad (3.28)$$

d istenen çıkış ve out ağın yanıtı olmak üzere; p . patern için toplam karesel hata

$$E_p = \frac{1}{2} \sum_j (d_j - out_j)^2 \quad (3.29)$$

şeklinde tanımlanır. Karesel hatayı minimum yapmak için gradyen azalma algoritması kullanılır. Karesel hata fonksiyonunun w 'ya göre gradyeninin negatif doğrultusunda hareket edilerek hata azaltılır. Buna göre herhangi bir ağırlık için değişim şu şekilde tanımlanabilir.

$$\Delta_p w_{ij} = -\alpha \frac{\partial E_p}{\partial w_{ij}} \quad (3.30)$$

Zincir kuralı ile (3.30) eşitliğinin ikinci kısmı aşağıdaki formda yazılabilir:

$$\frac{\partial E_p}{\partial w_{ij}} = \frac{\partial E_p}{\partial y_{pj}} \cdot \frac{\partial y_{pj}}{\partial w_{ij}} \quad (3.31)$$

Eşitliğin sağ tarafında çarpım halinde olan iki terim ayrı ayrı hesaplanırsa;

$$\frac{\partial y_{pj}}{\partial w_{ij}} = \frac{\partial}{\partial w_{ij}} \left[\sum_{k=1} (out_{pk} - c_{kj}) w_{kj} - \cos \omega_j \sqrt{\sum_{k=1} (out_{pk} - c_{kj})^2} \right] \quad (3.32)$$

buradan,

$$\frac{\partial y_{pj}}{\partial w_{ij}} = (out_{pi} - c_{ij}) \quad (3.33)$$

elde edilir.

$$\frac{\partial E_p}{\partial y_{pj}} = -\delta_{pj} \quad (3.34)$$

olarak tanımlanırsa, p . patern için toplam karesel hatanın w_{ij} ağırlığıyla değişimi

$$-\frac{\partial E_p}{\partial w_{ij}} = \delta_{pj} (out_{pi} - c_{ij}) \quad (3.35)$$

şeklinde yazılabilir. Dolayısıyla p . patern için w_{ij} 'deki değişim

$$\Delta_p w_{ij} = \eta \delta_{pj} (out_{pi} - c_{ij}) \quad (3.36)$$

olarak bulunur. Hata işaret terimi;

$$\delta_{pj} = -\frac{\partial E_p}{\partial y_{pj}} = -\frac{\partial E_p}{\partial out_{pj}} \cdot \frac{\partial out_{pj}}{\partial y_{pj}} \quad (3.37)$$

zincir kuralı ile (3.31) eşitliğindeki gibi tanımlanırsa, çarpımın ikinci terimi;

$$\frac{\partial out_{pj}}{\partial y_{pj}} = f'_j(y_{pj}) \quad (3.38)$$

dir. Çarpımın birinci terimi de

$$\frac{\partial E_p}{\partial out_{pj}} = -(d_{pj} - out_{pj}) \quad (3.39)$$

şeklinde elde edilir. Sonuç olarak hata işaret terimi,

$$\delta_{pj} = (d_{pj} - out_{pj}) f'_j(y_{pj}) \quad (3.40)$$

olarak bulunur. Gizli katman ağırlıklarının ayarlanmasında bütün katman çıkışlarının hata terimleri etkilidir. Buna göre;

$$\sum_k \frac{\partial E_p}{\partial y_{pk}} \cdot \frac{\partial y_{pk}}{\partial out_{pj}} = \sum_k \frac{\partial E_p}{\partial y_{pk}} \cdot \frac{\partial}{\partial out_{pj}} \left[\sum_i (out_{pi} - c_{ik}) w_{ik} - \cos \omega_k \|out - c_j\| \right] \quad (3.41)$$

$$\frac{\partial}{\partial out_{pj}} \left[\sum_i (out_{pi} - c_{ik}) w_{ik} - \cos \omega_k \|out - c_j\| \right] = w_{jk} - \cos_k \frac{out_j - c_j}{\|out_p - c_j\|} \quad (3.42)$$

$$\sum_k \frac{\partial E_p}{\partial y_{pk}} \cdot \frac{\partial y_{pk}}{\partial out_{pj}} = - \sum_k \delta_{pk} \left[w_{jk} - \cos_k \cdot \frac{out_j - c_j}{\|out_p - c_j\|} \right] = A \quad (3.43)$$

Gizli katman birimleri için hata işaret terimi terimi tekrar yazılacak olursa

$$\delta_{pj} = f'_j(y_{pj}) A \quad (3.44)$$

Sonuç olarak ağırlıkların güncelleştirilmesi çıkış katmanı için:

$$\Delta_p w_{ji} = \eta \delta_{pj} (out_{pi} - c_{ij}) = \eta out_{pj} (1 - out_{pj}) (d_{pj} - out_{pj}) (out_{pi} - c_{ij}) \quad (3.45)$$

Gizli katman için

$$\Delta_p w_{ji} = \eta out_{pj} (1 - out_{pj}) A (out_{pi} - c_{ij}) \quad (3.46)$$

olarak tanımlanabilir. Burada aktivasyon fonksiyonu olarak logaritmik sigmoid fonksiyonu kullanılmıştır.

$$out_{pj} = \frac{1}{1 + e^{-y_{pj}}} \quad (3.47)$$

Sigmoid fonksiyonunun türevi ise (3.48)'de verilmiştir.

$$f'_j(y_{pj}) = \frac{\partial out_{pj}}{\partial y_{pj}} = out_{pj} (1 - out_{pj}) \quad (3.48)$$

3.4.3 Merkezlerin güncelleştirilmesi

Merkezlerin güncelleştirilme işlemi de, ağırlıkların güncelleştirilmesine benzer şekilde genelleştirilmiş delta kuralından yararlanılarak yapılır.

$$\Delta_p c_{ij} \propto -\frac{\partial E_p}{\partial c_{ij}} \quad (3.49)$$

Hatanın merkezlere göre değişimini bulmak için zincir kuralı uygulanarak; (3.49) eşitliği iki parça halinde yazılabilir: Toplam karesel hatanın aktivasyona göre türevi ve aktivasyonun merkezlere göre türevi.

$$\frac{\partial E_p}{\partial c_{ij}} = \frac{\partial E_p}{\partial y_{pj}} \cdot \frac{\partial y_{pj}}{\partial c_{ij}} = -\delta_{pj} \cdot \frac{\partial y_{pj}}{\partial c_{ij}} \quad (3.50)$$

eşitliğindeki ikinci terim

$$\frac{\partial y_{pj}}{\partial c_{ij}} = -w_{ij} + \cos \omega_j \frac{a_i - c_i}{\|a - c_j\|} = B \quad (3.51)$$

olarak bulunur. Gizli katmandaki merkezlerin ayarlanmasında yine çıkış katmanındaki hata terimlerinin etkisi vardır. Buna göre, gizli katman birimleri için merkezlerin güncelleştirilmesi aşağıdaki gibi tanımlanabilir:

$$\Delta_p c_{ij} \propto out_{pj} (1 - out_{pj}) A.B \quad (3.52)$$

Çıkış katmanındaki bütün düğümler için, $\cos \omega = 0$ yani, CSFNN ifadesindeki (3.6) RBF kısmının etkisi olmadığı için merkezler güncelleştirilmez.

3.4.4 Açılma açısının güncelleştirilmesi

Açılma açısının güncelleştirilmesi, yine genelleştirilmiş delta kuralından yararlanılarak yapılır. Bunun için kural

$$\Delta_p \omega_j \propto -\frac{\partial E_p}{\partial \omega_j} \quad (3.53)$$

ile tanımlanır. Hatanın açılma açısına göre türevi,

$$\frac{\partial E_p}{\partial \omega_j} = \frac{\partial E_p}{\partial y_{pj}} \cdot \frac{\partial y_{pj}}{\partial \omega_j} = -\delta_{pj} \cdot \frac{\partial y_{pj}}{\partial \omega_j} \quad (3.54)$$

ile verilir. Burada çarpımın ikinci türevi,

$$\frac{\partial y_{pj}}{\partial \omega_j} = \sin \omega_j \cdot \|out - c_j\| = C \quad (3.55)$$

olarak bulunur. Gizli katman birimleri için açı güncelleştirilmesi, çıkış katmanından gelen hata terimlerinin etkisi de hesaba katılarak;

$$\Delta_p \omega_j \propto out_{pj} (1 - out_{pj}) A.C \quad (3.56)$$

şeklinde elde edilir. Burada da çıkış katmanında ağırlık MLP özelliği gösterdiği ve CSFNN (3.6) ifadesindeki RBF kısmının çıkış katmanında etkili olmadığı düşünülmüştür. Bu nedenle, açılma açısının güncelleştirilmesi sadece gizli katman için yapılmaktadır.

3.4.5 Adaptif öğrenme oranı ve momentum

Çok katmanlı ağlar için hata yüzeyinin şekli parametre uzayının farklı bölgelerinde çok farklı olabilir. Eğitim işlemi süresince öğrenme oranını ayarlayarak yakınsama hızlandırılabilir. Burada önemli olan öğrenme oranının ne zaman ve ne kadar değiştirileceğini belirlemektir. Adaptif öğrenme oranı ile sürekli olarak bir çevrimdeki hata $E(k)$, bir önceki çevrimdeki hata $E(k-1)$ ile karşılaştırılmaktadır. Böylece eğimin arttığı yerlerde öğrenme oranı azaltılarak veya düz yüzeylerde öğrenme oranı artırılarak yakınsamanın hızlanması sağlanabilir.

Yakınsamayı hızlandırmada kullanılan ikinci bir yöntem de momentum kullanarak ağırlık bir minimum hata civarında daha ileri bir öğrenme olmaksızın osilasyon yapmasını engellemek yani ağırlık daha kararlı hale getirmektir. Bölüm 2.3.5.1'de açıklandığı gibi momentum terimi, önceki ağırlık değişiminin etkisini belirleyen bir sabittir. Önceki ağırlık değişimi $\Delta w(k-1)$, büyükse, yeni değişim $\Delta w(k)$ de büyük olacaktır. Bu da, hata-ağırlık uzayındaki küçük dalgalanmaları düzleştirmeyi sağlar. Değişimin yönü doğruysa ve öğrenme minimuma doğru gidiyorsa; momentum, bu kararlılık etkisinden dolayı daha büyük öğrenme oranı değerlerine

izin verir. Momentum terimi kullanılmadığı zaman, düşük öğrenme oranındaki minimuma ulaşma çok zaman alır, yüksek öğrenme oranı için ise osilasyonlardan dolayı minimuma asla ulaşamaz. Momentum eklendiği zaman minimuma daha hızlı ulaşılabilir. (Zurada, 1995; Hagan vd., 1996)

Konik Kesit Fonksiyonlu Yapay Sinir Ağı'nın eğitilmesinde yakınsamayı hızlandırmak için standart geriye yayılma algoritmasındaki öğrenme oranı adaptif yapılmış ve momentum terimi eklenmiştir. Öğrenme oranını ayarlamak için değişik yaklaşımlar vardır. Burada algoritmanın performansına göre öğrenme oranının değiştirildiği Değişken Öğrenme Oranlı geriye yayılma algoritmasının (Variable Learning Rate Backpropagation-VLBP) kuralları kullanılmıştır (Hagan vd., 1996). Buna göre:

- 1- Eğer ağırlık güncelleştirilmesinden sonra karesel hata (tüm eğitim seti üzerinden) ξ oranında (genellikle %1.05'tir) artıyorsa, bu güncelleştirme işlemi iptal edilir. Öğrenme oranı, 0 ile 1 arasında olan bir faktörle (ρ) çarpılır ve momentum katsayısı sıfıra atanır.
- 2- Eğer ağırlık güncelleştirilmesinden sonra karesel hata azalıyorsa, güncelleştirme işlemi geçerli sayılır ve öğrenme oranı 1'den büyük bir faktör (η) ile çarpılır. Eğer momentum katsayısı sıfıra atanmışsa, bu iptal edilerek orijinal değerine set edilir.
- 3- Eğer karesel hata ξ oranından daha az artıyorsa, ağırlık güncelleştirilmesi geçerlidir. Fakat öğrenme oranı ve momentum katsayısı değiştirilmez.

4.KONİK KESİT FONKSİYONLU YAPAY SİNİR AĞININ ÇEŞİTLİ PROBLEMLER ÜZERİNDE İNCELENMESİ

Bu bölümde, adaptif öğrenmeli Konik Kesit Fonksiyonlu Yapay Sinir Ağı'nın (CSFNN) performansı çeşitli problemler üzerinde incelendi:

- Elektromiyografik işaretlerin sınıflandırılması problemi
- Kontakt lens uydurma problemi
- İris bitkisi sınıflandırma problemi
- Fetus gelişimi problemi
- Fonksiyon yaklaşımı problemi
- Ne21900 mikrodalga transistörün modellenmesi problemi
- Karakter tanıma problemi

CSFNN için geliştirilen eğitime ve test algoritması MATLAB programında yazılmıştır (Ek 6). Elde edilen sonuçlar, farklı ağ yapıları kullanılarak daha önce yapılan çalışmalarla ve adaptif öğrenme ve momentum terimi kullanılmayan standart CSFNN ile elde edilmiş sonuçlarla karşılaştırıldı.

4.1 Elektromiyografik İşaretlerin Sınıflandırılması Problemi

4.1.1 Elektromiyografik işaret nedir?

Elektromiyogram (EMG), kasın kasılması sonucu ortaya çıkan ve kaynağı elektro kimyasal olaylar olan biyopotansiyel işaretlerdir. EMG işaretleri, vücuttan özel elektrodlar yardımıyla alınırlar ve elektrod çeşidine göre farklı adlarla anılırlar. İşaretlerin değerlendirilmesi zaman domeninde ve/veya frekans domeninde yapılabilir. EMG, klinik uygulamalarda hastalık teşhislerinde ve kol kesilmesi gibi durumlarda kesik yere takılan protezin hareket ettirilmesini sağlayacak kaynak işaret olarak kullanılır. Protez kontrolü, bir çift yüzey elektrotundan sağlanan miyoelektrik işaretler kullanılarak yapılabilir. Bu durum özellikle dirsek üstünden kolunu kaybetmiş vakalarda uygulanabilir. Kontrol işlemi, her bir hareket için aynı olan ve tekrarlanabilen kas kasılma karakterleri kümesinin oluşturduğu miyoelektrik işaretlerden kolaylıkla çıkarılan karakteristik parametrelerle yapılır. Bu parametreler kullanılarak farklı kas kasılmalarına ait karakterleri sınıflara ayırmak mümkündür. Kas kasılması sınıflarının her biri, protez cihazının özel bir fonksiyonunu tetiklemek için kullanılır. Sınıflar arasında iyi bir

ayrım yapabilmek için, protez alet yerine takıldığındaki normal kol fonksiyonu ile hiçbir benzerlik göstermeyebilen olağan dışı kas kasılmalarının bir setinin de üretilmesi gerekebilir. Farklı kas kasılma karakterlerini ayırma kabiliyeti, kaslardan gelen miyoelektrik işaret bilgisine bağlıdır. Örnek olarak; üç kasın yakınına yerleştirilmiş bir çift deri yüzey elektrodundan elde edilen işaret, üç kastan her birinin aktivitesine ait bilgiler içerecektir. Yüzey elektrotunun yakınlarında birkaç kas varsa, sonuçta oluşan miyoelektrik işaret her kastaki aktiviteye ait bilgiler içerir. Kas setinin içindeki kasılmalarından oluşan karakter, alınan işaretten çıkarılan özelliklere dayalı olarak tanımlanabilir. (Graupe, 1975) tarafından gerçekleştirilen deneylere dayalı olarak, bu tip kas hareketlerinin, miyoelektrik işaret özelliklerinin oluştuğu kümelerden tanımlanabileceği söylenebilir. Normal kol fonksiyonlarının kümesi ile bileşik kas kasılma karakterleri arasında önemli bir fark varsa, normal kol fonksiyonlarının direkt tanımlanmasına bağlantılı olan miyoelektrik işaretin sınıflandırılması yapılabilir. (Kelly vd, 1990; Yazgan, 1991)

4.1.2 EMG işaretlerin modellenmesi

EMG işaretlerinin genlik değerleri rasgele dağılım gösterdiklerinden Gauss dağılım fonksiyonu ile temsil edilebilirler. Genellikle tepeden tepeye 0-6 mV veya 0-1.5 mV (rms) değerler arasında değişirler. İşaretin kullanışlı enerjisi baskın bölge 50-150 Hz olmak üzere 0-500 Hz frekansları ile sınırlıdır. Kullanışlı işaretlerden anlaşılan, elektriksel gürültü seviyesinin üstünde enerjiye sahip işaretler olmalıdır. Ölçülen miyoelektrik işaretlerin bir yapay sinir ağına uygulanabilmesi için, hücrelerin polarizasyon ve depolarizasyonu sonucu ortaya çıkan; yani, fiziksel bir anlamı olan kavramların matematiksel ifadelere dönüştürülmesi gerekir. Bu işleme modelleme denir. Sistem modellemesi ve işaret modellemesi olarak ikiye ayrılır. Bu uygulamada işaretlerin modellenmesi için AR (Auto-Regressive) katsayıları kullanılmıştır. Elektromiyografi işaretlerinin yeterince küçük zaman aralıklarında örneklenmesi halinde durağan sayılabilmesi, AR modelinin kullanılmasına olanak sağlamaktadır.

4.1.3 AR katsayıları

Sistemlerin çoğu, geçmişteki çıkışlarının, o andaki ve geçmişteki girişlerinin lineer kombinasyonları ile modellenebilir. Matematiksel olarak bu, aşağıdaki gibi gösterilebilir.

$$y_k = \sum_{i=1}^M a_i y_{k-i} + \sum_{j=0}^N b_j x_{k-j} \quad (4.1)$$

Burada x_k ve y_k sistemin ayrık zamandaki giriş ve çıkış serileridir. Girişe ait herhangi bir bilgiye erişilemiyorsa (EMG işaretlerinde olduğu gibi), giriş beyaz gürültü olarak alınabilmektedir. Yukarıdaki eşitlikten görüldüğü gibi model parametreleri olan $a_i, i=1,2,\dots,M$ ve $b_j, j=1,2,\dots,N$ bilindiği takdirde, geçmişteki giriş ve çıkış değerlerinden k anındaki y_k değeri hesaplanabilir. Bu nedenle, bu modelleme doğrusal kestirim (linear prediction) olarak da adlandırılır. (4.1) eşitliğinin Z dönüşümü alınarak sistemin transfer fonksiyonu

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{j=0}^N b_j Z^j}{1 + \sum_{i=0}^M a_i Z^i} \quad (4.2)$$

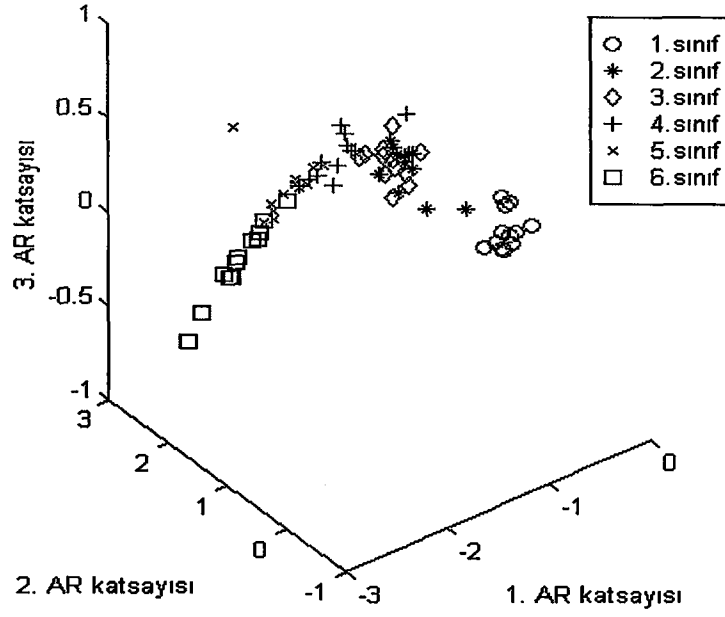
olarak bulunur. Sıfırları ve kutupları içeren bu genel model ARMA (Auto-Regressive Moving Average) olarak adlandırılır. b_j katsayılarının sıfır olduğu durumda sistem, yalnızca çıkışlar kullanılarak modellenir. Tüm kutup veya AR model olarak anılan bu tanımın fark denklemleriyle gösterimi

$$y_k = -\sum_{i=1}^M a_i y_{k-i} + b_0 x_k \quad (4.3)$$

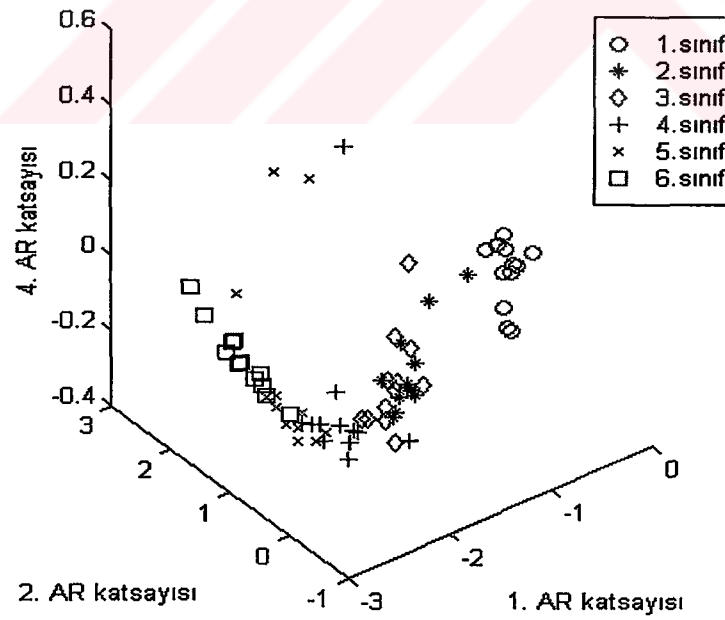
şeklinde dir. EMG işaretleri, yeterince küçük zaman aralıklarında örneklenmesi halinde durağan sayılabildiği için AR modeli kullanılabilmektedir. Bu noktadan sonra yapılacak olan (4.3) denklemindeki a_i ve b_0 'ı yani AR katsayılarını bulmaktır.

4.1.4 Miyoelektrik işaretlerin yapay sinir ağlarına uygulanması

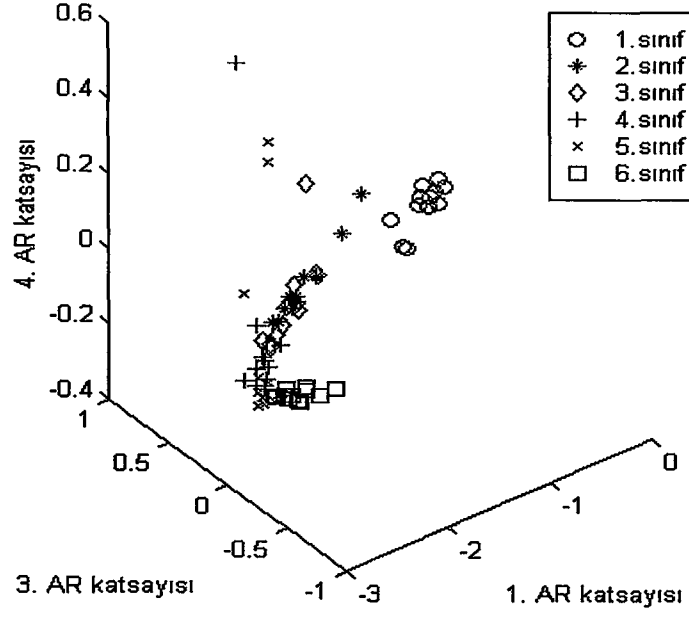
Miyoelektrik işaretler sinir ağına giriş bilgisi olarak uygulanmaktadır. Ölçülen miyoelektrik işaretlerin bir yapay sinir ağına uygulanabilmesi için bu kavramların bir önceki bölümde açıklandığı şekilde matematiksel ifadelere dönüştürülmesi gerekir. Kaslardan alınan EMG işaretlerinin AR katsayıları modellemesi ile oluşan 72 satır ve 4 sütunlu matris, yapay sinir ağına giriş uzayı olarak uygulanabilecek duruma getirilmiştir (Graupe, 1975; Karlık, 1994). Burada, satırlar kas hareketlerine ilişkin altı farklı hareketi belirler. Bunlar; dirsek kapama, dirsek açma, bilek döndürme, kavrama, bilek bükme ve dinlenme hareketleridir. Bu pozisyonların her biri bir sınıf olarak kabul edilmiş ve dört AR katsayısı ile temsil edilmiştir. Her bir sınıfa ait 12 örnek ele alınmıştır. Veri kümesine ait detaylar Ek1'de verilmiştir. Giriş verilerinin üç boyutlu dağılımları Şekil 4.1, 4.2, 4.3 ve 4.4'de gösterilmiştir.



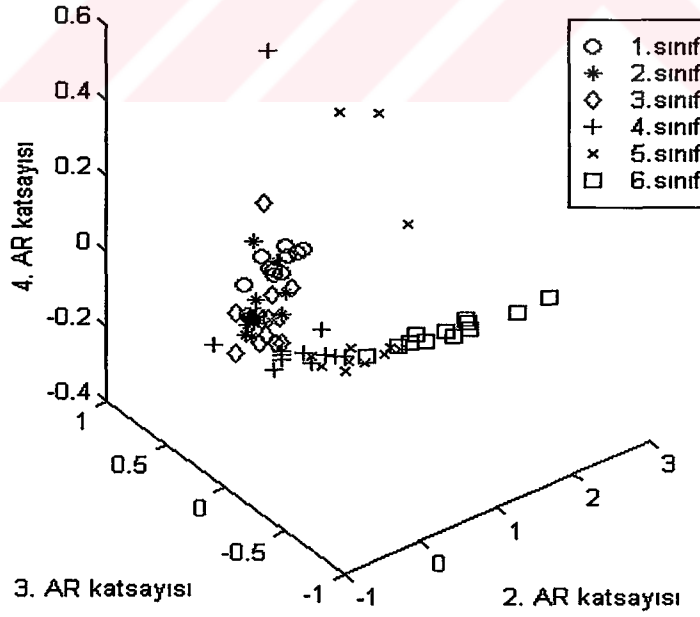
Şekil 4.1 EMG işaretlerinin 1., 2. ve 3. AR katsayılarına göre dağılımı



Şekil 4.2 EMG işaretlerinin 1., 2. ve 4. AR katsayılarına göre dağılımı



Şekil 4.3 EMG işaretlerinin 1., 3. ve 4. AR katsayılarına göre dağılımı



Şekil 4.4 EMG işaretlerinin 2., 3. ve 4. AR katsayılarına göre dağılımı

4.1.5 Miyoelektrik işaretleri sınıflamada kullanılan yöntem

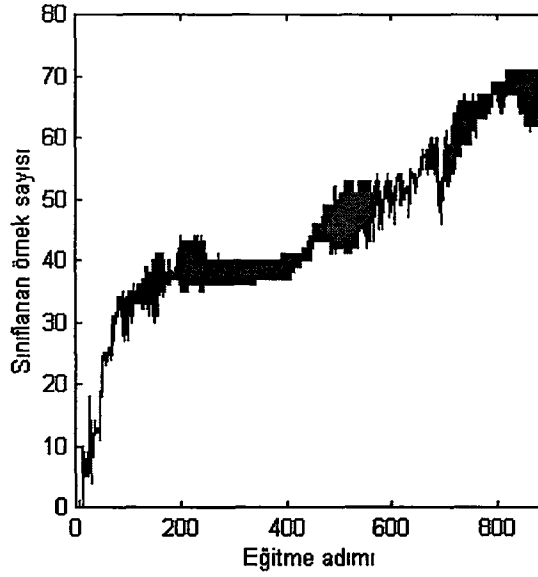
Konik Kesit Fonksiyonlu Yapay Sinir Ağı kullanılarak, AR katsayıları ile temsil edilen EMG işaretlerine ait farklı kas hareketleri sınıflandırılmıştır. Uygulamada üç farklı eğitim ve test kümesi kullanılmıştır. İlk olarak tüm veri bilgisi ağa girilerek eğitilmiş ve daha sonra ağ aynı verilerle test edilmiştir. İkinci olarak ‘birini dışarda bırak’ (Leave-one-out) yöntemi kullanılmış; yani, eğitim setinden bir örnek ayrılarak diğer örnekler eğitilmiştir. Bu yöntem, k-katlamalı çapraz-geçerlilik (k-fold cross-validation) olarak adlandırılan bir genelleştirme yönteminin özel bir halidir. Burada k, örnek sayısı kadar seçilerek 72 örnek ayrı ayrı denenmiştir. Ayrıca örnekleri temsil eden AR katsayıları da iki farklı şekilde incelenmiştir: 1) Tüm AR katsayıları kullanılarak 4 girişli 6 çıkışlı bir ağ yapısı kullanılmıştır. 2) İki AR katsayısı kullanılarak 2 girişli 6 çıkışlı bir yapı seçilmiştir. Son olarak da her sınıfa ait örneklerin %75'i eğitim için, geriye kalan ve eğitimde kullanılmayan %25'i de test için kullanılmıştır. Ayrıca elde edilen sonuçlar, MLP (Kelly vd, 1990; Karlık, 1994) ve Bulanık K-En Yakın Komşuluk Sınıflandırıcıları (Şeker, 1995) ile bulunan sonuçlarla karşılaştırılmıştır.

4.1.6 Eğitim ve test sonuçları

Daha önceki çalışmalarda (Kelly vd, 1990; Karlık, 1994) geriye yayılma algoritması (MLP ağı) kullanılarak, EMG işaretlerinin sınıflanması probleminde, tüm veri bilgisinin ağa girilmesi durumunda 3000 iterasyondan sonra %96 doğruluk oranı ile yakınsama sağlandığı ve iterasyon sayısı artırıldığında bu oranın iyileşmediği görülmüştür. Bu çalışmalarda yakınsama kriteri olarak da ± 0.25 kullanılmıştır. Konik Kesit Fonksiyonlu Yapay Sinir Ağı ile yapılan eğitim sonucunda ise, aynı yakınsama kriterleri ile farklı momentum katsayılarına ve kullanılan merkez sayılarına bağlı olarak 890 ila 1500 iterasyon sonucunda %100 doğruluk oranına ulaşılmıştır. Çizelge 4.1'de giriş verisinin tamamının eğitimde kullanılması durumunda elde edilen sonuçlar farklı algoritmalar için karşılaştırılmış ve test sonuçları yüzdesel olarak verilmiştir. Ayrıca en iyi sonuçların alındığı momentumun 0.8 ve merkez sayısının 5 olarak seçildiği durum için sınıflanan örnek sayısına ait grafik Şekil 4.5'de gösterilmektedir.

Çizelge 4.1 Tüm test verisinin farklı algoritmalar için karşılaştırılması

	MLP	CSFNN
Sınıflama Oranı	%96	%100
İterasyon Sayısı	3000	890-1500



Şekil 4.5 Momentum = 0.8, merkez sayısı = 5 için tüm veri kümesine ait test sonucu

Birini dışarda bırak (Leave-one-out) yöntemi kullanılarak elde edilen sonuçlar, aynı yöntemi kullanan Bulanık K-En Yakın Komşuluk Sınıflandırıcılarının sonuçları ile karşılaştırılmıştır (Çizelge 4.2).

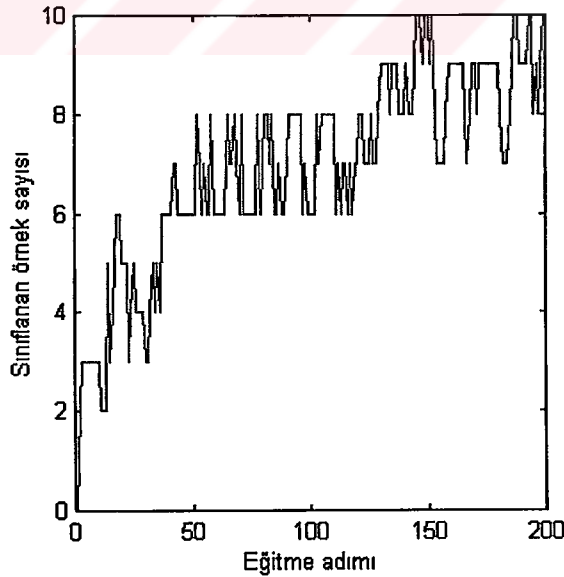
Çizelge 4.2 Leave-one-out yöntemi kullanılarak farklı algoritmaların karşılaştırılması

	Bulanık Sınıflayıcı	CSFNN
Sınıflama Oranı (4 AR Katsayısı için)	%64-%68	%100
İterasyon Sayısı (4 AR Katsayısı için)	5000	7-110
Sınıflama Oranı (2 AR Katsayısı için)	%53-%71	%98.6
İterasyon Sayısı (2 AR Katsayısı için)	5000	7-190

Değişik merkez sayıları ve momentum katsayısı ile test edilen verinin Konik Kesit Fonksiyonlu Yapay Sinir Ağı ile %98.6 - %100 doğrulukla sınıflandığı görülmüştür. Eğitim sırasında iki farklı giriş kümesi kullanılmıştır. Bunlardan birincisinde her bir örneğe ait özniteliklerin tamamı yani dört AR katsayısı eğitimde kullanılmıştır. İkinci kümede ise girişi belirleyen özniteliklerden sadece ikisi yani iki AR katsayısı kullanılarak eğitim yapılmıştır. Bulanık K-En Yakın Komşuluk Sınıflandırıcısında sınıflama oranı dört AR katsayısı kullanıldığında %64-%68, iki AR katsayısı kullanıldığında ise %53-%71 arasında verilmiştir (Şeker, 1995). Bulanık

K-En Yakın Komşuluk Sınıflayıcısında oranlar, K komşuluk sayısının 2 ile 6 arasında değişmesi halinde verilmektedir. Ayrıca Bulanık K-En Yakın Komşuluk Sınıflayıcısı bu sonuçlara 5000 iterasyon sonucunda ulaşırken Konik Kesit Fonksiyonlu Yapay Sinir Ağı, farklı merkez sayıları ve momentum katsayılarına bağlı olarak 7 ile 190 iterasyondan sonra ulaşmaktadır.

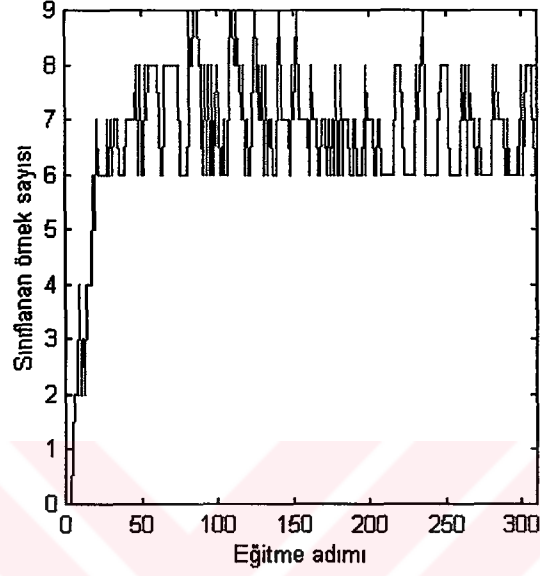
Üçüncü durumda 4-katlamalı çapraz-geçerlilik yöntemi uygulanmıştır. Denemeler sırasında her sınıfa ait 12 örnekten üçerli 4 alt küme oluşturulmuş ve sırasıyla bu alt kümelerden biri (yani, 3 örnek) test için dışarıda bırakılarak geriye kalan örneklerle ağ eğitilmiştir. Test verisi yakınsama kriterine bağlı olarak Konik Kesit Fonksiyonlu Yapay Sinir Ağı ile dört farklı grup için %50 ile %78 oranında sınıflanmakta, ayrıca iterasyon sayısının bazı durumlarda biraz arttığı görülmektedir. Bu tür bir genelleştirme Çok Katmanlı Algılayıcı ve Bulanık K-En Yakın Komşuluk Sınıflayıcısı için denenmemiştir. Fakat adaptif öğrenme oranı ve momentum kullanılmayan standart Konik Kesit Fonksiyonlu Yapay Sinir Ağı ile 4. üçlü gruba ait veriler eğitilmiş ve test edilmiştir. Sonuçta, bu grup için sınıflama oranının aynı olduğu ancak aynı sonuca yaklaşık üç kat fazla eğitime adımıyla ulaşıldığı görülmüştür. Adaptif öğrenmeli ve momentumlu Konik Kesit Fonksiyonlu Yapay Sinir Ağı ile elde edilen sonuçlar Çizelge 4.3'de verilmektedir. Her grup için eğitime adımına göre sınıflanmış örnek sayısını gösteren grafikler Şekil 4.6, 4.7, 4.8 ve 4.9'da gösterilmiştir.



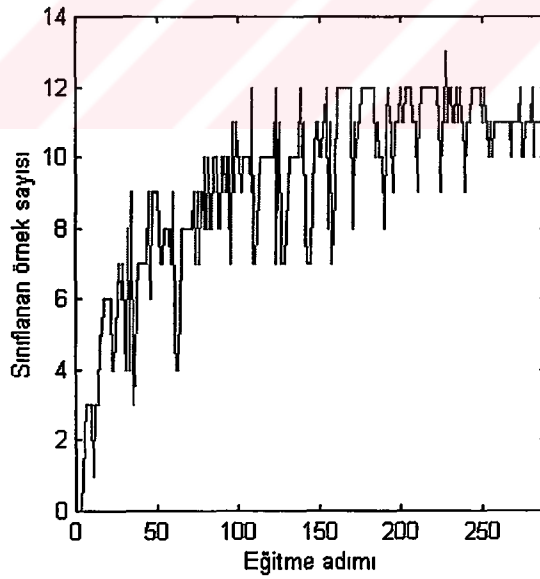
Şekil 4.6 4-katlamalı çapraz-geçerlilik yöntemine göre 1. üçlü grup için test sonucu

Çizelge 4.3 4-katlamalı çapraz-geçerlilik yöntemi ile elde edilen sonuçlar

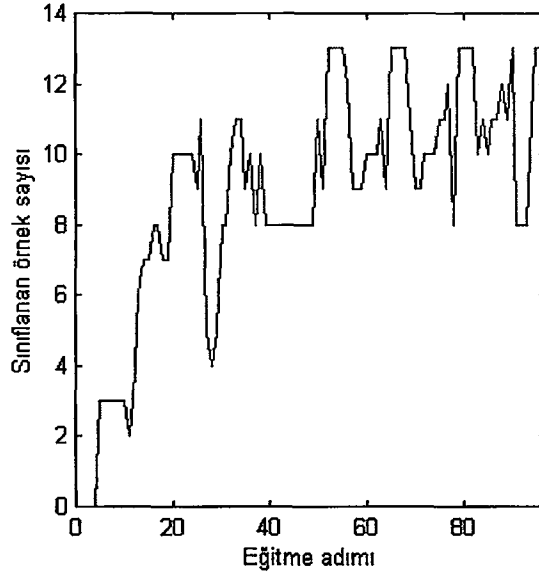
	1. üçlü grup	2. üçlü grup	3. üçlü grup	4. üçlü grup
Sınıflama Oranı	%56	%50	%72	%78
İterasyon Sayısı	145	90	230	90



Şekil 4.7 4-katlamalı çapraz-geçerlilik yöntemine göre 2. üçlü grup için test sonucu



Şekil 4.8 4-katlamalı çapraz-geçerlilik yöntemine göre 3. üçlü grup için test sonucu

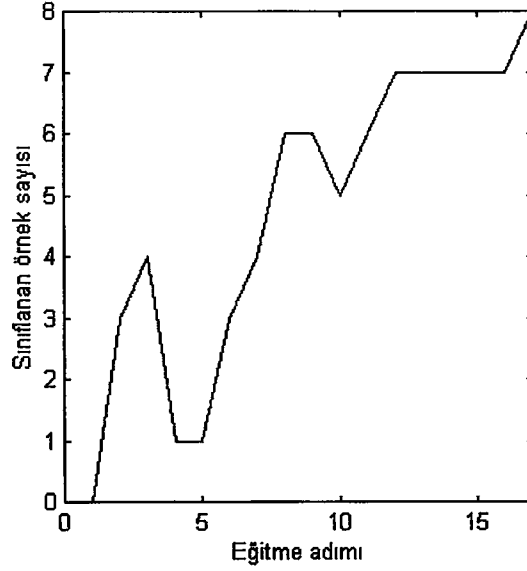


Şekil 4.9 4-katlamalı çapraz-geçerlilik yöntemine göre 4. üçlü grup için test sonucu

4.2 Kontakt Lens Uydurma Problemi

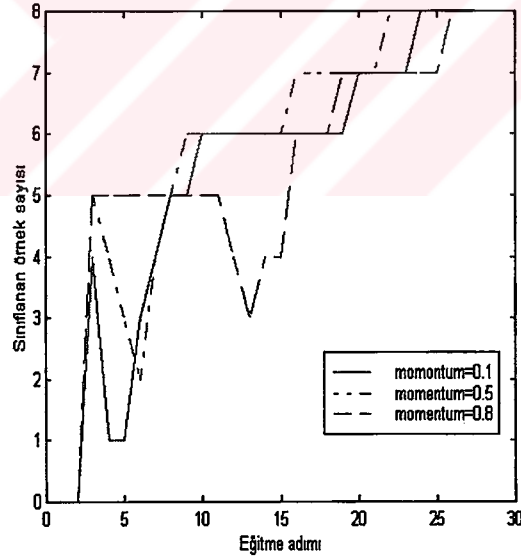
Konik Kesit Fonksiyonlu Yapay Sinir Ağını eğitmek için bir diğer uygulama olarak kontakt lens uydurma veritabanı test edilmiştir. Burada örnekler üç sınıfta toplanmakta, bu sınıflar kişiyle uyumlu olabilecek kontakt lens tipini vermektedir. Eğitim ve test kümeleri, toplam 24 örnekten oluşturulmaktadır. Bu örneklerin 15 tanesi üçüncü sınıfa, 5 tanesi ikinci sınıfa ve 4 tanesi de birinci sınıfa aittir. 16 örnek ağı eğitmek için; geriye kalan 8 örnek ise algoritmayı test etmek için kullanılmıştır. Her bir örnek dört parametre ile temsil edilmektedir. Bunlar: Kişinin yaşının genç, orta yaşlı veya yaşlı olmak üzere hangi gruba girdiği; görme bozukluğunun miyopi mi, hipermetropi mi olduğu; astigmatlığın olup olmadığı; göz yaşı üretme oranının azalmış veya normal olması durumlarıdır. Veri kümesinin detayları Ek 2'de verilmiştir. Bu veri kümesi nispeten basit ve eğitilmesi çok hızlı olduğu için, adaptif öğrenmeli CSFNN'nin performansını belirleyecek çeşitli parametrelerin denenmesinde kullanılmıştır.

Daha önce standart CSFNN (Yıldırım ve Marsland, 1997) ile yapılan çalışmada, test verisi 76 eğitim adımında ağ tarafından tamamen tanınmışken; adaptif algoritma ve momentum katsayısının kullanıldığı durumda uygun bir merkez değerinin belirlenmesi ile tüm test verisinin sadece 18 eğitim adımında ağ tarafından tanındığı görülmüştür. Buna ait sonuç Şekil 4.10'da verilmiştir.



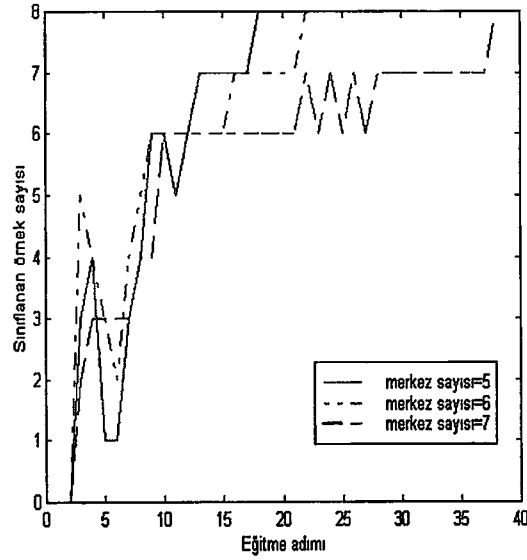
Şekil 4.10 Kontakt lens uydurma problemi için $mom=0.5$ ve 5 merkez kullanıldığında elde edilen sonuç

Şekil 4.11 farklı momentum katsayıları için merkez sayısı 6 olarak seçildiğinde elde edilen sonuçları göstermektedir. Buna göre momentum=0.5 seçildiğinde ağın daha az eğitim adımında test verisini doğru olarak sınıfladığı görülmüştür



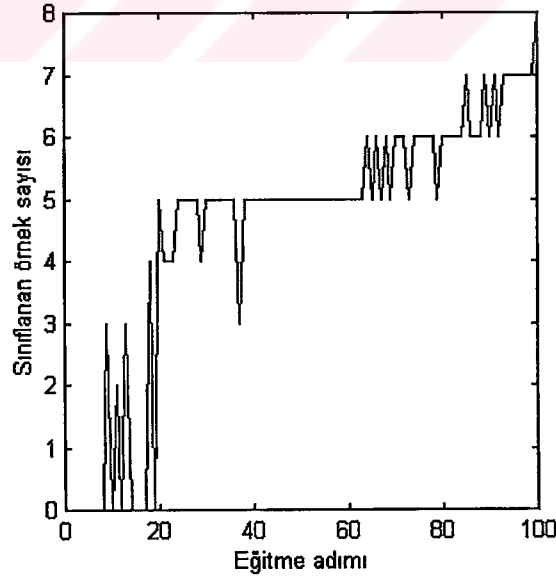
Şekil 4.11 Farklı momentum katsayıları için elde edilen sonuçlar

Adaptif öğrenmeli CSFNN ağında farklı merkez sayılarının öğrenmeyi nasıl etkilediği de ayrıca incelenmiştir. Bu amaçla, en iyi performansı veren momentum katsayısı (0.5) kullanılarak ağ, üç farklı merkez değeri seçilerek eğitilmiş ve test edilmiştir. Buna ait sonuçlar Şekil 4.12'de verilmiştir. Şekilden de görüldüğü gibi örneklerin tanınması için 5 merkez değerinin yeterli olduğu hatta en iyi sonucu verdiği görülmüştür.



Şekil 4.12 Farklı merkez sayıları kullanılarak elde edilen sonuçlar

Buraya kadar verilen sonuçlar, öğrenme aşamasındaki başlangıç fazında sabit bir merkez sayısı seçerek bu sayıya ulaşmaya kadar OLS algoritması ile merkez değerlerinin belirlenmesi yöntemi ile elde edilmiştir. İkinci yöntem olarak, başlangıç fazında merkezleri sıfıra atayıp eğitmenin devamında bu merkezlerin geriye yayılma algoritması ile güncelleştirilebileceği ifade edilmişti (Bölüm 3.4). Bu yöntem uygulandığında, örneklerin doğru sınıflanması için daha fazla eğitime adımının gerekli olduğu görülmüştür (Şekil 4.13).



Şekil 4.13 Başlangıç fazında merkezlerin sıfıra atanması ile elde edilen sonuç

Son olarak, elde edilen sonuçlar aynı veri kümesini kullanarak farklı ağ yapıları ile bulunan sonuçlarla karşılaştırılmıştır (Çizelge 4.4). Çizelgede geriye yayılmalı ve hızlı geriye

yayılmalı MLP ağları için verilen değerler algoritmaların birden fazla çalıştırılması sonucunda alınan ortalama değerlerdir (Yıldırım, 1997). Bu algoritmalarda ağırlıkların rasgele başlatılmasından dolayı sonuçlar kararlı değildir. CSFNN'de ise her çalıştırmadan sonra aynı sonuç elde edilmektedir.

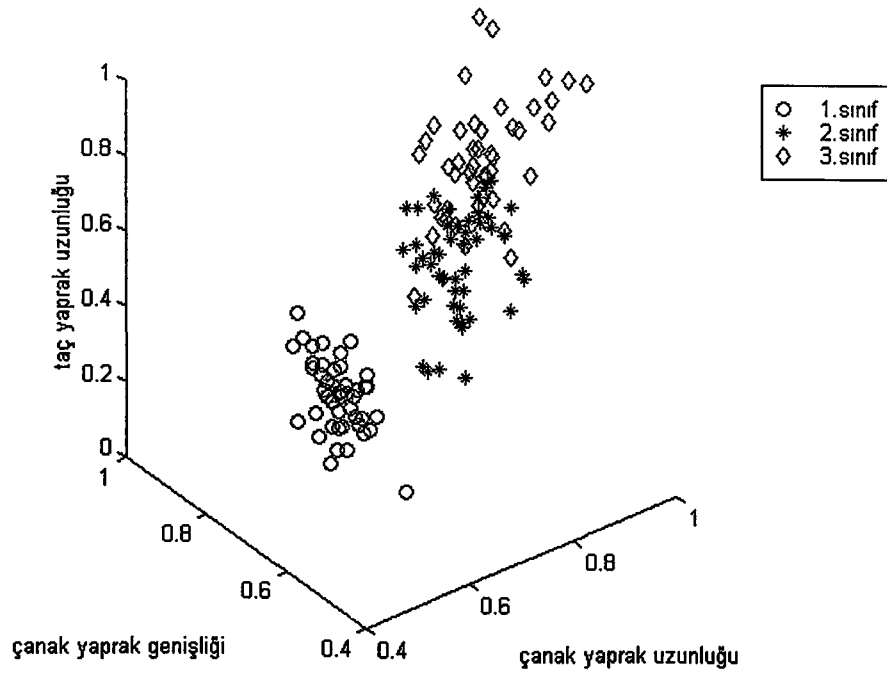
Çizelge 4.4 Farklı algoritmalar için elde edilen sonuçlar

Kontakt Lens Veritabanı	Eğitme adımı	Sınıflama oranı
Standart CSFNN	76	%100
Adaptif öğrenmeli CSFNN (merkezler OLS ile seçildiğinde)	18	%100
Adaptif öğrenmeli CSFNN (merkezler sıfırdan başlatıldığında)	100	%100
Geriye yayılmalı MLP	2187 (ort)	%100
Hızlı geriye yayılmalı MLP	120 (ort)	%100

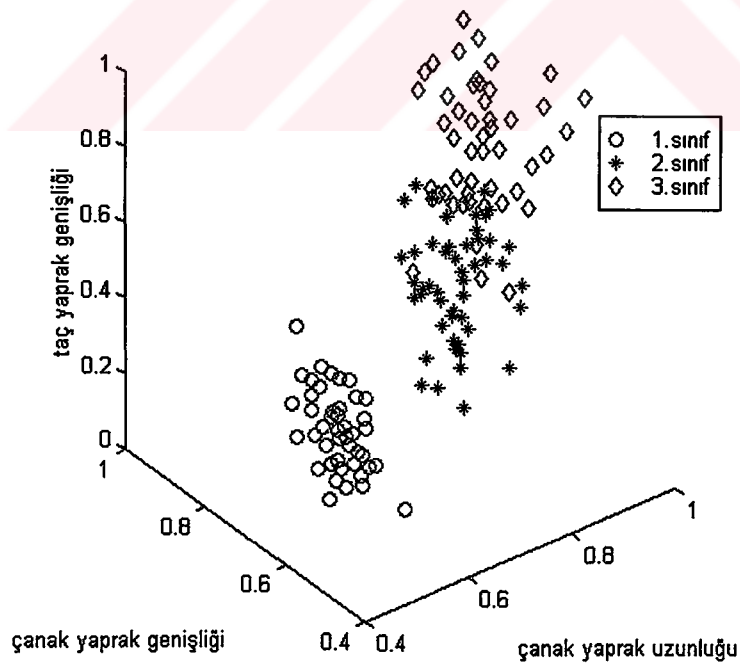
4.3 İris Bitkisi Sınıflandırma Problemi

Literatürde patern tanıma problemlerinde İris bitkisi sınıflandırma veritabanı oldukça sık kullanılmaktadır (Fisher, 1936). Bu veritabanı, her biri 50 örnek içeren üç sınıftan meydana gelmektedir. Bu sınıflar sırasıyla; Setosa, Versicolor ve Virginica olup İris bitkisinin tipini göstermektedir. Birinci sınıf, Setosa, diğer sınıflardan lineer olarak ayrılabilir; ancak diğer iki sınıf lineer olarak ayrılamaz. Her örnek dört öznelik ile tanımlanmıştır. Bunlar; çanak yaprak uzunluğu, çanak yaprak genişliği, taç yaprak uzunluğu ve taç yaprak genişliğidir. Ek 3'de veri kümesinin detayları verilmiştir. Farklı özneliklere göre sınıfların dağılımı Şekil 4.14, 4.15, 4.16 ve 4.17'de gösterilmiştir.

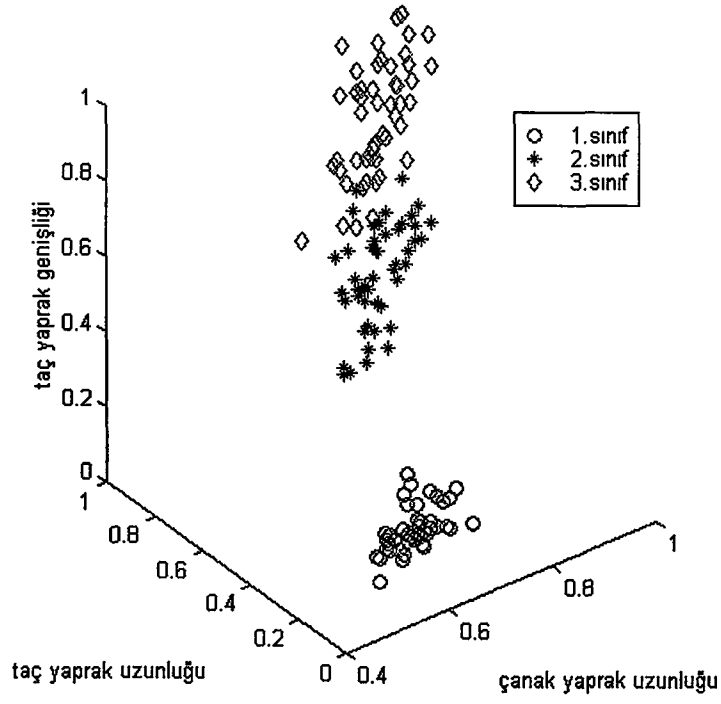
Bu veri tabanının eğitilmesi sırasında 2-katlamalı ve 5-katlamalı çapraz-geçerlilik yöntemleri kullanılmıştır. 2-katlamalı çapraz-geçerlilik yönteminde, her sınıftan 25 örnek olmak üzere, 75 örnek ağı eğitilmesi; geri kalan 75 örnek de test edilmesi için kullanılacak şekilde iki alt küme (A ve B) oluşturulmuştur. 5-katlamalı çapraz-geçerlilik yönteminde ise, her sınıfa ait 50 örnekten onluk beş alt küme (A1, A2, A3, A4, A5) oluşturulmuş ve sırasıyla bu alt kümelerden biri (yani, 30 örnek) test için dışarıda bırakılarak geriye kalan örneklerle ağ eğitilmiştir. Bu yöntemlere ait sonuçlar Çizelge 4.5'de özetlenmiştir.



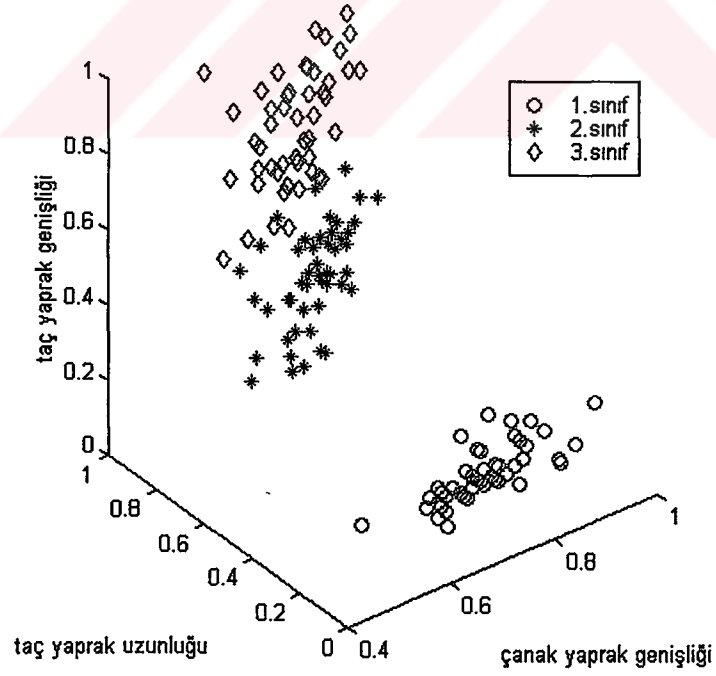
Şekil 4.14 İris bitkisine ait sınıfların 1., 2. ve 3. öz niteliklere göre dağılımı



Şekil 4.15 İris bitkisine ait sınıfların 1., 2. ve 4. öz niteliklere göre dağılımı



Şekil 4.16 İris bitkisine ait sınıfların 1., 3. ve 4. özniteliklere göre dağılımı



Şekil 4.17 İris bitkisine ait sınıfların 2., 3. ve 4. özniteliklere göre dağılımı

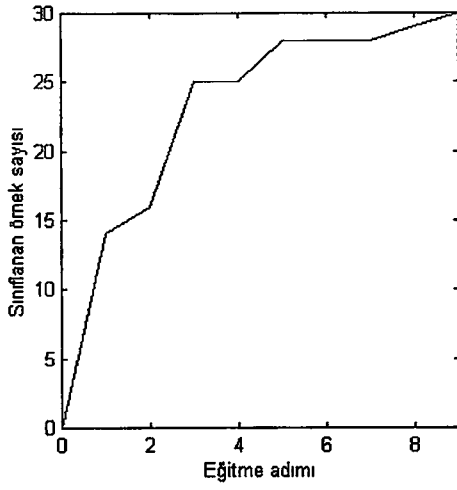
Çizelge 4.5 2-katlamalı ve 5-katlamalı çapraz-geçerlilik yöntemlerine ait sonuçlar

Alt Küme Adı	Eğitme Kümesindeki Örnek Sayısı	Test Kümesindeki Örnek Sayısı	Sınıflanan Örnek Sayısı	Sınıflama Oranı	Eğitme Adımı Sayısı
A1	120	30	30	%100	9
A2	120	30	30	%100	10
A3	120	30	29	%96.67	25
A4	120	30	29	%96.67	6
A5	120	30	30	%100	5
A	75	75	73	%97.33	44
B	75	75	74	%98.66	613

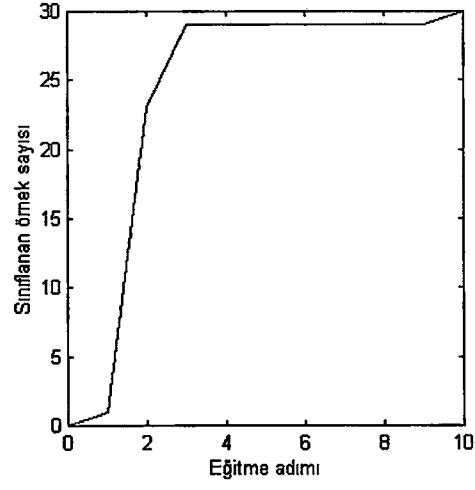
Adaptif öğrenmeli Konik Kesit Fonksiyonlu Yapay Sinir Ağının genelleştirme yeteneğinin incelenmesi amacı ile İris bitkisi sınıflandırma problemi için 2-katlamalı ve 5-katlamalı çapraz-geçerlilik yöntemleri uygulandığında her bir grup için elde edilen grafikler Şekil 4.18 ve 4.19'da gösterilmiştir. Bu yöntemlerle ağın eğitilmesi sırasında, merkez sayısı 6 seçilerek OLS algoritması ile bu merkezlerin yerleri belirlenmiştir. Ayrıca momentum katsayısı 0.1 olarak seçilmiştir. 5-katlamalı çapraz-geçerlilik yönteminde tüm alt kümeler için elde edilen sonuçların ortalaması alındığında sınıflama doğruluğunun %98.668 olduğu görülmüştür. Aynı yöntemle k-NN algoritması için elde edilen en iyi sınıflama doğruluğu %95.34, k-NNFP algoritması için ise %96.02'dir (Ayan, 1999). Buna göre adaptif öğrenmeli CSFNN'nin daha iyi bir genelleştirme yeteneğine sahip olduğu söylenebilir. Karşılaştırmalı sonuçlar Çizelge 4.6'da verilmiştir.

Çizelge 4.6 Farklı algoritmalar için elde edilen sonuçlar

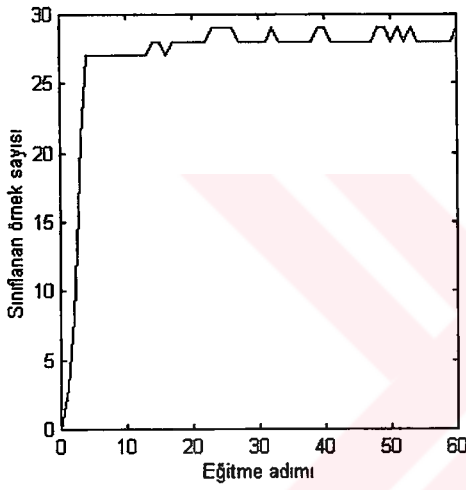
İris Veritabanı (5-katlamalı çapraz geçerlilik yöntemi)	Sınıflama oranı
Adaptif öğrenmeli CSFNN	%98.668
k-NN	%95.34
k-NNFP	%96.02



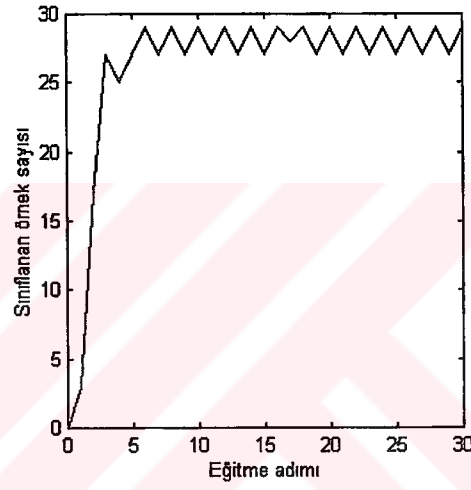
A1



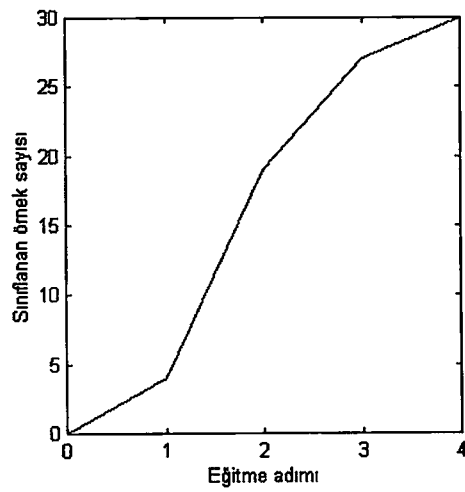
A2



A3

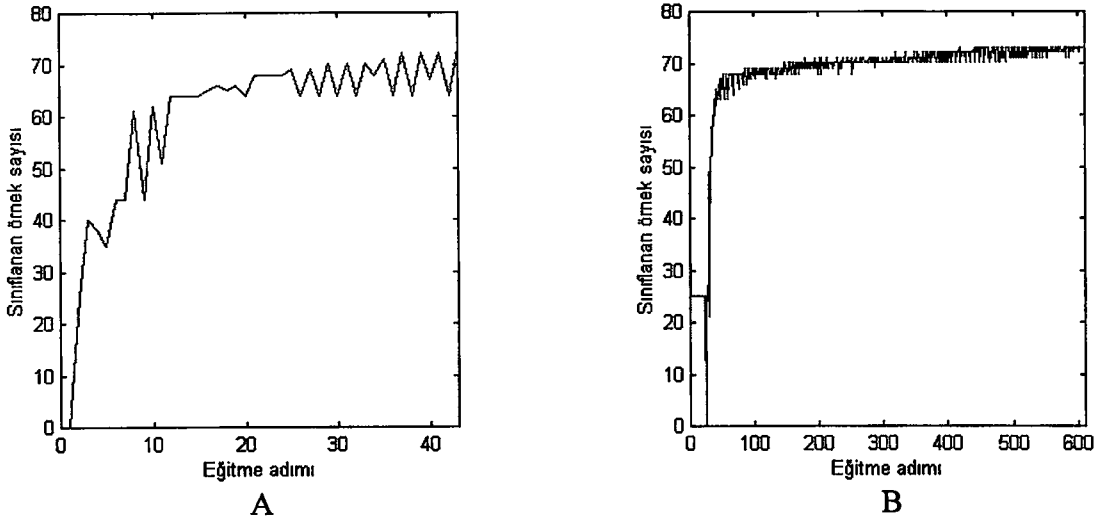


A4



A5

Şekil 4.18 5-katlamalı çapraz-geçerlilik yöntemi ile gruplara göre elde edilen sonuçlar



Şekil 4.19 2-katlamalı çapraz-geçerlilik yöntemi ile gruplara göre elde edilen sonuçlar

Çizelge 4.7 Farklı algoritmalar için elde edilen sonuçlar

İris Veritabanı A Grubu (2-katlamalı çapraz geçerlilik yöntemi)	Eğitime adımı	Sınıflama oranı
Standart CSFNN	460	%97.33
Adaptif öğrenmeli CSFNN	44	%97.33
Geriye yayımlı MLP	140 (ort.)	%94.66 (ort.)
Hızlı geriye yayımlı MLP	58 (ort.)	%94.66 (ort.)

Çizelge 4.8 Farklı algoritmalar için elde edilen sonuçlar

İris Veritabanı B Grubu (2-katlamalı çapraz geçerlilik yöntemi)	Eğitime adımı	Sınıflama oranı
Standart CSFNN	4621	%98.66
Adaptif öğrenmeli CSFNN	613	%98.66
Geriye yayımlı MLP	400 (ort.)	%94.66 (ort.)
Hızlı geriye yayımlı MLP	105 (ort.)	%96 (ort.)

2-katlamalı çapraz-geçerlilik yöntemi standart CSFNN için de denenmiştir. Burada, sınıflama doğruluğunun adaptif CSFNN ile aynı olduğu fakat eğitime adımının arttığı görülmektedir. Ayrıca aynı yöntem geriye yayımlı MLP ve hızlı geriye yayımlı MLP için de

uygulanmıştır. Çizelge 4.7 ve 4.8'de karşılaştırmalı sonuçlar verilmektedir. Çizelgelerde geriye yayılmanın kullanıldığı algoritmalara ait sonuçlar farklı çalıştırmalar sonucunda elde edilen sonuçların ortalamasıdır.

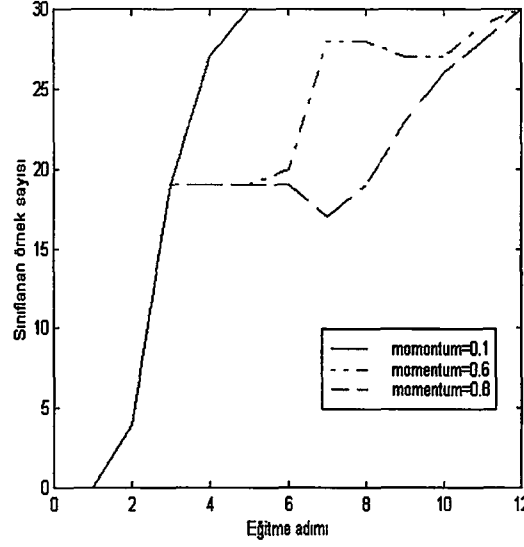
Çizelge 4.9'da 5-katlamalı çapraz-geçerlilik yönteminde kullanılan A5 grubuna ait sonuçlar, daha önce aynı gruba elde edilen standart Konik Kesit Fonksiyonlu Yapay Sinir Ağının sonuçları (Yıldırım, 1997) ile karşılaştırılmıştır. Buna göre, standart CSFNN ile 6 merkez için en iyi durumda 503 adımda sonuca ulaşılırken adaptif öğrenmeli CSFNN ile sınıfların ayrımı sadece 5 adımda yapılabilmektedir. Yine aynı kaynakta (Yıldırım, 1997) verilen sonuçlarda, geriye yayımlı MLP'nin rasgele başlatılan ağırlıklara göre 4000 ila 30000 eğitime adımında; hızlı geriye yayımlı MLP'nin ise 800-20000 eğitime adımından sonra örnekleri doğru olarak sınıfladığı bildirilmiştir. Ayrıca bazı durumlarda hızlı geriye yayımlı MLP'nin sonuca yakınsamadığı da not edilmiştir. Aynı kaynakta standart RBF ağının da ancak 32 merkez yerleştirdikten sonra aynı grup için örnekleri sınıfladığı bildirilmiştir. Oysa standart CSFNN ve adaptif öğrenmeli CSFNN, sınıfları sadece 6 merkez kullanarak ayırabilmektedir. Bu sonuç, CSFNN yapısının elektronik tasarımlar açısından daha az kompleks bir devre ile gerçekleştirilebileceğini göstermektedir.

Çizelge 4.9 Farklı algoritmalar için elde edilen sonuçlar

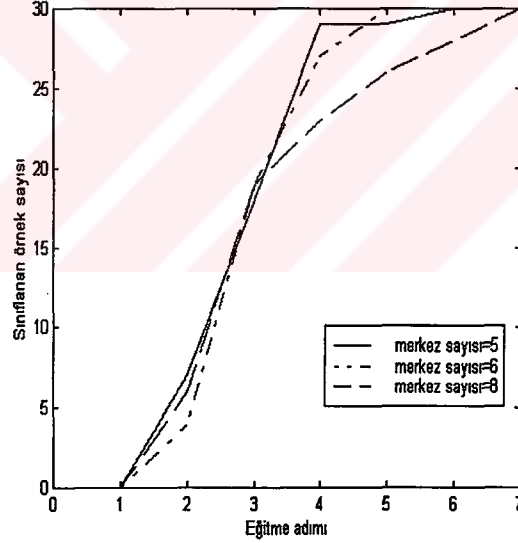
İris Veritabanı A5 Grubu	Eğitime adımı	Sınıflama oranı
Standart CSFNN	503	%100
Adaptif öğrenmeli CSFNN	5	%100
Geriye yayımlı MLP	9112 (ort)	%100
Hızlı geriye yayımlı MLP	7216 (ort)	%100

İris bitkisi sınıflandırma problemi için adaptif öğrenmeli CSFNN algoritmasında farklı momentum katsayılarının ve merkez sayılarının öğrenmeyi nasıl etkilediği de ayrıca incelenmiştir. Şekil 4.20, farklı momentum katsayıları için merkez sayısının 6 olarak seçilmesi halinde elde edilen sonuçları göstermektedir. Buna göre momentum=0.1 seçildiğinde ağın daha az eğitime adımında ve daha kararlı bir şekilde test verisini doğru olarak sınıfladığı görülmüştür. Bu sonuçlardan yararlanarak en iyi performansı veren momentum katsayısı (0.1) ile ağ, üç farklı merkez değeri seçilerek eğitilmiş ve test edilmiştir.

Buna ait sonuçlar da Şekil 4.21'de verilmiştir. Şekilden de görüldüğü gibi örneklerin tanınması için 5 merkez değerinin atanması yeterli olmakta, ancak 6 merkez atandığında algoritma en iyi sonucu vermektedir.



Şekil 4.20 Farklı momentum katsayıları için elde edilen sonuçlar



Şekil 4.21 Farklı merkez sayıları için elde edilen sonuçlar

4.4 Fetus Gelişimi Problemi

Bu uygulamada amaç, fetus gelişiminin normal veya anormal olup olmadığının belirlenmesidir. Bunun için, gebelik sırasında fetusla plesanta arasındaki bağlantıyı sağlayan ve anneden gıda alımı ile bebekten atıkların temizlenmesini sağlayan bağ damarlarındaki (Umbilical Artery-UA) kan akış hızı dalga şekillerinin dopler ultrason ölçüm sonuçları

kullanılmıştır. Haftalık ultrason ölçüm sonuçları ağa giriş bilgisi olarak verilmiş ve fetus gelişimi hakkında çıkış bilgisi elde edilmiştir.

Örnekler dört parametre ile temsil edilmektedir. İlk parametre, hafta olarak gebelik süresidir (Week Index-WI). Diğerleri ise; ultrason ışını ile kan akış yönü arasındaki açıdan bağımsız olan, kan akış hızı dalga şekilleri ölçüm sonuçlarıdır. Bu açıdan bağımsız indeksler aşağıda verilmiştir:

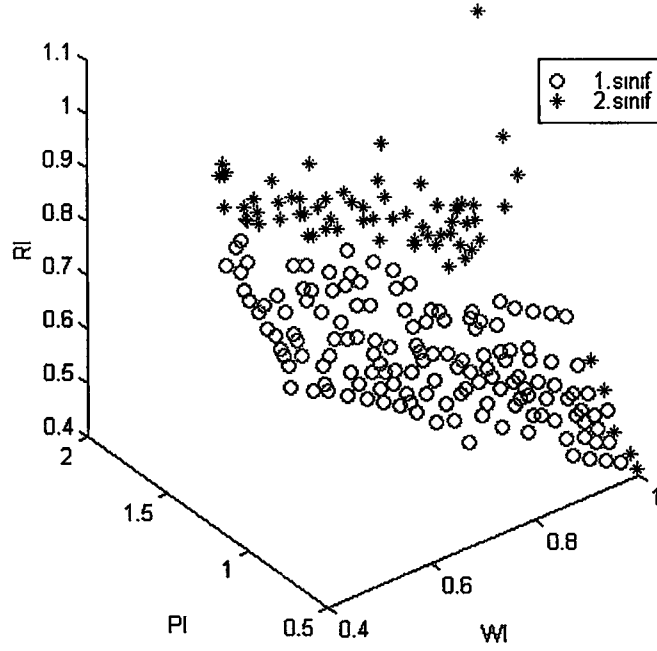
1. Sistolik/Diastolik (S/D) oranı
2. Direnç İndeksi (Resistance Index-RI)
3. Pulsatility Index (PI)

Burada $PI=(S-D)/\text{ortalama hız}$, $RI=(S-D)/D$ olarak tanımlanmaktadır. S/D, RI, PI parametrelerindeki düşüş, gebeliği olumlu yönde etkiler. Bir başka deyişle, anormal UA dalga şekli normal bir gebelik için elde edilen aralığın üstünde bir PI (ya da RI, ya da S/D oranı) değerine sahiptir. Doğumda genellikle yüksek bir S/D oranı, fetusun küçük boyutlu olmasına, yüksek hastalık veya ölüm oranına yol açar. Ek 4'de veri kümesinin detayları verilmiştir (Güler vd., 1999). Farklı öz niteliklere göre sınıfların dağılımı Şekil 4.22, 4.23, 4.24 ve 4.25'de gösterilmiştir.

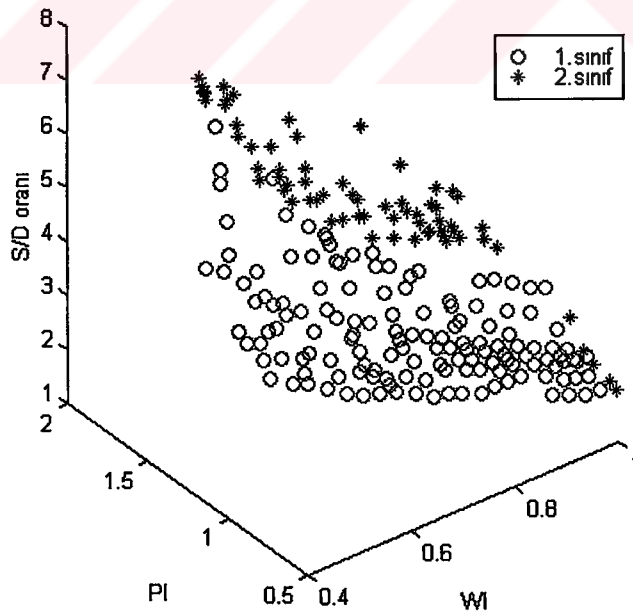
143'ü normal, 67'si anormal olmak üzere toplam 200 örnek ele alınmıştır. Bunların 147 tanesi ağın eğitilmesinde, 63 tanesi de test edilmesinde kullanılmıştır. Sonuçlar, daha önce MLP kullanılarak yapılan bir çalışmada (Güler vd., 1999) elde edilen sonuçlarla karşılaştırılmıştır (Çizelge 4.10).

Çizelge 4.10 Fetus gelişimi problemi için sonuçların karşılaştırılması

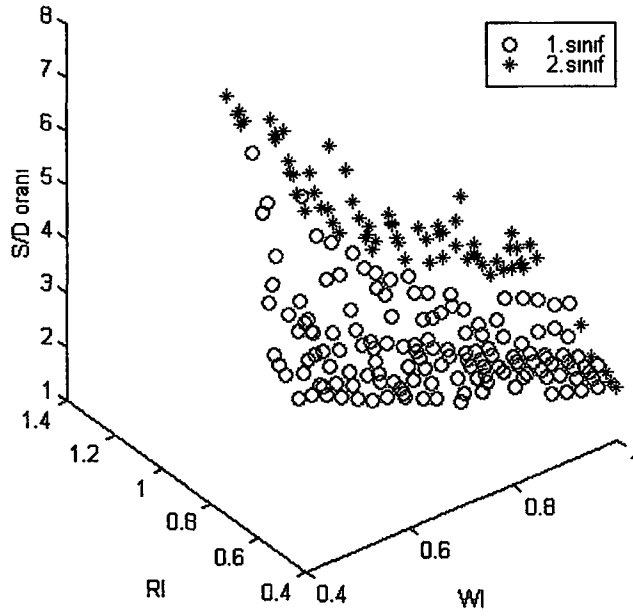
Fetus Gelişimi Veritabanı	Sınıflama oranı
Adaptif öğrenmeli CSFNN (6 gizli birim)	%100
Geriye yayılmalı MLP (8 gizli birim)	%90.47
Geriye yayılmalı MLP (12 gizli birim)	%95.23



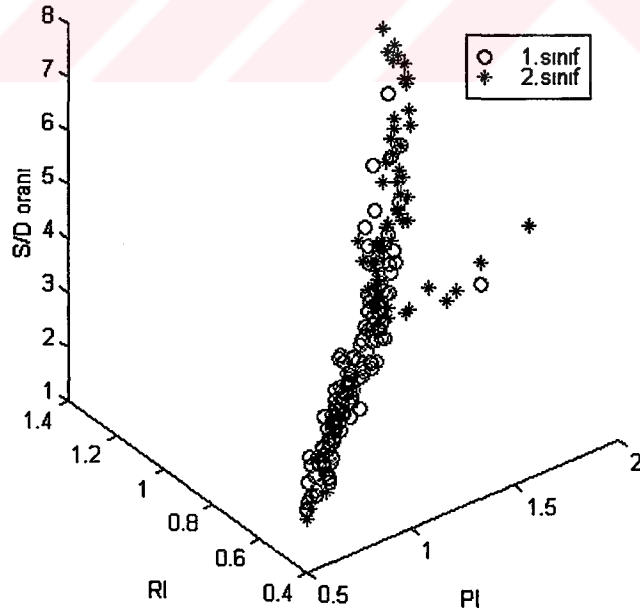
Şekil 4.22 Fetus gelişimine ait sınıfların WI, PI ve RI parametrelerine göre dağılımı



Şekil 4.23 Fetus gelişimine ait sınıfların WI, PI ve S/D oranı parametrelerine göre dağılımı

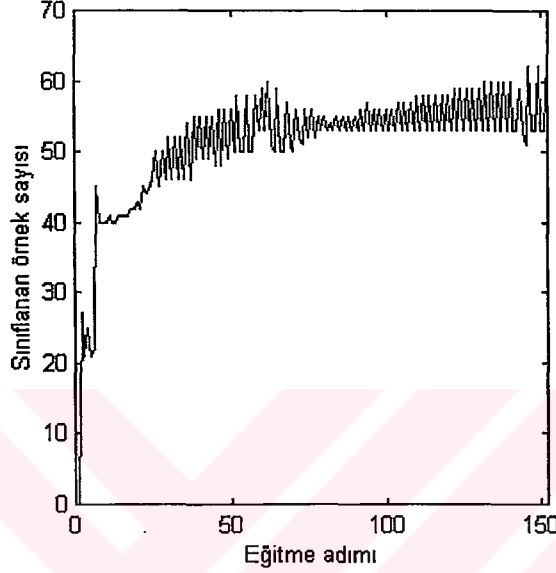


Şekil 4.24 Fetus gelişimine ait sınıfların WI, RI ve S/D oranı parametrelerine göre dağılımı



Şekil 4.25 Fetus gelişimine ait sınıfların PI, RI ve S/D oranı parametrelerine göre dağılımı

Çizelge 4.10 incelendiğinde MLP ile farklı gizli katman düğüm sayıları için en fazla %95.23 doğrulukla sınıflama yapıldığı görülmektedir. Yani, test edilen 63 örnekten 60 tanesi MLP ağı tarafından tanınmaktadır. Oysa adaptif öğrenmeli Konik Kesit Fonksiyonlu Yapay Sinir Ağı, test edilen bütün örnekleri %100 doğrulukla sınıflamaktadır. Bu sonuca göre, adaptif öğrenmeli CSFNN algoritmasının tıbbi teşhis amacıyla da kullanılabileceği söylenebilir. Bu algoritmaya ait sonuç Şekil 4.26'da verilmiştir.

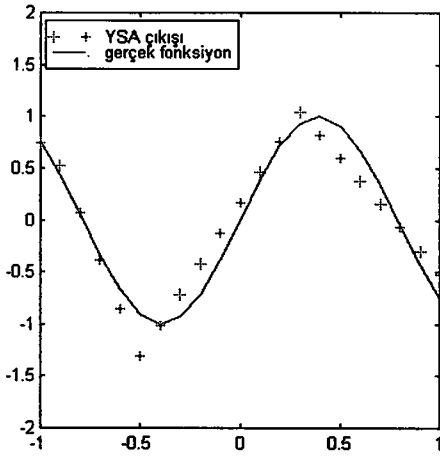


Şekil 4.26 Merkez sayısı=6 ve momentum katsayısı=0.5 için elde edilen sonuç

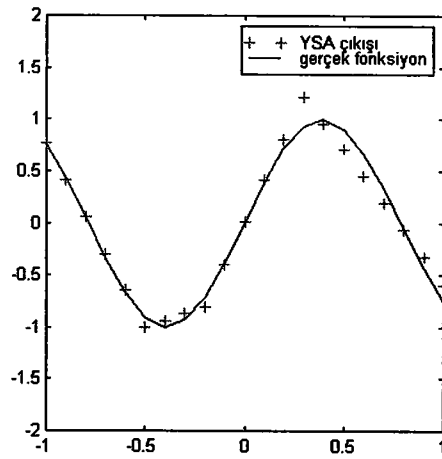
4.5 Fonksiyon Yaklaşımı Problemi

Bu bölüme kadar, geliştirilen adaptif öğrenmeli CSFNN çeşitli sınıflama problemlerinde incelendi. Bu bölümde ise bu ağın fonksiyon yaklaşırma işlemi için de kullanılıp kullanılamayacağı ayrı bir problem olarak ele alınmıştır. Yapay sinir ağlarının fonksiyon yaklaşımcılar olarak incelenmesi çeşitli alanlarda kullanılmaktadır. Örneğin; kontrol sistemlerinde amaç, ölçülen çıkışlardan kontrol girişlerine bir ilişki sağlayacak uygun bir geri besleme fonksiyonu bulmaktır. Benzer şekilde, adaptif filtreleme de amaç, bir giriş işaretinin gecikmiş değerlerinden uygun bir çıkış işaretine ilişki sağlayacak bir fonksiyon bulmaktır. Literatürde fonksiyonları gerçekleştirmek için Çok Katmanlı Algılayıcı ve Radyal Temelli Fonksiyonlu ağ yapılarından yararlanılmıştır. (Hagan vd., 1996)

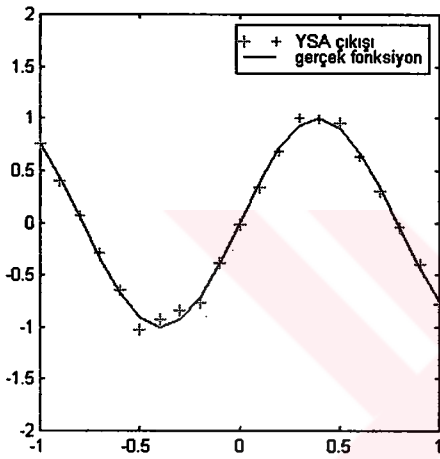
Burada x , $[-1,1]$ aralığında 0.1 adımla değişmek üzere $y=\sin(4x)$ fonksiyonu ele alınmıştır. Verilen aralıktaki x değerleri ağa giriş bilgisi, buna karşı gelen fonksiyon değerleri de hedef bilgisi olarak verilmiştir. Bu bilgilerle ağ belli bir hata değerine kadar eğitildikten sonra test



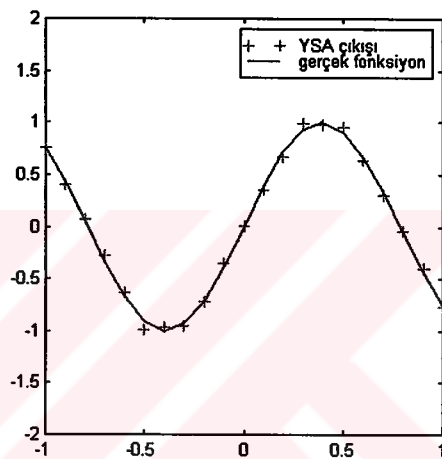
4 merkezli ağ



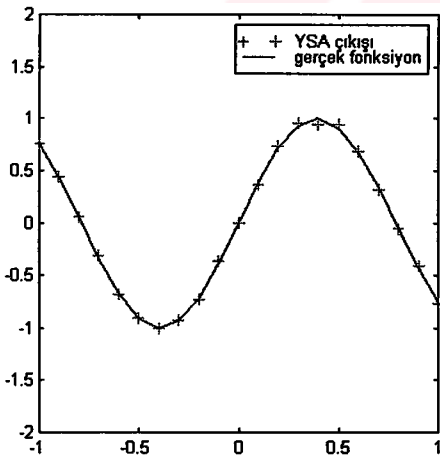
5 merkezli ağ



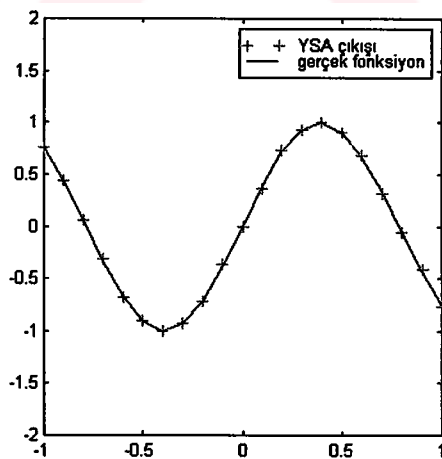
7 merkezli ağ



8 merkezli ağ



12 merkezli ağ

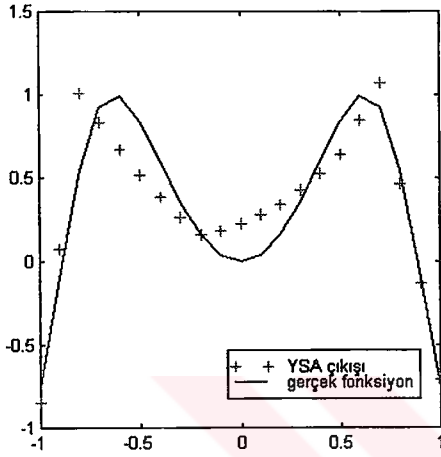


13 merkezli ağ

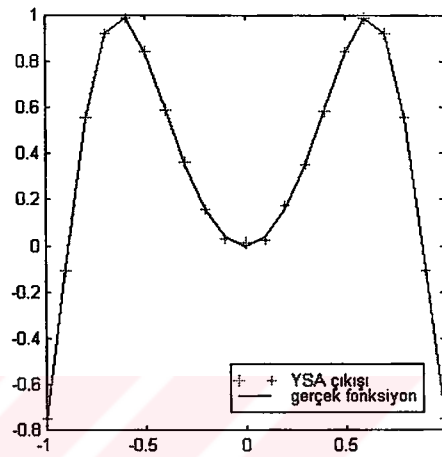
Şekil 4.27 Farklı merkez değerleri için $y=\sin(4x)$ fonksiyonunu yaklaştırma probleminin sonuçları

edilmiştir. Ağı test etmek için 0.1 adımla -1.05 ile 1.05 aralığındaki değerler ağı girilerek çıkış hesaplanmıştır. Gizli katmandaki birim sayısının yani belirlenen merkez sayısının farklı değerleri için yakınsamanın değiştiği görülmüştür. Bunlara ilişkin sonuçlar Şekil 4.27'de gösterilmiştir.

İkinci bir fonksiyon olarak $y=\sin(4x^2)$ ele alınarak farklı merkez değerleri için elde edilen sonuçlar Şekil 4.28'de verilmiştir.



6 merkezli ağ



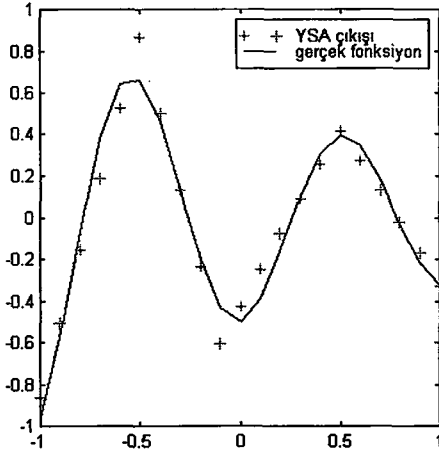
14 merkezli ağ

Şekil 4.28 Farklı merkez değerleri için $y=\sin(4x^2)$ fonksiyonunu yaklaştırma probleminin sonuçları

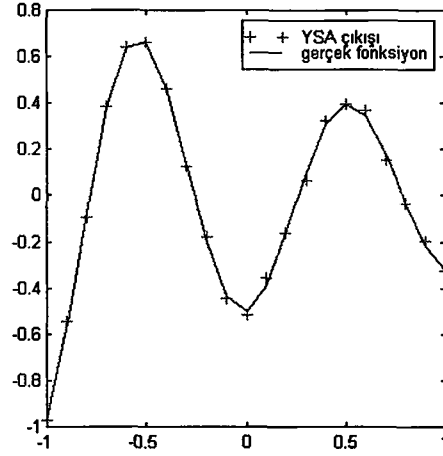
Çalışmada ayrıca Matlab Neural Networks Toolbox'ı içerisindeki fonksiyon yaklaşırtıcı algoritmalar (Geriye yayılma, Hızlı geriye yayılma, Levenberg Marquardt öğrenme algoritmaları ile Çok Katmanlı Algılayıcı yapısı ve OLS algoritması ile eğitilen Radyal Temelli Fonksiyonlu ağ yapısı) incelenmiş ve adaptif öğrenmeli CSFNN ile karşılaştırılmıştır (Çizelge 4.11). Adaptif öğrenmeli CSFNN'ye ait sonuçlar Şekil 4.29'da verilmiştir.

Çizelge 4.11 Fonksiyon yaklaşımı için farklı algoritmaların sonuçları

Algoritmalar	Eğitme Adımı	Eğitme Zamanı	İşlem Sayısı
Geriye yayılma	2783 (ort)	18.5 (ort)	14596683 (ort)
Hızlı geriye yayılma	532 (ort)	3.9 (ort)	3515312 (ort)
Levenberg Marquart	6 (ort)	0.4 (ort)	2804143 (ort)
RBF	6	0.5	37133
Adaptif CSFNN	12	1.04	108671



6 merkezli ağ



12 merkezli ağ

Şekil 4.29 Matlab Neural Networks Toolbox'ı içerisindeki fonksiyon kullanılarak elde edilen sonuçlar

4.6 Mikrodalga Transistörünün YSA ile modellenmesi

Bir mikrodalga transistörünün kutuplama noktası civarındaki küçük işaret ve gürültüye karşı olan davranışları çalışma bandı boyunca sırasıyla saçılma S_{11} , S_{22} , S_{21} , S_{12} ve gürültü F_{opt} , Γ_{opt} , R_N parametreleri ile belirlenebilir. Gürültü ve S parametrelerinin her ikisi de frekansa, kutuplama noktasına, konfigürasyon tipine ve elemanın fiziksel özelliklerine bağlıdır. S parametreleri, elemanın işaret güç kazançları ve giriş-çıkış kapı uyumsuzluk kayıplarını tayin ederken, gürültü parametreleri de elemanın girişi ile çıkışı arasındaki işaret/gürültü oranı kötüleşmesini belirler. Belli bir konfigürasyon tipinde, bir kutuplama noktası civarındaki mikrodalga transistörünün küçük-ışaret ve gürültü performansı ω -domeninde S saçılma ve N gürültü parametre vektörleri ile verilir. Ölçülmüş performans parametre verisi aşağıdaki gibi bir formda gösterilir.

$$\begin{bmatrix} f_1 : S^{(1)} N^{(1)} \\ f_2 : S^{(2)} N^{(2)} \\ \vdots \\ f_N : S^{(N)} N^{(N)} \end{bmatrix}$$

Burada $S^{(1)}$, $N^{(1)}$, ..., $S^{(N)}$, $N^{(N)}$, sırasıyla saçılma ve gürültü vektörleri; $f_1, \dots, f_N$ örnek çalışma frekanslarıdır. $S^{(N)}$ ve $N^{(N)}$ performans vektörleri bir defa hesaplandıktan sonra, aktif mikrodalga elemanı, f_N frekansında temsil edilebilir. (Torpi, 1998)

Çoklu kutuplama ve farklı konfigürasyon tipleri için, bir mikrodalga transistörünün işaret ve gürültü parametreleri YSA ile modellenebilir. Bu bölümde fonksiyon yaklaşımının özel bir hali olarak bir mikrodalga transistörünün YSA ile modellenmesi ele alınmıştır. Frekans, kutuplama noktası ve konfigürasyon tipi ağa giriş bilgisi olarak verilir; bunlara karşı gelen saçılma parametreleri de hedef çıkış olarak alınır ve ağ bu giriş-çıkış ilişkisini sağlamak amacı ile eğitilir. Yani giriş vektörü, f_N frekansını, (V_{CE} , I_{CE}) kutuplama noktasını ve konfigürasyon tipini içeren 4 boyutlu, çıkış vektörü ise S parametrelerini içeren 8 boyutlu bir vektördür.

NE21900 mikrodalga transistörü için veriler (Ek 5), $V_{CE} = 8V$, $I_C = 30mA$ ve $10mA$, CE (Common Emitter-Ortak Emetör) ve CC (Common Collector-Ortak Kolektör) konfigürasyon tipleri kullanılarak 2-6 GHz frekans aralığında, adaptif öğrenmeli Konik Kesit Fonksiyonlu Yapay Sinir Ağı kullanılarak eğitildi ve test edildi. Sonuçlar Çizelge 4.12'de verilmiştir.

Çizelge 4.12 NE21900 mikrodalga transistörü için 2-6GHz frekans aralığı ve $I_C = 20mA$ kutuplama noktasında elde edilen hata sonuçları

Konfigürasyon Tipi	Parametreler		Toplam Karesel Hata
CE	S_{11}	Büyüklük	0.0041
		Açı	0.0053
	S_{21}	Büyüklük	0.0052
		Açı	0.0171
	S_{12}	Büyüklük	0.0102
		Açı	0.0080
	S_{22}	Büyüklük	0.0386
		Açı	0.0231
CC	S_{11}	Büyüklük	0.0052
		Açı	0.0064
	S_{21}	Büyüklük	0.0023
		Açı	0.0210
	S_{12}	Büyüklük	0.0131
		Açı	0.0034
	S_{22}	Büyüklük	0.0382
		Açı	0.0077

4.7 Karakter Tanıma Problemi

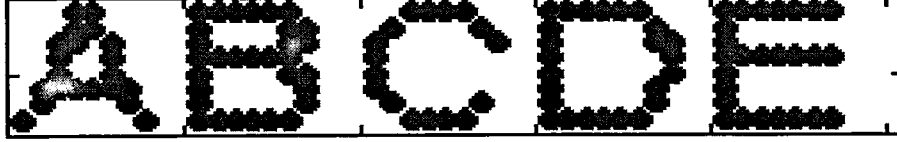
Çok katmanlı ileri beslemeli ağların limitlerini ve yeteneklerini belirleyebilmek için basılı karakterlerin tanınması iyi bir performans değerlendirme problemi olarak kullanılmaktadır. (Zurada, 1995; Bose ve Liang, 1996)



1. eğitim grubuna ait örnekler



2. eğitim grubuna ait örnekler



3. eğitim grubuna ait örnekler



4. eğitim grubuna ait örnekler



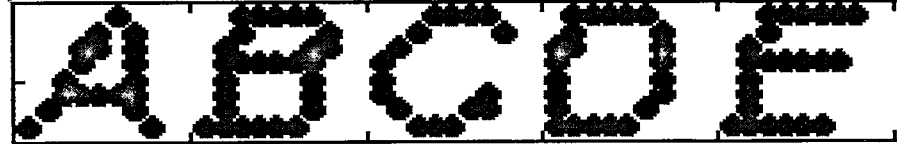
5. eğitim grubuna ait örnekler



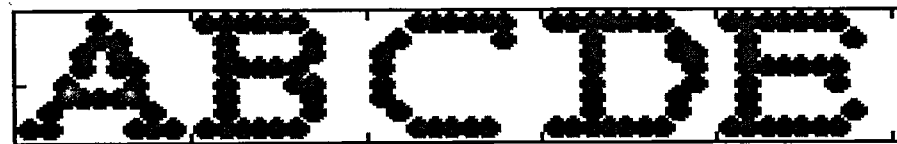
6. eğitim grubuna ait örnekler



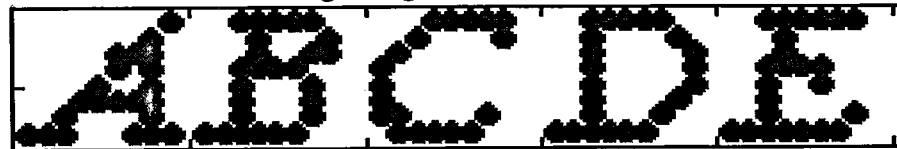
7. eğitim grubuna ait örnekler



8. eğitim grubuna ait örnekler



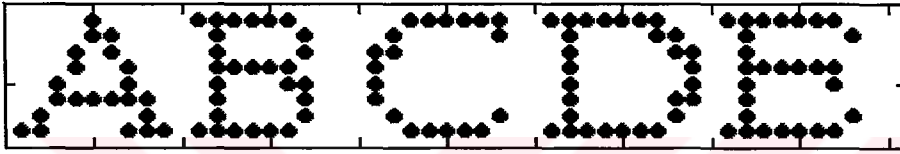
9. eğitim grubuna ait örnekler



10. eğitim grubuna ait örnekler

Şekil 4.30 Karakter tanıma probleminde kullanılan eğitim kümesi

Bu çalışmada, 5 farklı karakterden (A, B, C, D, E) 10 değişik örnek alınarak oluşturulan bir eğitime kümesi kullanılmıştır. Her bir karakter, 10-sütun $\times$ 8-satırlık bir piksel matris şeklinde tanımlanmıştır. Örneklerin tamamı temiz bir veri kümesinden elde edilmiştir. Açık çekilmiş fotokopi, koyu çekilmiş fotokopi ve fakstan alınan yazı karakterleri ile benzer şekilde oluşturulan bir başka kümeden seçilen örneklerle de ağ test edilmiştir. Eğitime kümesine ait örnekler Şekil 4.30'da gösterilmiştir. Test etme sonucunda A, C, D ve E karakterlerinin doğru bir şekilde tanındığı B karakterinin ise E ile karıştırıldığı görülmüştür. Eğitilen örnekler incelenirse E karakterine ait örneklerin büyük ölçüde bozulmuş olup B'ye çok benzediği görülebilir. Ayrıca test etme işlemi sırasında ağa girilen farklı herhangi bir karakterin bu beş karakter kümesinden hiç birine atanmadığı, yani ağ tarafından tanınmadığı görülmüştür. Test edilen karakterlerden her birine ait birer örnek Şekil 4.31' de verilmiştir.



Şekil 4.31 Test edilen karakterlerden bazı örnekler

5. KONİK KESİT FONKSİYONLU YAPAY SİNİR AĞININ DUYARLIĞI

5.1 Yapay Sinir Ağlarında Duyarlık Analizi

Duyarlık analizi, bir ağın giriş ve çıkışları arasındaki sebep ve sonuç ilişkilerini ortaya koyan bir yöntemdir. Bir yapay sinir ağını eğittikten sonra, her bir girişin ağ çıkışı üzerinde yarattığı etki bulunmak istenebilir. Bu durumda duyarlık analizi bir yöntem olarak kullanılabilir. Burada temel fikir, ağın her bir giriş kanalının az da olsa değişimi halinde çıkışlarda da buna uygun bir değişimin gözleneceğidir. Duyarlık analizi sonucu ağ çıkışında etkisi az olan giriş eğitme sırasında göz önüne alınmayabilir. Yani az duyarlığa sahip olan giriş kanalı ağdan çıkartılabilir. Böylece ağ boyutu, dolayısıyla karmaşıklık ve eğitme zamanı azaltılmış olur. Bu da ağ performansının iyileşmesini sağlar. Ayrıca, ağ boyutunun ve karmaşıklığının azalması, özellikle yapay sinir ağlarının elektronik olarak tasarımına ilişkin uygulamalarda önemlidir. (www.nd.com; Hashem, 1992)

5.2 Duyarlık Fonksiyonları

Ölçüm hatalarına veya sıcaklık, nem, vs. gibi çevre koşullarına bağlı olarak sistem parametrelerinde değişimler olabilir. Bu değişimler sistemi temsil eden büyüklüklere etki eder. Meydana gelen değişimlerin sonucu çok etkilememesi için belli sınırlar içinde kalması istenir. Bu nedenle de bu değişimlerin bulunması gereklidir. Bir sistemde y , x parametresine bağlı bir büyüklük olmak üzere, x 'de meydana gelen Δx değişmesi nedeni ile y 'de meydana gelen değişmeyi bulmak için kullanılan ve $S[y,x]$ ile gösterilen duyarlık fonksiyonu dört ayrı biçimde tanımlanır. (Acar, 1979; Baştürk, 1993; Baştürk ve Gürgen, 1993)

1. Normalleştirilmiş duyarlık

$$\frac{\Delta y}{y} = S[y,x] \frac{\Delta x}{x} \Rightarrow S[y,x] = \frac{\partial \ln y}{\partial \ln x} \quad (5.1)$$

2. y 'ye göre yarı-normalleştirilmiş duyarlık

$$\Delta y = S[y,x] \frac{\Delta x}{x} \Rightarrow S[y,x] = \frac{\partial y}{\partial \ln x} \quad (5.2)$$

3. x'e göre yarı-normalleştirilmiş duyarlık

$$\frac{\Delta y}{y} = S[y, x] \Delta x \Rightarrow S[y, x] = \frac{\partial \ln y}{\partial x} \quad (5.3)$$

4. Normalleştirilmemiş duyarlık

$$\Delta y = S[y, x] \Delta x \Rightarrow S[y, x] = \frac{\partial y}{\partial x} \quad (5.4)$$

5.3 Konik Kesit Fonksiyonlu Yapay Sinir Ağı Çıkışlarının Girişlere Göre Duyarlığı

Giriş verisinde meydana gelebilecek bir değişimden Konik Kesit Fonksiyonlu Yapay Sinir Ağı çıkışlarının nasıl etkileneceğini bulmak için ağ çıkışının girişe göre duyarlığı incelenmiştir. Bu amaçla $\partial out_{pk}^{(2)} / \partial x_{pi}$ değerinin hesaplanması gerekmektedir. Yani p . örnek için giriş katmanındaki i . düğüme gelen bilgideki değişimin, çıkış katmanındaki k . düğüm çıkışını nasıl etkilediğini bulmak gerekir. Bu hesap yapılırken çıkıştaki nonlineerlik için Eşitlik (5.5)'de tanımlanan lojistik fonksiyon kullanılmıştır. Ayrıca Konik Kesit Fonksiyonlu Yapay Sinir Ağı çıkışının Çok Katmanlı Algılayıcı karakteristiği gösterdiği gözönüne alınarak, aktivasyon ifadesinde (Eşitlik 5.6) sadece ağırlıklı toplam alınmış, Radyal Temelli Fonksiyonlu ağ karakteristiği gösteren bileşenler hesaba katılmamıştır.

$$out_{pk}^{(2)} = \frac{1}{1 + e^{-y_{pk}^{(2)}}} \quad (5.5)$$

$$y_{pk}^{(2)} = \sum_{j=1}^n w_{kj} out_{pj}^{(1)} \quad (5.6)$$

Burada kullanılan değişkenler aşağıda verildiği gibi tanımlanmaktadır:

- p : örnek sayısı
- x_{pi} : p.örnek için giriş katmanındaki i. birime gelen veri
- $out_{pk}^{(2)}$: p.örnek için çıkış katmanındaki k. birim yanıtı
- $out_{pj}^{(1)}$: p.örnek için gizli katmandaki j. birim çıkışı
- $y_{pk}^{(2)}$: p.örnek için çıkış katmanındaki k. birim için aktivasyon
- $y_{pj}^{(1)}$: p.örnek için gizli katmandaki j. birim için aktivasyon
- w_{ij} : girişin i. düğümünden gizli katmanın j. düğümüne olan bağlantı ağırlığı
- w_{jk} : gizli katmanın j. düğümünden çıkış katmanının k. düğümüne olan bağlantı ağırlığı

Buna göre, i . birim girişinin çıkış katmanı üzerindeki etkisini hesaplayabilmek için zincir kuralı uygulanarak kısmi türevler yazılacak olursa;

$$\frac{\partial out_{pk}^{(2)}}{\partial x_{pi}} = \frac{\partial out_{pk}^{(2)}}{\partial y_{pk}^{(2)}} \cdot \frac{\partial y_{pk}^{(2)}}{\partial x_{pi}} \quad (5.7)$$

ifadesi elde edilir. Burada eşitliğin sağ tarafındaki birinci terim aşağıdaki gibi ifade edilir;

$$\frac{\partial out_{pk}^{(2)}}{\partial y_{pk}^{(2)}} = out_{pk}^{(2)} (1 - out_{pk}^{(2)}) \quad (5.8)$$

Buradan (5.7) eşitliği tekrar yazılacak olursa

$$\frac{\partial out_{pk}^{(2)}}{\partial x_{pi}} = \frac{\partial y_{pk}^{(2)}}{\partial x_{pi}} out_{pk}^{(2)} (1 - out_{pk}^{(2)}) \quad (5.9)$$

olarak bulunur. Eşitlik (5.7)'deki ikinci terim

$$\frac{\partial y_{pk}^{(2)}}{\partial x_{pi}} = \sum_{j=1}^n w_{kj} \frac{\partial out_{pj}^{(1)}}{\partial x_{pi}} \quad (5.10)$$

ile ifade edilir. Burada aktivasyon fonksiyonu olarak Eşitlik (5.11)'de verilen lojistik fonksiyon kullanılmıştır. Gizli katman çıkışının hem Çok Katmanlı Algılayıcı hem de Radyal Temelli Fonksiyonlu ağ karakteristiği göstermesi nedeni ile $y_{pj}^{(1)}$ ile verilen aktivasyon ifadesinde her iki yapıya ait bileşenler göz önüne alınmıştır.

$$out_{pj}^{(1)} = \frac{1}{1 + e^{-y_{pj}^{(1)}}} \quad (5.11)$$

$$y_{pj}^{(1)} = \sum_{i=1}^m (x_{pi} - c_{ij}) w_{ij} - \cos \omega_j \sqrt{\sum_{i=1}^m (x_{pi} - c_{ij})^2} \quad (5.12)$$

Eşitlik (5.7)'de olduğu gibi gizli katmandaki j . bileşen çıkışının p . örnek için i . girişe göre kısmi türevi zincir kuralına göre iki bileşen ile ifade edilirse

$$\frac{\partial out_{pj}^{(1)}}{\partial x_{pi}} = \frac{\partial out_{pj}^{(1)}}{\partial y_{pj}^{(1)}} \cdot \frac{\partial y_{pj}^{(1)}}{\partial x_{pi}} \quad (5.13)$$

ifadesi elde edilir. Bu eşitlikte sağ taraftaki çarpımın ilk terimi

$$\frac{\partial out_{pj}^{(1)}}{\partial y_{pj}^{(1)}} = out_{pj}^{(1)} (1 - out_{pj}^{(1)}) \quad (5.14)$$

olarak bulunur. (5.13) eşitliği tekrar düzenlenirse;

$$\frac{\partial out_{pj}^{(1)}}{\partial x_{pi}} = \frac{\partial y_{pj}^{(1)}}{\partial x_{pi}} out_{pj}^{(1)} (1 - out_{pj}^{(1)}) \quad (5.15)$$

elde edilir. Konik Kesit Fonksiyonlu Yapay Sinir Ağı için aktivasyon ifadesinin girişe göre kısmi türevi ise;

$$\frac{\partial y_{pj}^{(1)}}{\partial x_{pi}} = w_{ij} - \cos \omega_j \frac{(x_{pi} - c_{ij})}{\|x_p - c_j\|} \quad (5.16)$$

eşitliğindeki gibi bulunur. Sonuç olarak, p . örnek için k . ağ çıkışının i . girişe göre değişimi; yani duyarlık ifadesi aşağıdaki ifade ile verilir.

$$\frac{\partial out_{pk}^{(2)}}{\partial x_{pi}} = \left[\sum_{j=1}^n w_{kj} \left(w_{ij} - \cos \omega_j \frac{(x_{pi} - c_{ij})}{\|x_p - c_j\|} \right) out_{pj}^{(1)} (1 - out_{pj}^{(1)}) \right] out_{pk}^{(2)} (1 - out_{pk}^{(2)}) \quad (5.17)$$

5.4 Duyarlık Analizinin Çeşitli Problemlere Uygulanması

Ağ çıkışlarının herhangi bir girişe göre duyarlığının az olması durumunda, o girişin ağdan çıkartılabileceği varsayılmıştır. Elde edilen duyarlık ifadesi, değişik problemlere uygulanarak sonuçlar verilmiş ve bu ifadenin öznetelik azaltmada kullanılabilirliği incelenmiştir.

5.4.1 Kontakt lens uydurma problemi için duyarlık analizi

Kontakt lens uydurma problemi için (5.17)'de verilen eşitlik ile eğitim setindeki tüm girişler için ağ çıkışlarının duyarlığı hesaplanarak bu örnekler için her bir ayırt edici özneteliğe ait en

büyük duyarlık bulunmuştur. Ağ çıkışının en az etkilendiği giriş, en az duyarlığa sahip giriş olduğu için bulunan maksimum duyarlıklar içinde en küçük olan tespit edilip, buna ait öznitelik ağdan çıkartılmıştır. Geriye kalan öznitelikler ile ağ yeniden eğitilerek test edilmiştir. Aşağıda eğitme sonucunda elde edilen duyarlık sonuçları verilmiştir. Burada sütunlar ayırt edici öznitelikleri, satırlar ise çıkışları göstermektedir.

1. örnek için

-0.1240	0.8733	-1.1205	-0.9778
0.2313	-0.2728	0.4834	-0.3527
-0.2952	-0.2722	0.2795	1.0505

2. örnek için

0.0566	0.3152	-0.5124	-0.4988
0.0593	-0.0694	0.1472	-0.1208
-0.2136	-0.1915	0.2303	0.7125

3. örnek için

0.2104	0.4163	-0.6624	-0.7400
0.0205	-0.0394	0.0958	-0.0600
-0.3299	-0.3857	0.4572	1.1444

4. örnek için

-0.1241	0.2996	-0.6024	-0.3348
0.3766	-0.1969	0.6015	-0.4000
-0.1958	-0.1134	0.0655	0.4757

5. örnek için

-0.0055	0.1095	-0.2446	-0.1308
0.2012	-0.1289	0.4699	-0.3664
-0.2392	-0.1483	0.0585	0.6584

6. örnek için

0.0200	0.0699	-0.0946	-0.0227
0.1222	-0.3020	0.5749	-0.6079
-0.2499	0.0275	-0.2281	0.8160

7. örnek için

-0.0398	0.0310	-0.0787	-0.0656
0.2718	-0.1175	0.4189	-0.3640
-0.1022	-0.0226	-0.0525	0.3824

8. örnek için

-0.0400	0.3063	-0.4628	-0.6812
0.0096	-0.0170	0.0401	-0.0241
-0.0537	-0.1280	0.1463	0.4351

9. örnek için

-0.0214	0.0255	-0.0334	-0.0233
0.4162	-0.2836	0.6098	-0.5830
-0.1063	0.0021	-0.1077	0.3467

10. örnek için

0.0274	0.0975	-0.1495	-0.2354
0.0078	-0.0119	0.0280	-0.0200
-0.0341	-0.0409	0.0465	0.1516

11. örnek için

0.0009	0.0498	-0.0352	-0.0750
0.0183	-0.0139	0.0319	-0.0410
-0.0180	-0.0202	0.0036	0.0726

12. örnek için

0.0550	0.6750	-0.7970	-0.7370
0.1482	-0.1973	0.3170	-0.2709
-0.4371	-0.3432	0.3712	1.3486

13. örnek için

0.0239	0.0193	-0.0584	-0.0707
0.0346	-0.0200	0.1094	-0.1193
-0.0478	-0.0111	-0.0160	0.1398

14. örnek için

0.0922	0.1511	-0.3690	-0.1552
0.0859	-0.1839	0.9035	-0.7814
-0.2546	-0.2034	-0.0623	0.9850

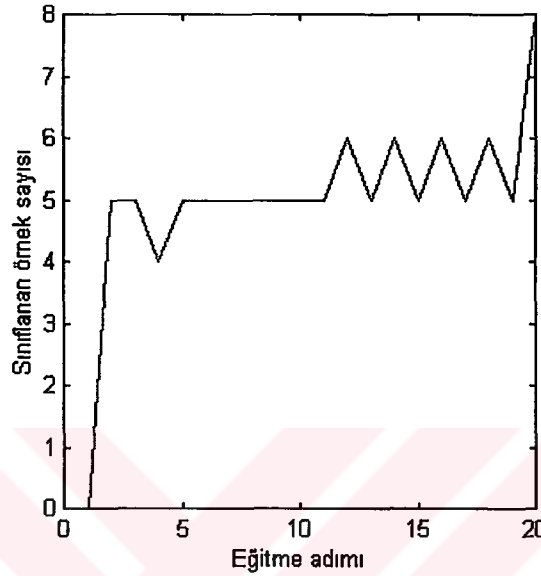
15. örnek için

0.0275	0.0459	-0.0277	-0.0852
0.0100	-0.0059	0.0177	-0.0282
-0.0196	-0.0145	0.0010	0.0551

16. örnek için

0.2245	0.5299	-0.6091	-0.6860
0.0671	-0.0988	0.1777	-0.1814
-0.3667	-0.2858	0.2857	1.0794

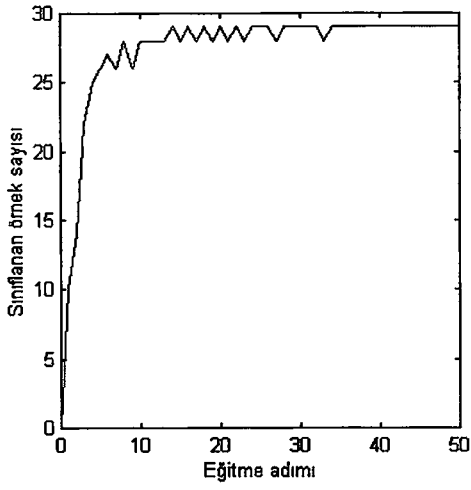
Buna göre 1. özniteliğe; yani kişinin yaşına ait duyarlılığın en az olduğu görülmüş ve bu giriş ihmal edilerek, geriye kalan üç öznitelik ile örnekler tekrar eğitilmiş ve test edilmiştir. Test etme işlemi sonucunda, Şekil 5.1'de görüldüğü gibi daha önceki benzer şekilde tüm örneklerin tanındığı, ayrıca eğitime adımının dört öznitelik kullanılarak yapılan eğitime işleminden daha az olduğu görülmüştür. Bu sonuç da, bu problem için özellik çıkarmada duyarlık analizinin kullanılabileceğini göstermiştir.



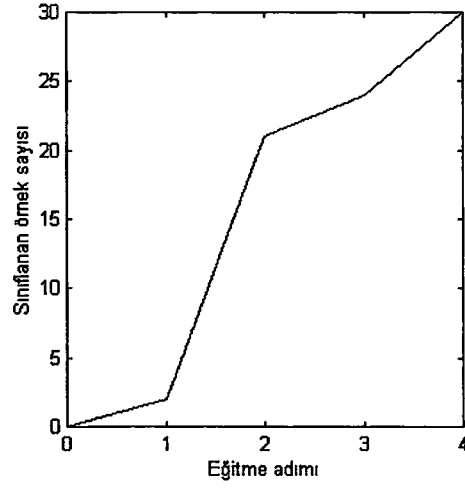
Şekil 5.1 Kontakt lens uydurma problemi için duyarlık analizi sonucunda yeniden eğitilen ağın sınıfladığı örnek sayısı

5.4.2 İris bitkisi sınıflandırma problemi için duyarlık analizi

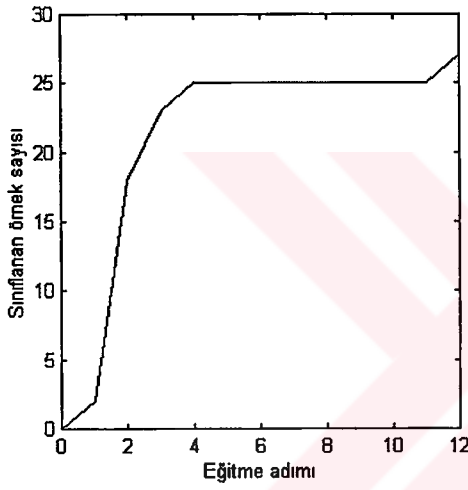
İris bitkisi sınıflandırma problemi için yine aynı yöntemle duyarlık analizi yapılmıştır. Bölüm 4.3'de kullanılan 5-katlamalı çapraz geçerlilik yönteminde en iyi sonucu veren (yani, en kısa sürede %100 doğrulukla sınıflanan) A5 grubu için en az duyarlılığa sahip olan öznitelik belirlenmiştir. Analiz sonucunda, 4. özelliğe yani taç yaprak genişliğine ait duyarlılığın en küçük olması nedeniyle bu öznitelik ağdan çıkartılarak ağ, her bir grup (A1, A2, A3, A4, A5) için yeniden eğitilmiş ve test edilmiştir. Duyarlık analizi sonrasında, üç öznitelik kullanılarak 5-katlamalı çapraz geçerlilik yöntemi ile yeniden elde edilen sonuçlar Şekil 5.2 ve Çizelge 5.1'de verilmiştir. Burada ortalama sınıflama oranı %96'dır. Bu sonuç, dört öznitelik kullanılarak elde edilen orandan (%98.668) daha düşük olmasına rağmen, karşılaştırılan diğer algoritmalara (k-NN -%95.34, k-NNFP-%96.02) yakın sonuç vermektedir. Karşılaştırılan bu algoritmalarda da dört öznitelik kullanıldığına dikkat edilmelidir.



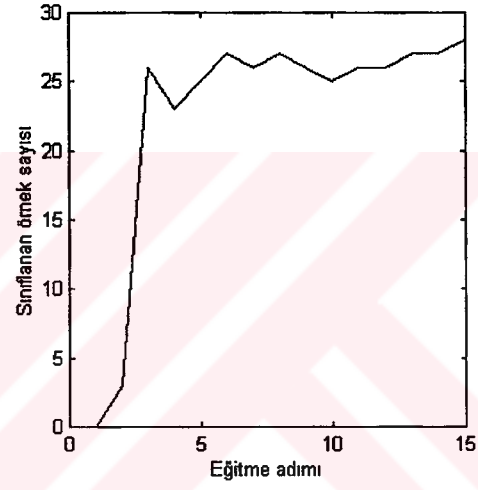
A1



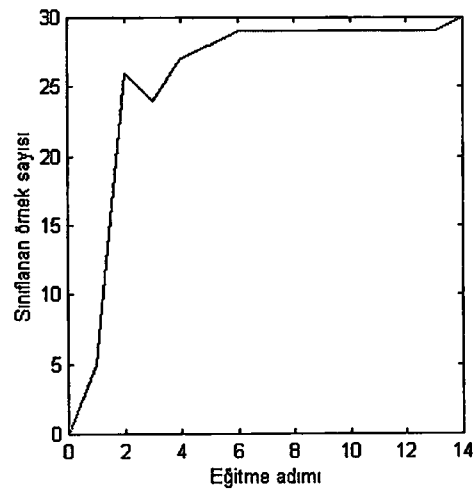
A2



A3



A4



A5

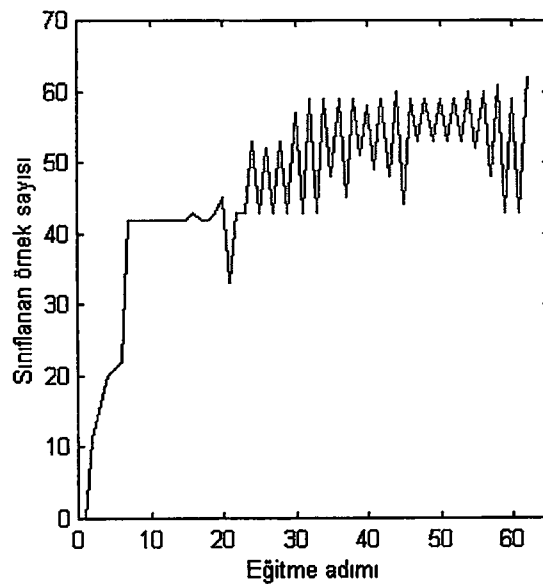
Şekil 5.2 İris bitkisi sınıflandırma problemi için duyarlık analizi sonucunda yeniden eğitilen ağın her bir grup için sınıfladığı örnek sayısı

Çizelge 5.1 5-katlamalı çapraz geçerlilik yöntemi ile elde edilen sonuçlar

İris Veritabanı	Duyarlık Analizinden Önce		Duyarlık Analizinden Sonra	
	Eğitme adımı	Sınıflama oranı	Eğitme adımı	Sınıflama oranı
A1 Grubu	9	%100	12	%96.67
A2 Grubu	10	%100	4	%100
A3 Grubu	25	%96.67	12	%90
A4 Grubu	6	%96.67	15	%93.33
A5 Grubu	5	%100	15	%100

5.4.3 Fetus gelişimi problemi için duyarlık analizi

Kontakt lens uydurma probleminde açıklandığı gibi, fetus gelişimi problemi için de aynı yöntem izlenmiş ve eğitim setindeki örnekler için (5.17) eşitliği kullanılarak duyarlık hesabı yapılmıştır. Elde edilen sonuçlar incelendiğinde, ağ çıkışını en az etkileyen yani en az duyarlılığa sahip olan girişin 4. özneteliği gösteren giriş olduğu görülmüştür. Buna göre, bu özneteliğe, yani S/D (Systolic/Diastolic) oranına ait giriş ihmal edilerek, geriye kalan üç öznetelik ile örnekler tekrar eğitilmiş ve test edilmiştir. Test etme işlemi sonucunda elde edilen sonuç Şekil 5.3'de verilmiştir.

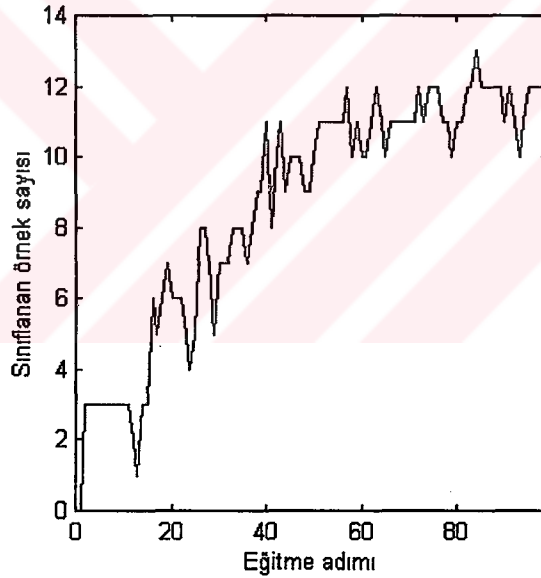


Şekil 5.3 Fetus gelişimi problemi için duyarlık analizi sonucunda yeniden eğitilen ağın sınıfladığı örnek sayısı

Sonuçlar, S/D oranı ihmal edildiği durumda test edilen 63 örnekten 62'sinin tanındığını (%98.41 sınıflama doğruluğu) göstermiştir. Bu sonuç, dört öznitelik kullanılarak elde edilen sonuç (%100 sınıflama doğruluğu) kadar iyi olmamasına rağmen, daha önce yapılan bir çalışmada (Güler, 1999) elde edilen sınıflama oranından (%95.23) daha iyidir.

5.4.4 EMG işaretlerini sınıflandırma problemi için duyarlık analizi

EMG işaretlerinin sınıflanması probleminde, duyarlığın belirlenmesi için altı farklı hareketi gösteren her bir sınıfa ait 12 örnekten ilk 9 tanesi eğitim, son 3 tanesi test için kullanılarak ağ eğitilmiş ve girişlerin duyarlıkları hesaplanmıştır. Elde edilen duyarlık sonuçlarına göre, ağ çıkışında en az etkisi olan özniteliğin 3. AR parametresi olduğu görülmüştür. Daha önce dört AR parametresinin kullanımı durumunda 14 örnek tanınırken, bu parametreye ait girişin ağdan çıkartılması durumunda örneklerin 13 tanesi tanınmıştır. Elde edilen sonuç Şekil 5.4'de verilmiştir.



Şekil 5.4 EMG işaretlerinin sınıflandırılması problemi için duyarlık analizi sonucunda yeniden eğitilen ağın sınıfladığı örnek sayısı

5.4.5 Sonuçlar

Bu bölümde Konik Kesit Fonksiyonlu Yapay Sinir Ağı için duyarlık formülleri elde edilmiş ve bu formüller kullanılarak dört farklı problem için duyarlık analizi yapılmıştır. Analiz sonucunda her bir problem için eğitim kümesinden bir öznitelik çıkartılarak ağ yeniden eğitilmiş ve test edilmiştir. Bir özniteliğin çıkartılmasına rağmen elde edilen sonuçlar, bütün problemler için kabul edilebilir bir sınıflama doğruluğuna ulaşıldığını göstermiştir. Bu da

yapılan duyarlık analizi işleminin yapay sinir ağlarında öznelik azaltmada kullanılabileceğini göstermektedir. Farklı problemler için duyarlık analizi öncesi ve sonrası elde edilen toplu sonuçlar Çizelge 5.2'de verilmiştir.

Çizelge 5.2 Tüm veritabanları için elde edilen sonuçlar

Veritabanı Adı	Duyarlık Analizinden Önce		Duyarlık Analizinden Sonra	
	Eğitme adımı	Sınıflama oranı	Eğitme adımı	Sınıflama oranı
Lens	24	%100	18	%100
İris	11	%98.668	12	%96
Fetus	153	%100	62	%98.41
EMG	98	%77.77	84	%72.22

6. SONUÇLAR ve TARTIŞMA

Oldukça yeni bir ağ yapısı olan Konik Kesit Fonksiyonlu Yapay Sinir Ağı için adaptif bir öğrenme algoritması geliştirilerek bu ağın performansı ve duyarlılığı çeşitli problemler üzerinde incelenmiştir. Bu çalışma üç ana konu üzerinde yoğunlaşmaktadır: 1) Öğrenme algoritmasının geliştirilmesi, 2) ağın limitlerinin ve farklı amaçlar için kullanımının belirlenmesi, 3) ağ çıkışının girişlere olan duyarlılığının incelenmesi.

6.1 Öğrenme Algoritmasının Geliştirilmesi

CSFNN'nin eğitilmesinde yakınsamayı hızlandırmak için standart geriye yayılma algoritmasındaki öğrenme oranı adaptif yapılmış ve momentum terimi eklenmiştir. Burada algoritmanın performansına göre öğrenme oranının değiştirildiği Değişken Öğrenme Oranlı geriye yayılma algoritmasının kuralları kullanılmıştır. İncelenen problemler için adaptif öğrenmenin standart CSFNN'ye göre üstün olduğu görülmüştür.

6.2 Ağın Performansının Değerlendirilmesi

Ağın performansının belirlenmesinde genelleştirme yeteneği, çok amaçlı (farklı problemler için) kullanılabilirliği, ağ yapısının karmaşıklığı, elektronik tasarıma uygunluğu incelenmiştir. Burada elektromiyografik işaretlerin sınıflandırılması, kontakt lens uydurma, iris bitkisi sınıflandırma, fetus gelişimi, fonksiyon yaklaşımı, mikrodalga transistörün modellenmesi ve karakter tanıma problemleri ele alınmıştır. Her bir problem için adaptif CSFNN için elde edilen sonuçlar daha önce aynı problemler için başka algoritmalar kullanılarak bulunmuş sonuçlarla karşılaştırılmıştır.

EMG işaretlerinin sınıflanması problemi üç ayrı yöntemle incelenmiştir. 1) Tüm veri bilgisi ağa girilerek eğitilmiş ve test edilmiştir, elde edilen sonuç geriye yayılmalı MLP ile bulunanlarla karşılaştırılmıştır. MLP'de yaklaşık 3000 iterasyonda ancak %96 sınıflama doğruluğuna ulaşılırken adaptif öğrenmeli CSFNN ile farklı momentum katsayılarına ve kullanılan merkez sayılarına bağlı olarak 890 ila 1500 iterasyon sonucunda %100 doğruluk ile yakınsama sağlanmaktadır. 2) Birini dışarda bırak (Leave-one-out) yöntemi kullanılarak elde edilen sonuçlar, aynı yöntemi kullanan Bulanık K-En Yakın Komşuluk Sınıflandırıcılarının sonuçları ile karşılaştırılmıştır. Her bir örneğe ait özneliliklerin tamamının ve sadece ikisinin kullanılması ile yapılan eğitime sonrasında adaptif CSFNN %98.6 - %100 doğruluğa farklı

merkez sayıları ve momentum katsayılarına bağlı olarak 7 ila 190 iterasyondan sonra ulaşmaktadır. Bulanık K-En Yakın Komşuluk Sınıflandırıcısında 5000 iterasyondan sonra sınıflama oranı, K komşuluk sayısının 2 ila 6 arasında değişmesine bağlı olarak dört öznitelik kullanıldığında %64-%68, iki öznitelik kullanıldığında ise %53-%71 arasındadır. 3) Üçüncü durumda 4-katlamalı çapraz-geçerlilik yöntemi uygulanmıştır. Test verisi yakınsama kriterine bağlı olarak adaptif CSFNN ile dört farklı grup için %50 ila %78 oranında sınıflanmakta, ayrıca iterasyon sayısının bazı durumlarda biraz arttığı görülmektedir. Bu problem için literatürde başka ağlar için böyle bir genelleştirme yapılmamıştır.

Kontakt lens uydurma problemi için CSFNN yapısında adaptif algoritma ve momentum katsayısının kullanıldığı durumda uygun bir merkez değerinin belirlenmesi ile tüm test verisinin sadece 18 eğitime adımında ağ tarafından tanındığı görülmüştür. Benzer sonuca standart CSFNN ile 76, geriye yayımlı MLP ile ortalama 2187 ve hızlı geriye yayımlı MLP ile ise ortalama 120 eğitime adımında ulaşılmaktadır. Bu problem için farklı momentum katsayıları ve merkez sayılarının öğrenmeyi nasıl etkilediği de ayrıca incelenmiştir. Örneklerin tanınması için 5 merkez değerinin yeterli olduğu hatta en iyi sonucu verdiği görülmüştür. Bu da elektronik tasarım açısından daha küçük boyutlu bir ağa izin vermektedir.

Adaptif öğrenmeli CSFNN'nin genelleştirme yeteneğini daha iyi inceleyebilmek amacı ile İris bitkisi sınıflandırma problemi için 2-katlamalı ve 5-katlamalı çapraz-geçerlilik yöntemleri uygulanmıştır. 2-katlamalı çapraz-geçerlilik yöntemi standart CSFNN ile karşılaştırılmıştır. Burada, eğitime grubuna bağlı olarak sınıflama doğruluğunun adaptif CSFNN ile aynı olduğu (%97.33 - %98.66) fakat eğitime adımının arttığı görülmüştür. Ayrıca aynı yöntem geriye yayımlı MLP ve hızlı geriye yayımlı MLP için de uygulanmıştır. Geriye yayımlı MLP ile 140-400 iterasyonda ortalama %94.66, hızlı geriye yayımlı MLP ile 58-105 iterasyonda ortalama %94.66 - %96 sınıflama doğruluğu elde edilmektedir. 5-katlamalı çapraz-geçerlilik yönteminde tüm alt kümeler için elde edilen sonuçların ortalaması alındığında sınıflama doğruluğunun %98.668 olduğu görülmüştür. Aynı yöntem için elde edilen en iyi sınıflama doğruluğu, k-NNFP algoritması ile %96.02'dir. 2-katlamalı ve 5-katlamalı çapraz-geçerlilik yöntemlerinden elde edilen sonuçlar adaptif öğrenmeli CSFNN'nin diğer yapıların tamamından daha iyi bir genelleştirme yeteneğine sahip olduğunu göstermiştir. Adaptif CSFNN için 5-katlamalı çapraz-geçerlilik yönteminde A5 grubu en iyi sonucu (5 eğitime adımında %100 doğrulukla) vermektedir. Buna karşılık standart CSFNN ile en iyi durumda 503 adımda sonuca ulaşılırken geriye yayımlı MLP ile rasgele başlatılan ağırlıklara göre 4000 ila 30000; hızlı geriye yayımlı MLP ile 800-20000 eğitime adımından sonra örnekler

doğru olarak sınıflamaktadır. Ayrıca bazı durumlarda hızlı geriye yayılmalı MLP'nin sonuca yakınsamadığı da not edilmiştir. Farklı momentum katsayılarının ve merkez sayılarının adaptif CSFNN için öğrenmeyi nasıl etkilediği de incelenmiştir. Buna göre standart CSFNN ve adaptif öğrenmeli CSFNN, sınıfları sadece 6 merkez kullanarak ayırabilirken standart RBF ağı ancak 32 merkez yerleştirdikten sonra aynı grup için örnekleri sınıflamaktadır. Bu sonuç da yine, CSFNN yapısının elektronik tasarımlar açısından daha az kompleks bir devre ile gerçekleştirilebileceğini göstermektedir.

Adaptif CSFNN'nin tıbbi teşhis amacı ile kullanılıp kullanılamayacağını incelemek için fetus gelişimi problemi ele alınmıştır. Sonuçlar, daha önceki bir çalışmada MLP kullanılarak elde edilen sonuçlarla karşılaştırılmıştır. MLP ile farklı gizli katman düğüm sayıları için en fazla %95.23 sınıflama doğruluğu elde edilirken adaptif öğrenmeli CSFNN ile test edilen örnekler %100 doğrulukla sınıflanmaktadır. Buna göre, adaptif CSFNN algoritmasının tıbbi teşhis amacıyla da kullanılabileceği görülmüştür.

Adaptif CSFNN'nin fonksiyon yaklaşımcı olarak nasıl davrandığı da ayrı bir problem olarak incelenmiştir. Burada üç değişik fonksiyon ele alınarak gizli düğüm sayısının farklı değerleri için istenen fonksiyonlara doğru olarak yaklaşıldığı görülmüştür. Çalışmada ayrıca Matlab Neural Networks Toolbox'ı içerisindeki fonksiyon yaklaşımcı algoritmalar (Geriye yayılma, Hızlı geriye yayılma ve Levenberg Marquardt öğrenme algoritmaları ile Çok Katmanlı Algılayıcı yapısı ve OLS algoritması ile eğitilen Radyal Temelli Fonksiyonlu ağ yapısı) incelenmiş ve adaptif öğrenmeli CSFNN ile karşılaştırılarak eğitime adımı, eğitime süresi ve eğitime sırasında yapılan işlem sayısı cinsinden sonuçlar verilmiştir. Burada, Çok Katmanlı Algılayıcı yapısındaki ağların başlangıç ağırlıklarının rasgele seçilmesinden dolayı kararsız olduğu görülmüştür. Böyle bir durum adaptif CSFNN için geçerli değildir. Başka bir deyişle, adaptif CSFNN istenen belli bir fonksiyona her seferinde aynı eğitime adımı, eğitime süresi ve işlem sayısından sonra yakınsamaktadır.

Fonksiyon yaklaşımının özel bir hali olarak bir mikrodalga transistörünün YSA ile modellenmesi ele alınmıştır. NE21900 mikrodalga transistörü için veriler adaptif öğrenmeli CSFNN kullanılarak eğitilmiş ve test edilmiştir. Sonuçlar böyle bir modellemenin mümkün olduğunu göstermiştir.

Son olarak, 5 farklı karakterden (A, B, C, D, E) 10 değişik örnek alınarak oluşturulan bir eğitime kümesi kullanılarak karakter tanıma problemi incelenmiştir. Test etme sonucunda A,

C, D ve E karakterlerinin doğru bir şekilde tanındığı B karakterinin ise E ile karıştırıldığı görülmüştür. Eğitilen örnekler incelenirse E karakterine ait örneklerin büyük ölçüde bozulmuş olup B'ye çok benzediği görülebilir. Ayrıca test etme işlemi sırasında ağa girilen farklı herhangi bir karakterin bu beş karakter kümesinden hiç birine atanmadığı, yani ağ tarafından tanınmadığı görülmüştür. Burada örnek sayısının yetersiz olması nedeni ile B karakterinin tanınmadığı ve ileride yapılacak çalışmalarla daha geniş bir karakterden oluşan ve daha çok sayıda örnek içeren bir kümenin incelenmesi düşünülmektedir.

Sonuç olarak, geliştirilen algoritmanın değişik problemler üzerindeki performansının diğer ağ yapılarından pek çok durumda daha üstün olduğu belirlenmiştir. Burada adaptif CSFNN'nin oldukça iyi bir genelleştirme yeteneğine sahip olduğu, farklı problemler için kullanılabileceği, gizli birim sayısının az olmasından dolayı ağ yapısının fazla karmaşık olmadığı ve bu nedenle de elektronik tasarıma daha uygun olduğu görülmüştür.

6.3 Duyarlık Analizinin Değerlendirilmesi

Literatürde ilk kez Konik Kesit Fonksiyonlu Yapay Sinir Ağı için duyarlık analizi yapılmış; başka bir deyişle, ağ çıkışlarının giriş verisinde meydana gelebilecek bir değişimden nasıl etkilendiği incelenmiştir. Duyarlık formülleri çıkartıldıktan sonra elde edilen ifade, değişik problemlere uygulanarak sonuçlar verilmiş ve bu ifadenin girişleri tanımlayan özniteliklerin azaltılmasında kullanılıp kullanılamayacağı değerlendirilmiştir.

Kontakt lens uydurma problemi için yapılan duyarlık analizi sonucunda kişinin yaşına ait duyarlılığın en az olduğu görülmüş ve bu giriş ihmal edilerek, geriye kalan üç öznitelik ile örnekler tekrar eğitilmiş ve test edilmiştir. Test etme işlemi sonucunda tüm örneklerin tanındığı, ayrıca eğitime adımının dört öznitelik kullanılarak yapılan eğitime işleminden daha az olduğu görülmüştür.

İris bitkisi sınıflandırma problemi için 5-katlamalı çapraz geçerlilik yönteminde en iyi sonucu veren (yani, en kısa sürede %100 doğrulukla sınıflanan) A5 grubu kullanılarak duyarlık analizi yapılmıştır. Analiz sonucunda, taç yaprak genişliğine ait duyarlılığın en küçük olması nedeniyle bu öznitelik ağdan çıkartılarak ağ, her bir grup (A1, A2, A3, A4, A5) için yeniden eğitilmiş ve test edilmiştir. Burada ortalama sınıflama oranı %96'dır. Bu sonuç, dört öznitelik kullanılarak elde edilen orandan (%98.668) daha düşük olmasına rağmen, karşılaştırılan diğer

algoritmalar (k-NN -%95.34, k-NNFP-%96.02) yakın sonuç vermektedir. Karşılaştırılan bu algoritmalarda da dört öznitelik kullanıldığına dikkat edilmelidir.

Fetus gelişimi problemi için de aynı yol izlenmiş ve eğitim setindeki örneklerle ilişkin duyarlık hesabı yapılmıştır. Elde edilen sonuçlar incelendiğinde, ağ çıkışının en az etkilendiği yani en az duyarlığa sahip olan girişin S/D (Systolic/Diastolic) oranına ait giriş olduğu görülmüştür. Bu öznitelik ağdan çıkartılarak yeniden elde edilen sonuca göre test edilen 63 örnekten 62'sinin tanınmaktadır. Bu sonuç, 4 özellik kullanılarak elde edilen (63'te 63) kadar iyi olmamasına rağmen, MLP ile elde edilen sonuçtan (63'te 60) daha iyidir.

EMG işaretlerinin sınıflanması probleminde, 4-katlamalı çapraz-geçerlilik yönteminde en iyi sınıflama yapan alt küme kullanılarak girişlerin duyarlıkları hesaplanmıştır. Daha önce dört AR parametresinin kullanımı durumunda 14 örnek tanınırken, duyarlık analizi sonuçlarına göre, 3. AR parametresine ait girişin ağdan çıkartılması durumunda örneklerin 13 tanesi tanınmıştır.

Elde edilen sonuçlar, burada uygulanan tüm problemler için öznitelik azaltmada duyarlık analizinin kullanılabileceğini göstermiştir. Duyarlık analizi sonucu ağ çıkışında etkisi az olan girişin, eğitime sırasında göz önüne alınmaması ile ağ boyutu, dolayısıyla karmaşıklık azaltılmaktadır. Bu da ağ performansının iyileşmesini ayrıca ağı elektronik tasarım açısından daha uygun hale gelmesini sağlamaktadır.

6.4 Gelecek Çalışma

Başka bir çalışmada elektronik olarak tasarlanmış olan CSFNN'nin geliştirilmesi bundan sonraki çalışmalar için seçilecek bir hedef olabilir. Burada elde edilen sonuçlar, gerçek dünyaya ait problemlerin çözümünü sağlayacak şekilde tasarlanmış yeni ağı test edilmesi için kullanılabilir. Bir başka hedef olarak, burada kullanılmayan başka problemler üzerinde de ağı performans ve duyarlılığı incelenebilir. Ayrıca bu yapının başka yapılarla (Bulanık mantık yaklaşımları, genetik algoritmalar) bir arada kullanılması da ayrı bir problem olarak ele alınabilir.

YAYIN LİSTESİ

Özyılmaz, L., Yıldırım, T., (1999), “Konik Kesit Fonksiyonlu Yapay Sinir Ağı Kullanılarak Miyoelektrik İşaretlerin Sınıflandırılması”, SİU'99 1999 IEEE Sinyal İşleme ve Uygulamaları Kurultayı, 16-19 Haziran 1999, Ankara.

Özyılmaz, L., Yıldırım, T., Şeker, H., (1999), “EMG Signal Classification Using Conic Section Function Neural Networks”, International Joint Conference on Neural Networks, 10-16 Temmuz 1999, Washington.

Yıldırım, T., Torpi, H., Özyılmaz, L., (1999), “Modelling of Active Microwave Transistors Using Artificial Neural Networks”, International Joint Conference on Neural Networks, 10-16 Temmuz 1999, Washington.

Özyılmaz, L., Yıldırım, T., (2000), “Sensitivity Analysis for Conic Section Function Neural Networks”, International Joint Conference on Neural Networks'e kabul edildi. 24-27 Temmuz, Como, İtalya.

Özyılmaz, L., Yıldırım, T., (2000), “Konik Kesit Fonksiyonlu Yapay Sinir Ağı İçin Duyarlık Analizi”, SİU'2000 IEEE Sinyal İşleme ve Uygulamaları Kurultayı'na kabul edildi. 12-14 Haziran 2000, Belek-Antalya.

Özyılmaz, L., Yıldırım, T., (2000), “Fonksiyon Yaklaşımı İçin Konik Kesit Fonksiyonlu Yapay Sinir Ağının Kullanılması”, TAINN'2000 Turkish Symposium on Artificial Intelligence and Neural Networks'e kabul edildi. 21-23 Haziran 2000, İzmir.

KAYNAKLAR

- Acar, C., (1979), Duyarlık ve Tolerans Analizi, Fatih Yayınevi, İstanbul.
- Ayan, N.F., (1999), "Using Information Gain as Feature Weight", TAINN'99 8th Turkish Symposium on Artificial Intelligence and Neural Networks, 23-25 Haziran 1999, İstanbul.
- Baştürk, L., (1993), Çok Katmanlı Algılayıcının Optimizasyonu ve Ağırlıklara Bağlı Duyarlığı, Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi, İstanbul.
- Baştürk, L., ve Gürgen, F., (1993), "Çok Katmanlı Algılayıcının Optimizasyonu ve Duyarlığı Üzerine Bir Çalışma", Elektrik Mühendisliği 5. Ulusal Kongresi, 1993, Trabzon.
- Bose, N.K., ve Liang, P., (1996), Neural Networks Fundamentals with Graphs, Algorithms, and Applications, McGraw-Hill Inc., Singapore.
- Broomhead, D.S, ve Lowe, D., (1988) "Multivariable Functional Interpolating And Adaptive Networks", Complex Systems, 321-355.
- Chen, S., Cowan, C.F.N., ve Grant, P.M., (1991), "Orthogonal Least Squares learning algorithm for Radial Basis Function Networks", IEEE Transactions on Neural Networks, 2 (2): 302-309.
- Chen, S., Grant, P.M., McLaughlin, S., ve Mulgrew, B., (1993), "Complex-Valued Radial Basis Function Networks", IEE 3rd International Conference on Artificial Neural Networks, 1993.
- Cichocki, A., ve Unbehauen, R., (1993), Neural Networks for Optimization and Signal Processing, John-Wiley and Sons, USA.
- Dorffner, G., ve Porenta, G., (1994), "On Using Feedforward Neural Networks for Clinical Diagnostic Tasks", Artificial Intelligence in Medicine, 6(5), Special Issue on Neurocomputing in Medicine, 417-435.
- Dorffner, G., (1994), "A Unified Framework for MLPs and RBFNs: Introducing Conic Section Function Networks", Cybernetics and Systems, 25: 511-554.
- Geva, S., ve Sitte, J., (1992), "A Constructive Method for Multivariate Function Approximation by Multilayer Perceptrons", IEEE Transactions on Neural Networks, 3 (4), 621-624.
- Graupe, D., (1975), "Functional Separation of EMG Signals Via ARMA Identification Methods for Prosthesis Control Purposes," IEEE Trans. On Syst. Man Cyber., 5: 252-259.
- Güler, N., Gürgen, F., ve Varol, F., (1999), "Desicion of Normal and Growth Retarded Pregnancies by Neural Networks Umblical Arterial Blood Flows Velocity Waveforms," TAINN'99 8th Turkish Symposium on Artificial Intelligence and Neural Networks, 23-25 Haziran 1999, İstanbul.
- Gurney, K., (1996), An Introduction to Neural Networks, UCL Press, London, UK.
- Hagan, M.T., Demuth, H.B., ve Beale, M., (1996), Neural Network Design, PWS Publishing Company, Boston.
- Harvey, R.L., (1994), Neural Network Principles, Prentice-Hall, Inc., USA.

Hashem, S., (1992), "Sensitivity Analysis for Feedforward Artificial Neural Networks with Differentiable Activation Functions," International Joint Conference on Neural Networks, 1992, Baltimore, New Jersey.

Haykin, S., (1994), Neural Networks: A Comprehensive Foundation, Macmillan College Publishing, New York.

Hecht-Nielsen, R., (1991), Neurocomputing, Addison-Wesley Publishing Company, USA.

Hertz, J., Krogh, A., ve Palmer, R.G., (1991), Introduction to The Theory of Neural Computation, Computation and Neural Systems Series, Addison-Wesley.

Hinton, G.E., (1989), "Connectionist Learning Procedures", Artificial Intelligence 40:185-234.

Hirahara, M., ve Oka, N., (1993), "A Hybrid Model Composed of A Multilayer Perceptron and A Radial Basis Function Network", International Joint Conference on Neural Networks, IJCNN'93, Nagoya, Japonya.

Hush, D.R., Horne, B., ve Salas, J.M., (1992), "Error Surfaces for Multilayer Perceptrons", IEEE Transactions on Systems, Man, and Cybernetics, 22 (5): 1152-1161.

Hush, D.R. ve Horne, B.G., (1993), "Progress in Supervised Neural Networks", IEEE Signal Processing Magazine. 8-39.

Karlık, B., (1994), Çok Fonksiyonlu Protezler İçin Yapay Sinir Ağları Kullanılarak Miyoelektrik Kontrol., Doktora Tezi, Yıldız Teknik Üniversitesi, İstanbul.

Kelly, M.F., Parker, P.A., ve Scott, R.N., (1990), "Application of Neural Networks to Myoelectric Signal Analysis: A Preliminary Study", IEEE Transactions on Biomedical Engineering, 37 (3): 221-230.

Kröse, B.J.A., ve van der Smagt, P.P., (1993) An Introduction to Neural Networks, University of Amsterdam.

Lippmann, R.P., (1987), "An Introduction to Computing with Neural Nets", IEEE ASSP Magazine, 4-22.

Maruyama, M., Girosi, F., ve Poggio, T., (1992), A Connection Between GRBF and MLP, A.I. Memo No.1291, Massachusetts Institute of Technology, Boston.

Middleton, N., (1997), "Distributed Representations in Radial Basis Function Networks", in Perspectives in Neural Computing: Connectionist Representations, ed. J.A. Bullinaria, D.W. Glasspool, and G. Houghton, 16-25, Springer-Verlag, London, UK.

Moody, J. ve Darken, C., (1988), "Learning with Localized Receptive Fields", Connectionist Models Summer School, 1988.

Moody, J., ve Darken, C.J., (1989), "Fast Learning in Network of Locally-Tuned Processing Units", Neural Computation 1:281-294.

Moreira, M. ve Fiesler, E., (1995), "Neural Networks with Adaptive Learning Rate and Momentum Terms", IDIAP Technical Report, No.95-04, İsviçre.

- NeuroSolutions, (1999), Sensitivity Analysis for NeuroSolutions, [http:// www.nd.com/products/](http://www.nd.com/products/)
- Ogilvy, C.S., (1969), Excursions in Geometry, Oxford University Press, New York
- Orr, M.J.L., (1996), Introduction to Radial Basis Function Networks, University of Edinburgh, Scotland.
- Poggio, T., ve Girosi, F., (1989), A Theory of Networks for Approximation And Learning. A.I.Memo No.1140, Massachusetts Institute and Technology, Boston.
- Poggio, T., ve Girosi, F., (1990), "Networks for Approximation and Learning", Proceedings of the IEEE, 78 (9): 1481-1497.
- Rumelhart, D.E., Hinton, G.E., ve Williams, R.J., (1987), "Learning Internal Representations by Error Propagation", in Parallel Distributed Processing, ed. D.E. Rumelhart and J.L. McClelland, 1: 318-362, Cambridge, MA: MIT Press.
- Ruth, J., (1994), Low-Power Stochastic Arithmetic Feed-Forward Neural Network, Main subject thesis, Dept. of Informatics, Univ. of Oslo.
- Sarle, W.S., (1999), <ftp://ftp.sas.com/pup/neural/FAQ3.html>.
- Schalkoff, R.J., (1997), Artificial Neural Networks, McGraw-Hill Inc., Singapore.
- Smyth, S.G., (1992), "Designing Multilayer Perceptrons From Nearest Neighbor Systems", IEEE Transactions on Neural Networks, 3(2): 329-333
- Şeker, H., (1995), "Classification of EMG Signals Using Fuzzy Classifiers," Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi, İstanbul.
- Tarassenko, L., ve Roberts, S., (1994), "Supervised and Unsupervised Learning in Radial Basis Function Classifiers", IEEE Proceedings-Vision Image Signal Processing, 141 (4): 210-216.
- Torpi, H., (1998), Mikrodalga Transistörlerinin Yapay Sinir Ağı Eşdeğerlikleri, Doktora Tezi, Yıldız Teknik Üniversitesi, İstanbul.
- Treleaven, P., Pacheo, M., ve Vellasco, M., (1989) "VLSI Architectures for Neural Networks", IEEE Micro, 8-27.
- Tsoi, A.C., (1989), "Multilayer Perceptron Trained Using Radial Basis Functions", Electronics Letters, 25 (19) :1296-1297.
- Warwick, K., Irwin, G.W., ve Hunt, K.J., (1992), Neural Networks for Control and Systems. Peter Peregrinus Ltd., London, UK.
- Weymaere, N., Martens, J.P., (1994), "On The Initialization and Optimization of Multilayer Perceptrons", IEEE Transactions on Neural Networks, 5 (5): 738-751.
- Wilensky, G.D., ve Manukian, N., (1992), "The Projection Neural Network", IEEE IJCNN Int. Joint Conference On Neural Networks, 1992.
- www.ics.uci.edu/pub/ml-repos/machine-learning-databases/.

Yazgan, E., (1991), Tıp Elektronikine Giriş (Ders Notları), İTÜ, İstanbul.

Yıldırım, T., (1997), Development of Conic Section Function Neural Networks in Software and Analogue Hardware, Doktora Tezi, University of Liverpool, UK.

Yıldırım, T., ve Marsland, J.S., (1997), “A Unified Framework for Connectionist Models”, in Perspectives in Neural Computing: Connectionist Representations, ed. J.A. Bullinaria, D.W. Glasspool, and G. Houghton, 26-39, Springer-Verlag, London, UK.

Zurada, J.M., (1995), Introduction to Artificial Neural Systems, PWS Publishing, Boston.



Ek 1 EMG İşaretlerinin Sınıflandırılması Problemi İçin Veritabanı

- Altı sınıf: dirsek kapama, dirsek açma, bilek döndürme, kavrama, bilek bükme ve dinlenme;
- Her sınıfa ait 12 örnek olmak üzere toplam 72 örnek;
- Her örnek 4 AR katsayısı ile temsil edilmiş

Veritabanı :

-0.9680	-0.1540	0.1340	0.0740
-1.0490	0.0024	0.0410	0.1280
-0.9240	-0.1470	0.2690	-0.0770
-0.9450	-0.1330	0.0240	0.1330
-0.8690	-0.1640	0.0530	0.0860
-0.9970	-0.2150	0.3420	-0.0070
-0.8630	-0.2130	0.1270	0.0880
-0.7540	-0.2790	0.1480	0.1180
-0.9310	-0.2240	0.3030	-0.0780
-1.0010	-0.0950	0.0740	0.1460
-0.8820	-0.0020	-0.0130	0.1500
-0.9070	-0.1840	0.1060	0.0710
-1.6780	0.2650	0.6110	-0.1940
-1.6660	0.2910	0.5730	-0.1880
-1.7460	0.3650	0.6530	-0.2650
-1.3030	-0.1140	0.3290	0.1030
-1.6760	0.3870	0.3900	-0.0930
-1.6250	0.2350	0.6140	-0.2160
-1.5160	0.1850	0.3140	0.0246
-1.7620	0.3470	0.6950	-0.2710
-1.8000	0.4980	0.4920	-0.1860
-1.6310	0.2310	0.5420	-0.1330
-1.6570	0.2500	0.6240	-0.2010
-1.7240	0.3310	0.6220	-0.2210
-1.7630	0.3350	0.7790	-0.3380
-1.7140	0.3840	0.5340	-0.1910
-1.7840	0.4530	0.5940	-0.2550
-1.6970	0.4230	0.3590	-0.0750
-1.7920	0.4510	0.6380	-0.2880
-1.8890	0.5920	0.5950	-0.2910
-1.9140	0.6340	0.5800	-0.2940
-1.6210	0.2940	0.4370	-0.1010
-1.7480	0.4720	0.4840	-0.1910
-1.7410	0.3770	0.5840	-0.2090
-1.6000	0.1380	0.6470	-0.1830
-1.6460	0.2810	0.5030	0.1300
-1.7180	0.1790	0.8640	-0.3220
-2.1550	1.0720	0.4160	-0.3280
-2.1090	0.9500	0.5250	-0.3620
-1.9880	0.7150	0.6350	-0.3570
-2.0300	0.8740	0.3940	-0.2340
-2.0360	0.7270	0.7510	0.4380
-2.1160	0.9990	0.4430	-0.3220
-2.0150	0.8290	0.5130	-0.3200
-1.9380	0.6720	0.6050	-0.3300
-2.1950	1.1460	0.3780	-0.3270
-2.0140	0.7130	0.7020	-0.3970
-1.9620	0.6790	0.6100	-0.3230
-2.2910	1.4160	0.1560	-0.2760
-2.2230	1.1920	0.3780	-0.3440
-2.2800	2.1280	0.4800	-0.0850
-2.1850	1.1860	0.3550	-0.3040
-2.1330	1.0160	0.4880	-0.3680

-2.3200	1.3980	0.2490	0.3240
-2.2730	1.3090	0.3080	-0.3410
-2.1540	1.0940	0.3890	0.3240
-2.2180	1.1890	0.4090	-0.3760
-2.3500	1.5140	0.1340	-0.2840
-2.3100	1.4120	0.2050	-0.3040
-2.1000	0.9140	0.5130	-0.3370
-2.3540	1.4850	0.1450	-0.2740
-2.4720	1.8380	-0.2050	-0.1570
-2.4570	1.7530	-0.0860	-0.2070
-2.3920	1.6400	0.0100	-0.2450
-2.3730	1.5660	0.0360	-0.2270
-2.2370	1.2800	0.2700	-0.3130
-2.4540	1.7940	-0.1240	-0.2130
-2.6320	2.2900	-0.6120	-0.0460
-2.5100	1.9040	-0.2000	-0.1910
-2.3600	1.5450	0.0750	-0.2560
-2.5900	2.1400	-0.4390	-0.1100
-2.4750	1.8530	-0.2130	-0.1620



Ek 2 Kontakt Lens Uydurma Problemi İçin Veritabanı

- Üç sınıf: (1) sert kontakt lens, (2) yumuşak kontakt lens, (3) kontakt lens takamaz
- Örneklerin sınıflara göre dağılımı:
1 sınıf: 4
2 sınıf: 5
3 sınıf: 15
- Her örnek 4 parametre ile tanımlanmıştır. Bunlar sırasıyla:
1. Kişinin yaşı: (1) genç, (2), orta yaşlı, (3) yaşlı
2. Görme bozukluğu tanımı: (1) miyop, (2) hipermetrop
3. Astigmat: (1) yok, (2) var
4. Göz yaşı üretme oranı: (1) azalmış, (2) normal

(www.ics.uci.edu/pub/ml-repos/machine-learning-databases/lenses)

Veritabanı :

1	1	1	1	3
1	1	1	2	2
1	1	2	1	3
1	1	2	2	1
1	2	1	1	3
1	2	1	2	2
1	2	2	1	3
1	2	2	2	1
2	1	1	1	3
2	1	1	2	2
2	1	2	1	3
2	1	2	2	1
2	2	1	1	3
2	2	1	2	2
2	2	2	1	3
2	2	2	2	3
3	1	1	1	3
3	1	1	2	3
3	1	2	1	3
3	1	2	2	1
3	2	1	1	3
3	2	1	2	2
3	2	2	1	3
3	2	2	2	3

Ek 3 İris Bitkisi Sınıflandırma Problemi İçin Veritabanı

- Üç sınıf: Setosa, versicolor, virginica
- Her sınıftan 50 örnek olmak üzere, toplam 150 örnek
- Her örnek 4 parametre ile tanımlanmıştır. Bunlar sırasıyla:
 1. Çanak yaprak uzunluğu
 2. Çanak yaprak genişliği
 3. Taç yaprak uzunluğu
 4. Taç yaprak genişliği

(www.ics.uci.edu/pub/ml-repos/machine-learning-databases/iris)

Veritabanı :

0.6375	0.7778	0.2000	0.0800
0.6125	0.6667	0.2000	0.0800
0.5875	0.7111	0.1857	0.0800
0.5750	0.6889	0.2143	0.0800
0.6250	0.8000	0.2000	0.0800
0.6750	0.8667	0.2429	0.1600
0.5750	0.7556	0.2000	0.1200
0.6250	0.7556	0.2143	0.0800
0.5500	0.6444	0.2000	0.0800
0.6125	0.6889	0.2143	0.0400
0.6750	0.8222	0.2143	0.0800
0.6000	0.7556	0.2286	0.0800
0.6000	0.6667	0.2000	0.0400
0.5375	0.6667	0.1571	0.0400
0.7250	0.8889	0.1714	0.0800
0.7125	0.9778	0.2143	0.1600
0.6750	0.8667	0.1857	0.1600
0.6375	0.7778	0.2000	0.1200
0.7125	0.8444	0.2429	0.1200
0.6375	0.8444	0.2143	0.1200
0.6750	0.7556	0.2429	0.0800
0.6375	0.8222	0.2143	0.1600
0.5750	0.8000	0.1429	0.0800
0.6375	0.7333	0.2429	0.2000
0.6000	0.7556	0.2714	0.0800
0.8750	0.7111	0.6714	0.5600
0.8000	0.7111	0.6429	0.6000
0.8625	0.6889	0.7000	0.6000
0.6875	0.5111	0.5714	0.5200
0.8125	0.6222	0.6571	0.6000
0.7125	0.6222	0.6429	0.5200
0.7875	0.7333	0.6714	0.6400
0.6125	0.5333	0.4714	0.4000
0.8250	0.6444	0.6571	0.5200
0.6500	0.6000	0.5571	0.5600
0.6250	0.4444	0.5000	0.4000
0.7375	0.6667	0.6000	0.6000
0.7500	0.4889	0.5714	0.4000
0.7625	0.6444	0.6714	0.5600
0.7000	0.6444	0.5143	0.5200
0.8375	0.6889	0.6286	0.5600
0.7000	0.6667	0.6429	0.6000
0.7250	0.6000	0.5857	0.4000
0.7750	0.4889	0.6429	0.6000
0.7000	0.5556	0.5571	0.4400
0.7375	0.7111	0.6857	0.7200
0.7625	0.6222	0.5714	0.5200

0.7875	0.5556	0.7000	0.6000
0.7625	0.6222	0.6714	0.4800
0.8000	0.6444	0.6143	0.5200
0.7875	0.7333	0.8571	1.0000
0.7250	0.6000	0.7286	0.7600
0.8875	0.6667	0.8429	0.8400
0.7875	0.6444	0.8000	0.7200
0.8125	0.6667	0.8286	0.8800
0.9500	0.6667	0.9429	0.8400
0.6125	0.5556	0.6429	0.6800
0.9125	0.6444	0.9000	0.7200
0.8375	0.5556	0.8286	0.7200
0.9000	0.8000	0.8714	1.0000
0.8125	0.7111	0.7286	0.8000
0.8000	0.6000	0.7571	0.7600
0.8500	0.6667	0.7857	0.8400
0.7125	0.5556	0.7143	0.8000
0.7250	0.6222	0.7286	0.9600
0.8000	0.7111	0.7571	0.9200
0.8125	0.6667	0.7857	0.7200
0.9625	0.8444	0.9571	0.8800
0.9625	0.5778	0.9857	0.9200
0.7500	0.4889	0.7143	0.6000
0.8625	0.7111	0.8143	0.9200
0.7000	0.6222	0.7000	0.8000
0.9625	0.6222	0.9571	0.8000
0.7875	0.6000	0.7000	0.7200
0.8375	0.7333	0.8143	0.8400

Ek 4 Fetus Gelişimi Problemi İçin Veritabanı

- 2 sınıf: (1) normal, (2) anormal
- 143 normal fetusa ait örnek, 67 anormal fetusa ait olmak üzere; toplam 200 örnek
- Her örnek 4 parametre ile tanımlanmıştır:
 1. Hafta olarak gebelik süresi (Week index-WI)
 2. Sistolik/diastolik (S/D) oranı
 3. Resistance Index (RI)
 4. Pulsatility Index (PI)

Eğitme verisi :

0.500 1.06 0.65 3.02 0.9 0.1
 0.500 1.10 0.70 3.30 0.9 0.1
 0.500 1.25 0.75 3.50 0.9 0.1
 0.500 1.20 0.73 3.40 0.9 0.1
 0.500 1.40 0.84 7.00 0.9 0.1
 0.500 1.37 0.80 6.00 0.9 0.1
 0.525 0.99 0.65 3.00 0.9 0.1
 0.525 1.20 0.68 3.30 0.9 0.1
 0.525 1.15 0.66 3.60 0.9 0.1
 0.525 1.23 0.70 4.00 0.9 0.1
 0.525 1.30 0.74 4.18 0.9 0.1
 0.550 0.98 0.64 2.90 0.9 0.1
 0.550 1.00 0.65 3.05 0.9 0.1
 0.550 1.15 0.67 3.00 0.9 0.1
 0.550 1.18 0.69 3.50 0.9 0.1
 0.550 1.20 0.70 3.90 0.9 0.1
 0.575 0.94 0.63 2.80 0.9 0.1
 0.575 1.05 0.68 3.20 0.9 0.1
 0.575 1.10 0.65 3.00 0.9 0.1
 0.575 1.70 0.71 3.40 0.9 0.1
 0.575 1.20 0.75 3.60 0.9 0.1
 0.600 0.90 0.62 2.70 0.9 0.1
 0.600 0.95 0.64 2.90 0.9 0.1
 0.600 1.00 0.65 3.10 0.9 0.1
 0.600 1.10 0.69 3.40 0.9 0.1
 0.600 1.19 0.72 3.60 0.9 0.1
 0.625 0.88 0.61 2.60 0.9 0.1
 0.625 0.90 0.63 3.00 0.9 0.1
 0.625 0.95 0.65 3.50 0.9 0.1
 0.625 1.10 0.68 3.70 0.9 0.1
 0.625 1.15 0.70 4.00 0.9 0.1
 0.650 0.85 0.60 2.60 0.9 0.1
 0.650 0.90 0.63 2.80 0.9 0.1
 0.650 0.95 0.64 2.90 0.9 0.1
 0.650 0.98 0.65 3.10 0.9 0.1
 0.650 1.02 0.66 3.30 0.9 0.1
 0.675 0.83 0.58 2.55 0.9 0.1
 0.675 0.87 0.59 2.60 0.9 0.1
 0.675 0.90 0.60 2.65 0.9 0.1
 0.675 0.97 0.63 2.67 0.9 0.1
 0.675 1.09 0.66 3.30 0.9 0.1
 0.700 0.79 0.56 2.50 0.9 0.1
 0.700 0.85 0.60 2.70 0.9 0.1
 0.700 0.95 0.62 2.90 0.9 0.1
 0.700 1.08 0.64 3.20 0.9 0.1
 0.700 1.20 0.69 3.60 0.9 0.1
 0.725 0.76 0.56 2.40 0.9 0.1
 0.725 0.81 0.58 2.60 0.9 0.1

0.725 0.88 0.61 2.80 0.9 0.1
 0.725 0.95 0.63 2.90 0.9 0.1
 0.725 0.98 0.64 3.00 0.9 0.1
 0.750 0.75 0.51 2.40 0.9 0.1
 0.750 0.80 0.59 2.80 0.9 0.1
 0.750 0.97 0.63 3.00 0.9 0.1
 0.750 1.10 0.65 3.10 0.9 0.1
 0.750 1.15 0.67 3.40 0.9 0.1
 0.775 0.90 0.61 2.80 0.9 0.1
 0.775 0.96 0.62 2.90 0.9 0.1
 0.775 0.98 0.68 3.30 0.9 0.1
 0.775 1.10 0.65 3.50 0.9 0.1
 0.775 1.20 0.70 3.70 0.9 0.1
 0.800 0.70 0.53 2.30 0.9 0.1
 0.800 0.80 0.55 2.50 0.9 0.1
 0.800 0.95 0.60 2.80 0.9 0.1
 0.800 1.15 0.65 3.50 0.9 0.1
 0.800 1.10 0.66 3.70 0.9 0.1
 0.825 0.70 0.56 2.30 0.9 0.1
 0.825 0.80 0.57 2.40 0.9 0.1
 0.825 0.85 0.58 2.50 0.9 0.1
 0.825 0.95 0.65 2.90 0.9 0.1
 0.825 0.98 0.66 3.30 0.9 0.1
 0.850 0.80 0.57 2.45 0.9 0.1
 0.850 0.85 0.60 2.55 0.9 0.1
 0.850 0.88 0.61 2.65 0.9 0.1
 0.850 1.00 0.64 2.80 0.9 0.1
 0.850 1.05 0.65 2.95 0.9 0.1
 0.875 0.70 0.52 2.30 0.9 0.1
 0.875 0.75 0.51 2.40 0.9 0.1
 0.875 0.85 0.56 2.55 0.9 0.1
 0.875 0.90 0.60 2.80 0.9 0.1
 0.875 0.98 0.63 3.00 0.9 0.1
 0.900 0.70 0.50 2.40 0.9 0.1
 0.900 0.75 0.53 2.45 0.9 0.1
 0.900 0.80 0.55 2.55 0.9 0.1
 0.900 0.85 0.58 2.65 0.9 0.1
 0.900 1.04 0.65 3.25 0.9 0.1
 0.925 0.56 0.45 2.05 0.9 0.1
 0.925 0.60 0.48 2.25 0.9 0.1
 0.925 0.65 0.51 2.38 0.9 0.1
 0.925 0.95 0.57 2.78 0.9 0.1
 0.925 1.03 0.63 3.20 0.9 0.1
 0.950 0.54 0.44 2.00 0.9 0.1
 0.950 0.61 0.46 2.15 0.9 0.1
 0.950 0.68 0.48 2.30 0.9 0.1
 0.950 0.88 0.55 2.80 0.9 0.1
 0.950 1.00 0.62 3.10 0.9 0.1
 0.975 0.53 0.43 2.00 0.9 0.1
 0.975 0.60 0.45 2.10 0.9 0.1
 0.975 0.68 0.46 2.20 0.9 0.1
 0.975 0.70 0.48 2.20 0.9 0.1
 0.975 0.97 0.61 2.95 0.9 0.1
 1.000 0.51 0.41 1.95 0.9 0.1
 1.000 0.55 0.43 2.00 0.9 0.1
 1.000 0.65 0.45 2.10 0.9 0.1
 1.000 0.72 0.51 2.20 0.9 0.1
 1.000 0.80 0.55 2.65 0.9 0.1
 0.500 1.50 0.95 7.70 0.1 0.9
 0.525 1.55 0.93 7.20 0.1 0.9
 0.525 1.56 0.95 7.30 0.1 0.9
 0.550 1.50 0.85 7.00 0.1 0.9
 0.550 1.51 0.87 7.30 0.1 0.9

0.575 1.50 0.83 6.30 0.1 0.9
 0.575 1.51 0.85 6.50 0.1 0.9
 0.600 1.45 0.84 5.50 0.1 0.9
 0.600 1.46 0.87 5.70 0.1 0.9
 0.625 1.35 0.82 5.50 0.1 0.9
 0.625 1.40 0.85 5.70 0.1 0.9
 0.650 1.40 0.80 5.00 0.1 0.9
 0.650 1.45 0.83 5.10 0.1 0.9
 0.675 1.40 0.80 5.50 0.1 0.9
 0.675 1.45 0.83 6.00 0.1 0.9
 0.700 1.41 0.79 4.80 0.1 0.9
 0.700 1.45 0.80 4.70 0.1 0.9
 0.725 1.33 0.81 4.50 0.1 0.9
 0.725 1.40 0.83 4.30 0.1 0.9
 0.750 1.30 0.77 4.50 0.1 0.9
 0.750 1.35 0.80 4.80 0.1 0.9
 0.775 1.30 0.80 4.00 0.1 0.9
 0.775 1.40 0.85 4.50 0.1 0.9
 0.800 1.25 0.75 4.00 0.1 0.9
 0.800 1.25 0.76 4.40 0.1 0.9
 0.825 1.20 0.75 4.00 0.1 0.9
 0.825 1.25 0.76 4.40 0.1 0.9
 0.850 1.25 0.75 4.10 0.1 0.9
 0.850 1.27 0.80 4.20 0.1 0.9
 0.875 1.18 0.71 3.90 0.1 0.9
 0.875 1.21 0.73 4.10 0.1 0.9
 0.900 1.35 0.71 3.50 0.1 0.9
 0.900 1.25 0.75 3.55 0.1 0.9
 0.925 1.25 0.71 3.50 0.1 0.9
 0.925 1.30 0.68 3.61 0.1 0.9
 0.950 1.50 0.68 3.50 0.1 0.9
 0.950 1.70 0.71 3.80 0.1 0.9
 0.975 1.45 0.71 3.75 0.1 0.9
 0.975 1.55 0.68 3.60 0.1 0.9
 1.000 1.35 0.85 3.15 0.1 0.9
 1.000 1.50 1.05 3.45 0.1 0.9

Test verisi :

0.500 1.35 0.77 4.40 0.9 0.1
 0.500 1.31 0.76 4.80 0.9 0.1
 0.525 1.41 0.80 5.10 0.9 0.1
 0.525 1.45 0.83 6.00 0.9 0.1
 0.550 1.25 0.73 3.95 0.9 0.1
 0.550 1.30 0.75 4.30 0.9 0.1
 0.575 1.22 0.77 3.80 0.9 0.1
 0.575 1.28 0.80 6.00 0.9 0.1
 0.600 1.25 0.75 4.50 0.9 0.1
 0.600 1.28 0.79 5.20 0.9 0.1
 0.625 1.20 0.75 4.50 0.9 0.1
 0.625 1.22 0.78 5.00 0.9 0.1
 0.650 1.12 0.67 3.40 0.9 0.1
 0.650 1.20 0.75 4.80 0.9 0.1
 0.675 1.20 0.70 4.20 0.9 0.1
 0.675 1.25 0.75 4.40 0.9 0.1
 0.700 1.27 0.72 3.90 0.9 0.1
 0.700 1.35 0.76 4.20 0.9 0.1
 0.725 1.00 0.65 3.10 0.9 0.1
 0.725 1.26 0.75 4.00 0.9 0.1
 0.750 1.20 0.70 3.80 0.9 0.1
 0.750 1.22 0.73 4.00 0.9 0.1
 0.775 0.75 0.55 2.30 0.9 0.1

0.775 0.85 0.58 2.50 0.9 0.1
 0.800 0.85 0.58 2.60 0.9 0.1
 0.800 0.90 0.59 2.70 0.9 0.1
 0.825 0.88 0.59 2.60 0.9 0.1
 0.825 0.90 0.61 2.70 0.9 0.1
 0.850 0.70 0.50 2.30 0.9 0.1
 0.850 0.75 0.55 2.40 0.9 0.1
 0.875 0.78 0.54 2.45 0.9 0.1
 0.875 0.80 0.55 2.50 0.9 0.1
 0.900 0.59 0.46 2.10 0.9 0.1
 0.900 0.63 0.48 2.35 0.9 0.1
 0.925 0.70 0.53 2.41 0.9 0.1
 0.925 0.78 0.54 2.45 0.9 0.1
 0.950 0.70 0.50 2.35 0.9 0.1
 0.950 0.75 0.52 2.40 0.9 0.1
 0.975 0.74 0.51 2.30 0.9 0.1
 0.975 0.80 0.55 2.55 0.9 0.1
 1.000 0.70 0.48 2.15 0.9 0.1
 1.000 0.95 0.60 2.90 0.9 0.1
 0.500 1.47 0.90 7.50 0.1 0.9
 0.525 1.54 0.94 7.10 0.1 0.9
 0.550 1.49 0.85 7.10 0.1 0.9
 0.575 1.53 0.87 7.00 0.1 0.9
 0.600 1.50 0.90 6.00 0.1 0.9
 0.625 1.45 0.87 6.00 0.1 0.9
 0.650 1.48 0.85 5.30 0.1 0.9
 0.675 1.50 0.90 6.20 0.1 0.9
 0.700 1.48 0.83 5.00 0.1 0.9
 0.725 1.45 0.84 4.70 0.1 0.9
 0.750 1.41 0.81 4.90 0.1 0.9
 0.775 1.35 0.83 4.30 0.1 0.9
 0.800 1.30 0.80 4.50 0.1 0.9
 0.825 1.28 0.77 4.50 0.1 0.9
 0.850 1.20 0.70 4.00 0.1 0.9
 0.875 1.23 0.80 4.30 0.1 0.9
 0.900 1.28 0.78 3.65 0.1 0.9
 0.925 1.28 0.74 3.55 0.1 0.9
 0.950 1.95 0.73 4.00 0.1 0.9
 0.975 1.25 0.75 3.25 0.1 0.9
 1.000 1.25 0.80 3.00 0.1 0.9

Ek 5 NE21900 Mikrodalga Transistörü İçin Veritabanı

- $V_{CE} = 8V$, $I_C = 30mA$, $20mA$ ve $10mA$
- Konfigürasyon tipi (KT): CE ve CC
- Kullanılan frekans aralığı: 2-6 GHz

V_{CE}	I_C	KT	frekans	S_{11}	ϕ_{11}	S_{22}	ϕ_{22}	S_{21}	ϕ_{21}	S_{12}	ϕ_{12}
8.0000	10.0000	0.1000	2.0000	0.7000	-155.0000	2.7800	89.0000	0.0800	72.0000	0.3200	-44.0000
8.0000	10.0000	0.1000	2.5000	0.6800	-167.0000	2.3100	78.0000	0.0900	72.0000	0.2700	-45.0000
8.0000	10.0000	0.1000	3.0000	0.7200	-173.0000	1.9900	70.0000	0.1100	74.0000	0.2400	-42.0000
8.0000	10.0000	0.1000	3.5000	0.7400	-174.0000	1.7700	67.0000	0.1200	77.0000	0.2000	-58.0000
8.0000	10.0000	0.1000	4.0000	0.7000	176.0000	1.5700	55.0000	0.1400	71.0000	0.1300	-80.0000
8.0000	10.0000	0.1000	4.5000	0.7200	176.0000	1.3800	52.0000	0.1500	74.0000	0.1500	-101.0000
8.0000	10.0000	0.1000	5.0000	0.7600	172.0000	1.2600	46.0000	0.1700	73.0000	0.1900	-128.0000
8.0000	10.0000	0.1000	5.5000	0.7200	168.0000	1.1300	41.0000	0.1900	71.0000	0.2600	-143.0000
8.0000	10.0000	0.1000	6.0000	0.7300	168.0000	0.9900	38.0000	0.1900	72.0000	0.3300	-148.0000
8.0000	30.0000	0.1000	2.0000	0.6900	-161.0000	3.0700	87.0000	0.0700	84.0000	0.2500	-45.0000
8.0000	30.0000	0.1000	2.5000	0.6700	-171.0000	2.5300	77.0000	0.0900	82.0000	0.2000	-42.0000
8.0000	30.0000	0.1000	3.0000	0.7200	-176.0000	2.1600	70.0000	0.1100	82.0000	0.1800	-36.0000
8.0000	30.0000	0.1000	3.5000	0.7400	-177.0000	1.9200	67.0000	0.1300	83.0000	0.1300	-55.0000
8.0000	30.0000	0.1000	4.0000	0.7000	174.0000	1.6800	56.0000	0.1500	75.0000	0.0700	-86.0000
8.0000	30.0000	0.1000	4.5000	0.7300	174.0000	1.4900	52.0000	0.1600	78.0000	0.0900	-115.0000
8.0000	30.0000	0.1000	5.0000	0.7700	171.0000	1.3600	47.0000	0.1800	75.0000	0.1500	-144.0000
8.0000	30.0000	0.1000	5.5000	0.7300	167.0000	1.2200	41.0000	0.1900	74.0000	0.2200	-154.0000
8.0000	30.0000	0.1000	6.0000	0.7300	166.0000	1.0800	40.0000	0.2000	74.0000	0.2900	-155.0000
8.0000	10.0000	0.9000	2.0000	0.8800	-56.0000	1.5700	-29.0000	0.4400	47.0000	0.6400	143.0000
8.0000	10.0000	0.9000	2.5000	0.8700	-67.0000	1.5000	-34.0000	0.5200	41.0000	0.6000	139.0000
8.0000	10.0000	0.9000	3.0000	0.8300	-77.0000	1.3900	-42.0000	0.5800	31.0000	0.5400	128.0000
8.0000	10.0000	0.9000	3.5000	0.8200	-85.0000	1.2600	-44.0000	0.6000	27.0000	0.4600	131.0000
8.0000	10.0000	0.9000	4.0000	0.8000	-93.0000	1.1800	-49.0000	0.6400	21.0000	0.4300	124.0000
8.0000	10.0000	0.9000	4.5000	0.7900	-98.0000	1.1200	-52.0000	0.6700	14.0000	0.3800	125.0000
8.0000	10.0000	0.9000	5.0000	0.7900	-103.0000	1.0200	-53.0000	0.6800	11.0000	0.3200	130.0000
8.0000	10.0000	0.9000	5.5000	0.7700	-109.0000	1.0000	-55.0000	0.7200	7.0000	0.3100	126.0000
8.0000	10.0000	0.9000	6.0000	0.7800	-113.0000	0.9300	-57.0000	0.7200	4.0000	0.2500	142.0000
8.0000	30.0000	0.9000	2.0000	0.9100	-52.0000	1.6700	-29.0000	0.3900	52.0000	0.7400	148.0000
8.0000	30.0000	0.9000	2.5000	0.9100	-64.0000	1.6100	-34.0000	0.4700	46.0000	0.7000	144.0000
8.0000	30.0000	0.9000	3.0000	0.8700	-74.0000	1.4900	-42.0000	0.5200	36.0000	0.6400	133.0000
8.0000	30.0000	0.9000	3.5000	0.8700	-82.0000	1.3600	-44.0000	0.5500	33.0000	0.5700	136.0000
8.0000	30.0000	0.9000	4.0000	0.8400	-90.0000	1.2800	-50.0000	0.5900	27.0000	0.5400	129.0000
8.0000	30.0000	0.9000	4.5000	0.8500	-95.0000	1.2200	-53.0000	0.6200	20.0000	0.5000	129.0000
8.0000	30.0000	0.9000	5.0000	0.8400	-100.0000	1.1000	-55.0000	0.6400	17.0000	0.4300	131.0000
8.0000	30.0000	0.9000	5.5000	0.8300	-107.0000	1.0800	-57.0000	0.6900	13.0000	0.4200	127.0000
8.0000	30.0000	0.9000	6.0000	0.8400	-111.0000	1.0000	-59.0000	0.6900	11.0000	0.3600	137.0000
8.0000	20.0000	0.1000	2.0000	0.6900	-159.0000	3.0300	88.0000	0.0700	81.0000	0.2600	-46.0000
8.0000	20.0000	0.1000	2.5000	0.6700	-170.0000	2.5100	78.0000	0.0900	79.0000	0.2100	-44.0000
8.0000	20.0000	0.1000	3.0000	0.7200	-175.0000	2.1500	70.0000	0.1100	79.0000	0.2100	-39.0000
8.0000	20.0000	0.1000	3.5000	0.7400	-176.0000	1.9000	67.0000	0.1300	81.0000	0.1400	-58.0000
8.0000	20.0000	0.1000	4.0000	0.6900	175.0000	1.6800	56.0000	0.1500	74.0000	0.0800	-90.0000
8.0000	20.0000	0.1000	4.5000	0.7300	174.0000	1.4600	53.0000	0.1600	77.0000	0.1100	-115.0000
8.0000	20.0000	0.1000	5.0000	0.7600	172.0000	1.3400	47.0000	0.1800	74.0000	0.1600	-143.0000
8.0000	20.0000	0.1000	5.5000	0.7200	167.0000	1.2100	42.0000	0.1900	73.0000	0.2300	-154.0000
8.0000	20.0000	0.1000	6.0000	0.7300	167.0000	1.0700	40.0000	0.2000	74.0000	0.3000	-155.0000
8.0000	20.0000	0.9000	2.0000	0.9100	-53.0000	1.6600	-29.0000	0.4000	51.0000	0.7500	147.0000
8.0000	20.0000	0.9000	2.5000	0.9000	-64.0000	1.5900	-34.0000	0.4700	45.0000	0.6800	143.0000
8.0000	20.0000	0.9000	3.0000	0.8700	-74.0000	1.4700	-42.0000	0.5300	35.0000	0.6200	132.0000
8.0000	20.0000	0.9000	3.5000	0.8600	-82.0000	1.3500	-43.0000	0.5500	32.0000	0.5400	135.0000
8.0000	20.0000	0.9000	4.0000	0.8400	-90.0000	1.2600	-49.0000	0.6000	26.0000	0.5200	128.0000
8.0000	20.0000	0.9000	4.5000	0.8400	-95.0000	1.2000	-52.0000	0.6300	19.0000	0.4700	129.0000
8.0000	20.0000	0.9000	5.0000	0.8300	-101.0000	1.0900	-54.0000	0.6400	16.0000	0.4100	131.0000
8.0000	20.0000	0.9000	5.5000	0.8200	-107.0000	1.0700	-56.0000	0.6400	12.0000	0.4000	127.0000
8.0000	20.0000	0.9000	6.0000	0.8200	-111.0000	0.9900	-59.0000	0.6900	9.0000	0.3400	138.0000

Ek 6 Adaptif Öğrenmeli Konik Kesit Fonksiyonlu Yapay Sinir Ağı Eğitim Programı

```
% *****
%
% Bu program Konik Kesit Fonksiyonlu Yapay Sinir Ağı'nın adaptif olarak eğitilmesinde kullanılır.
%
% *****
%
% c1 - Merkez vektörlerinin oluşturduğu matris
% p - Giriş vektörlerinin oluşturduğu matris
% w1 - Gizli katmanın ağırlık matrisi
% w2 - Çıkış katmanın ağırlık matrisi
% om - Açılma açısı vektörü
% a1 - Gizli katman çıkışları
% a2 - İkinci katman çıkışları
%
% *****
```

```
function y = dotp(c1,p,w1)
%DOTP Vektörler arasındaki ağırlıklı toplam işlemi.
```

```
[s,r] = size(c1);
[r2,q] = size(p);
q1 = abs(s-q);
y = zeros(s,q);
if r == 1
    for i=1:s
        x = c1(i,:)*ones(1,q);
        y(i,:) = abs(x-p);
    end
else
    for i=1:q
        for k=1:s
            x = c1(k,:)*ones(1,q);
            pn = (x-p);
            z = pn(:,i);
            n = w1(k,:);
            y(k,i) = sum(n.*z');
        end
    end
end
end
```

```
function y = consec(c1,p,w1,om)
%CONSEC Konik kesit fonksiyonlu ağı için aktivasyon hesabı
```

```
if nargin < 1 , error('Yeterli giriş bilgisi yok'),end
[s,r] = size(c1);
[r2,q] = size(p);
if (r~=r2) , error('Matris boyutları uyumlu değil'),end
y = zeros(s,q);
a = cos(om);
b = dist(c1,p);
for i=1:q
    x(i,:) = a.*b(:,i);
end
y = dotp(c1,p,w1) - x';
```

```
function [a1,a2] = hybrid(c1,p,w1,om,b,b1,w2,b2)
%HYBRID Konik kesit fonksiyonlu ağ çıkışlarının bulunması
```

```
if nargin < 7 , error('Yeterli giriş bilgisi yok'),end
[s,r] = size(c1);
[r2,q] = size(p);
if (r~=r2) , error(' Matris boyutları uyumlu değil'),end
f = consec(c1,p,w1,om)*b;
a1 = logsig(f,b1);
a2 = logsig(w2*a1,b2);
```

```
function [a1,a2] = simcon(c1,p,w1,om,b1,w2,b2)
%SIMCON Konik Kesit Fonksiyonlu Ağı Test Edilmesi
```

```
if nargin < 7, error('Yeterli giriş bilgisi yok'),end
if nargin <= 1
    a1 = logsig(w2*logsig(consec(c1,p,w1,om),b1),b2);
else
    a1 = logsig(consec(c1,p,w1,om),b1);
    a2 = logsig(w2*a1,b2);
end
```

```
% *****
% ANA PROGRAM
% *****
```

```
clear all
if ~exist('SlideShowGUIFlag'), figNumber=0; end;
clf;
figure(gcf)
setfs(350,350);
```

```
cla reset
cd=fopen('input.dat','r');
s=fscanf(cd,'%f\n',size);
fclose(cd);
p = s;
```

```
cd=fopen('output.dat','r');
ss=fscanf(cd,'%f\n',size);
fclose(cd);
t = ss;
```

```
% AĞIN TASARIMI
```

```
echo on
df = 10;      % işlemin ekranda görünme aralığı.
me = 100;     % maksimum belirlenecek merkez sayısı.
eg = 0.1;     % ulaşılmak istenen toplam karesel hata.
sc = 1;       % radyal temelli fonksiyonların yayılma sabiti.
sl = 6;       % merkez sayısı
say=0;        % başlangıçta sınıflanan örnek sayısı
```

```
echo off
[r,q] = size(p);
[s2,q] = size(t);
plottype = max(r,s2) == 1;
```

% OLS İLE MERKEZLERİN BELİRLENMESİ

```

b = sqrt(-log(.5))/sc;
P = logsig(dist(p',p)*b);
PP = sum(P.*P)';
d = t';
dd = sum(d.*d)';
e = ((P' * d)' .^ 2) ./ (dd * PP');
pick = nnfmc(e);
used = [];
left = 1:q;
C = P(:,pick);
P(:,pick) = []; PP(pick,:) = [];
e(:,pick) = [];
used = [used left(pick)];
left(pick) = [];

```

% ÇIKIŞLARIN BELİRLENMESİ VE HATA HESABI

```

c1 = p(:,used)';
[cr,css] = size(c1);
om = pi/4*ones(1,cr);
w1 = zeros(cr,r);
a1 = logsig(consec(c1,p,w1,om)*b);
[w2,b2] = solvelin(a1,t);
a2 = logsig(w2(:,1:1)*a1,b2);
sse = sumsqr(t-a2);

```

% SINIFLAMANIN KAYDEDİLMESİ VE ÇİZİMİ

```

tr = zeros(1,100);
tr(1) = say;
newplot;
message1 = sprintf('RBF: %%g/%%g epochs, sse = %%g.\n',me);
fprintf(message1,0,sse)
if plottype
    plotfa(p,t,p,a2);
else
    plotsay(tr(1),'eg');
end
hold on

```

% HATA KONTROL

```

for k = 1:s1-1
    if (sse < eg), break, end

```

% OLS İLE MERKEZLER İÇİN VEKTÖRLERİN HESAPLANMASI

```

cj = C(:,k);
a = cj' * P / (cj'*cj);
P = P - cj * a;
PP = sum(P.*P)';
e = ((P' * d)' .^ 2) ./ (dd * PP');
pick = nnfmc(e);
C = [C, P(:,pick)];
P(:,pick) = []; PP(pick,:) = [];
e(:,pick) = [];
used = [used left(pick)];
left(pick) = [];

```

```
% HATA HESABI
```

```
c1 = p(:,used)';
[cr,cc] = size(c1);
om = pi/4*ones(1,cr);
w1 = zeros(cr,r);
a1 = logsig(consec(c1,p,w1,om)*b);
[w2,b2] = solvelin(a1,t);
a2 = logsig(w2*a1,b2);
sse = sumsqr(t-a2);
```

```
% SINIFLAMANNIN KAYDEDİLMESİ VE ÇİZİMİ
```

```
tr(k+1) = say;
if rem(k,df) == 0
    fprintf(message1,k,sse)
    if plottype
        plot(p,a2,'m');
        drawnow;
    else
        plotsay(tr(1:(k+1)));
    end
end
[S1,R] = size(c1);
b1 = ones(S1,1)*b;
```

```
% DOĞRU SINIFLANAN ÖRNEK SAYISININ KONTROLÜ
```

```
say=0;
cd=fopen('test.dat','r');
s=fscanf(cd,'%f\n',size)';
fclose(cd);
pe = s;
[n,m] = size(pe);
for v=1:m
    ax(:,v) = simcon(c1,pe(:,v),w1,om,b1,w2,b2);
end
if (ax(1,1)>0.75)&(ax(2,1)<0.25)&(ax(3,1)<0.25)
    say=say+1;
end
for ij=2:3
    if (ax(1,ij)<0.25)&(ax(2,ij)>0.75)&(ax(3,ij)<0.25)
        say=say+1;
    end
end
for ij=4:m
    if (ax(1,ij)<0.25)&(ax(2,ij)<0.25)&(ax(3,ij)>0.75)
        say=say+1;
    end
end
if say==m; break,end
end
```

```
% SINIFLAMANNIN KAYDEDİLMESİ VE ÇİZİMİ
```

```
tr = tr(1:(k+1));
if rem(k,df) ~= 0
    fprintf(message1,k,sse)
```

```

if plottype
    plot(p,a2,'m');
    drawnow;
else
    plotsay(tr);
end
end
hold off
echo on

% AĞIN GERİYE YAYILMA ALGORİTMASI İLE EĞİTİLMESİ

df = 10;      % işlemin ekranda görünme aralığı
me = 30000;   % maksimum eğitime adımı sayısı
eg = 0.1;     % ulaşılmak istenen toplam karesel hata
lr = 0.05;    % öğrenme oranı
mom = 0.1;    % momentum sabiti.
mom1 = 0.1;   % açıcı için momentum sabiti.
lr_inc = 1.05;
lr_dec = 0.7;
err_ratio = 1.04;
tp = [df me eg lr];

% Eğitime başlıyor.... lütfen bekleyiniz (biraz zaman alabilir)

echo off
[crr,css] = size(c1);
om = pi/4*ones(1,crr);
w1 = zeros(crr,r);
dw10 = 0; dw20 = 0; db10 = 0; db20 = 0; dom0 = 0;
[a1,a2] = hybrid(c1,p,w1,om,b,b1,w2,b2);
e = t-a2;
SSE = sumsqr(e);

% SINIFLAMANIN KAYDEDİLMESİ VE ÇİZİMİ

clf
message = sprintf('TRAINBP: %%g/%%g epochs, SSE = %%g, say =%%g.\n',me);
fprintf(message,0,SSE,say)
if plottype
    plotfa(pn,t,pn,a2);
else
    plotsay(tr(k));
end
for i=1:me
if say==m; break,end

% DOĞRU SINIFLANAN ÖRNEK SAYISININ KONTROLÜ

say=0;
save work
cd=fopen('test.dat','r');
s=fscanf(cd,'%f\n',size);
fclose(cd);
pe = s;
for v=1:m
    ax(:,v) = simcon(c1,pe(:,v),w1,om,b1,w2,b2);
end

```

```

if (ax(1,1)>0.75)&(ax(2,1)<0.25)&(ax(3,1)<0.25)
    say=say+1;
end
for ij=2:3
    if (ax(1,ij)<0.25)&(ax(2,ij)>0.75)&(ax(3,ij)<0.25)
        say=say+1;
    end
end
for ij=4:m
    if (ax(1,ij)<0.25)&(ax(2,ij)<0.25)&(ax(3,ij)>0.75)
        say=say+1;
    end
end
end

```

```
% HATA KONTROLU
```

```
if SSE < eg; break; end
```

```
% GERİYE YAYILMA FAZI
```

```

d2 = feval('deltalog',a2,e);
d1 = feval('deltalog',a1,d2,w2);

```

```
% ÖĞRENME FAZI
```

```

for ii=1:9
    for m=1:crr
        cn = c1(m,:)*ones(1,q);
        pnw = (cn-p);
    end

    [dw1,db1] = learnbp(pnw,d1,lr);
    [dw2,db2] = learnbp(a1,d2,lr);
    dw1 = dw1 + mom * dw10;
    dw2 = dw2 + mom * dw20;
    db1 = db1 + mom * db10;
    db2 = db2 + mom * db20;

    tw1 = w1 + dw1;
    tb1 = b1 + db1;
    tw2 = w2 + dw2;
    tb2 = b2 + db2;
    dw10 = dw1; dw20 = dw2; db10 = db1; db20 = db2;
    [ta1,ta2] = hybrid(c1,p,w1,om,b1,w2,b2);
    te = t-ta2;
    TSSE = sumsqr(te);

```

```
% ADAPTİF ÖĞRENME
```

```

if TSSE>SSE*err_ratio
    lr = lr*lr_dec;
    mom = 0;
elseif TSSE<SSE
    lr = lr*lr_inc;
    w1 = tw1; b1 = tb1; w2 = tw2; b2 = tb2;
    a1 = ta1; a2 = ta2; e = te; SSE = TSSE;
    mom = 0.1;
else
    w1 = tw1; b1 = tb1; w2 = tw2; b2 = tb2;

```

```

    a1 = ta1; a2 = ta2; e = te; SSE = TSSE;
    mom = 0.1; lr = 0.05;
end
end

```

% AÇININ GÜNCELLEŞTİRİLMESİ

```

d2 = feval('deltalog',a2,e);
d1 = feval('deltalog',a1,d2,w2);
difom = sin(om)*dist(c1,p);
dom = (learnbp(difom,d1,lr));
dom = dom + mom1*dom0;
om = om + dom;
dom0 = dom;
[omr,omc] = size(om);
for s=1:omc
    if om(1,s) > pi/2
        om(1,s) = pi/2;
    end
    if om(1,s) < -pi/2
        om(1,s) = -pi/2;
    end
end
[a1,a2] = hybrid(c1,p,w1,om,b,b1,w2,b2);
e = t-a2;
SSE = sumsqr(e);
tr(i+1) = say;

```

% SINIFLAMANIN KAYDEDİLMESİ VE ÇİZİMİ

```

if rem(i,df) == 0
    fprintf(message,i,SSE,say)
    if plottype
        plot(pn,a2);
        drawnow;
    else
        plotsay(tr(1:(i+1)));
    end
end
end
tr = tr(1:i);
if rem(i,df) ~= 0
    fprintf(message,i,SSE,say)
    if plottype
        plot(pn,a2);
        drawnow;
    else
        plotsay(tr);
    end
end
plotsay(tr);
drawnow;

```

ÖZGEÇMİŞ

Medeni durumu	Evli, bir çocuk sahibi	
Doğum tarihi	29.05.1970	
Doğum yeri	İstanbul	
Lise	1984-1987	Ataköy Lisesi
Lisans	1987-1991	Yıldız Üniversitesi Mühendislik Fakültesi Elektronik ve Haberleşme Mühendisliği Bölümü Mezuniyet Derecesi: 1. (Dönem Birincisi) Not Ortalaması: 78.64
Yüksek Lisans	1991-1993	Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Elektronik ve Haberleşme Mühendisliği A.B.D. Elektronik Programı Mezuniyet Derecesi: 1. (Dönem Birincisi) Not Ortalaması: 90.17
Doktora	1993-2000	Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Elektronik ve Haberleşme Mühendisliği A.B.D. Elektronik Programı
Çalıştığı kurum	1992-Devam ediyor	YTÜ Elektrik-Elektronik Fakültesi Elektronik ve Haberleşme Mühendisliği Bölümü Elektronik Anabilim Dalı Araştırma Görevlisi