

KOCAELİ ÜNİVERSİTESİ * FEN BİLİMLERİ ENSTİTÜSÜ

**YAPAY SİNİR AĞLARI KULLANARAK EEG MODELLEMESİ
VE UYARILMIŞ BEYİN POTANSİYELLERİNİN ÇIKARTIMI**

YÜKSEK LİSANS TEZİ

İrfan SAĞDINÇ

Ana Bilim Dalı: Elektronik-Bilgisayar Eğitimi

28242

OCAK 1998

**YAPAY SİNİR AĞLARI KULLANARAK EEG MODELLEMESİ VE
UYARILMIŞ BEYİN POTANSİYELLERİNİN ÇIKARTIMI**

YÜKSEK LİSANS TEZİ
İrfan SAĞDINÇ

Tezin Enstitüye Verildiği Tarih : 09/02/1998
Tezin Savunulduğu Tarih : 02/02/1998

Tez Danışmanı

Üye

Üye

Y. Doç. Dr. Mehmet Engin

(.....)

Doç. Dr. Mehmet Korürek

(.....)

Y. Doç. Dr. Sıtkı Öztürk

(.....)

OCAK 1998

**T.C. YÜKSEK ÖĞRETİM KURULU
DOKÜMANTASYON MERKEZİ**

YAPAY SİNİR AĞLARI KULLANARAK EEG MODELLEMESİ VE UYARILMIŞ BEYİN POTANSİYELLERİNİN ÇIKARTIMI

İrfan SAĞDINÇ

Anahtar Kelimeler: EEG Modelleme, UP Çıkartımı, Doğrusal Olmayan Sistem Tanılama

Özet: Uyarılmış beyin potansiyelleri(UP'ler) klinik tanı için çok önemlidir. UP'ler genellikle süregelen elektroensefalogram (EEG) içerisine gömülü durumdadırlar ve çıkartımları en çok bilinen UP elde etme yöntemi olan grup ortalaması almakla elde edilir. Bu yöntemle, kullanışlı bir dalga şekli elde edebilmek için yüzlerce segment UP ölçümü gerekmektedir. Bu çalışmada, tek segmentlik kayıtlarda uyarılmış potansiyellerin tesbiti için, ölçülen aktiviteyi süregelen EEG ve uyarılmış potansiyel parçalarına ayıran bir yöntem kullanılmıştır. Çalışmada ölçülen aktivitelerin süregelen kısmı Yapay Sinir Ağı (YSA) tarafından kestirildikten sonra kestirim hataları UP olarak yorumlanmıştır. EEG işaretinin zamanla-değişen nitelikte olması nedeni ile, EEG'ye ait bir parametrik model çıkarabilmek için dinamik yaklaşımların kullanılması tercih edilmelidir. Bu nedenle uyarı sonrası EEG aktivitesinin, uyarı öncesi EEG ile eğitilen bir dinamik YSA ile kestirilebileceği düşünülmüştür. Uyarı sonrası aktiviteyi kestirmek için "Gerçek-Zaman Geri-Dönüşümlü" ağ ve "Çok Katmanlı Algılayıcı", dinamik ve statik yaklaşımlar olarak kullanılmıştır. Bu yöntemlerde, UP'ler bir segmentlik veri için çıkartılmaya çalışılmış ve sonuçlar ortalama alma yöntemi ile zaman ve frekans düzleminde karşılaştırılmıştır. Tez çalışmasında, yöntemlerin doğruluğunu test etmek amacı ile bir simülasyon çalışması gerçekleştirilmiştir. Simülasyon çalışmasında, EEG ve UP ayrı ayrı üretildikten sonra bir model altında toplanmıştır. Yöntemlerin yapay veri ile test edilmesinden sonra bir segmentlik UP'lerin çıkartılması için gerçek işaret kullanılmıştır.

EEG MODELLING AND EXTRACTION OF EVOKED BRAIN POTENTIALS USING ARTIFICIAL NEURAL NETWORKS

İrfan SAĞDINÇ

Keyword: EEG Modelling, UP Detection, Non-linear System Identification

Abstract: Evoked potentials (EPs) of the brain are very meaningful for clinical diagnosis. The EPs are usually embedded in ongoing electroencephalogram (EEG). The traditional method of EP extraction is take ensemble averaging. In this method, to obtain a usable waveform thousands of evoked potential measurements are required. In this study, for the investigation of evoked potentials in single segment measurements, a method which separates the measured activity into spontaneous part and evoked potentials is used. Spontaneous part of the measured activity was estimated by ANN and estimation errors have been interpreted as EPs. Since EEG's are time-varying signals, dynamic approaches must be used to obtain accurate results. Therefore, it was considered that poststimulus EEG activity may be estimated by a dynamic ANN which is trained by prestimulus data. In order to estimate the poststimulus activity, real-time recurrent neural network (RNN) and multilayer perceptron (MP) were used. In these approaches, EPs have successfully been extracted in single segment and results compared with the ensemble averaging in time and frequency domain. In order to test the accuracy of the methods a simulation study has been performed. In simulation study, EEG and EPs were produced separately and then combined in a model. After testing the methods with artificial data, real signal was used for the single segment extraction of EPs.

ÖNSÖZ ve TEŞEKKÜR

Bilindiđi gibi insan beyni çok karmaşık bir yapıya sahiptir ve insan ođlunun bu güne kadar beyin hakkında öğrenebildikleri oldukça sınırlıdır. Bu nedenle, son yıllarda bu konuda bir çok bilimsel çalışma gerçekleştirilmiştir. Yapay sinir ađları bir çok bilim dalında olduđu gibi Tıp Elektronikinde de rahatlıkla kullanılacak bir hesaplama yöntemi olmasından dolayı, bu konunun bir çok üstünlüğünün beyin işaretlerinin incelenmesinde de verimli bir şekilde kullanılacağı açıktır. Bu alandaki çalışmalara katkıda bulunması dileđiyle.

Bana bu konudaki yardım ve önerilerinde dolayı Sayın Hocam, Y. Doç. Dr. Mehmet Engin bey'e, değerli arkadaşım Arş. Gör. Sekan Kıracı'a , Aileme ve tüm çalışma arkadaşlarıma teşekkürlerimi sunarım.

İÇİNDEKİLER

ÖZET.....	i
ABSTRACT.....	ii
ÖNSÖZ ve TEŞEKKÜRLER.....	iii
İÇİNDEKİLER.....	iv
SİMGELER LİSTESİ VE KISALTMALAR.....	vii
ŞEKİLLER LİSTESİ.....	ix
TABLolar LİSTESİ.....	xi
BÖLÜM 1.GİRİŞ.....	1
BÖLÜM.2 ELEKTROENSEFALOGRAM VE UYARILMIŞ POTANSİYELLER.....	4
2.1.Elektroensefalogram(EEG).....	4
2.1.1.EEG işaretinin frekans bileşenleri.....	5
2.2.Uyarılmış Potansiyeller.....	6
2.2.1.Uyarılmış potansiyellerin ölçümü.....	7
2.2.2.Uyarılmış potansiyel çeşitleri.....	7
2.2.2.1.Somatik duysal uyarılmış potansiyeller (SUP).....	8
2.2.2.2.Duysal (akustik) uyarılmış potansiyeller (DUP).....	9
2.2.2.3.Görsel uyarılmış potansiyeller (GUP)	10
BÖLÜM.3 SİSTEM TANILAMA.....	12
3.1.Sistem Tanılama.....	12
3.1.1.Sistem tanılamamanın amaçları.....	13
3.1.1.1.Kestirilebilirlik.....	13
3.1.1.2.Yorumlanabilirlik.....	14

3.1.1.3.Uyarlanabilirlik.....	14
3.2.Sistem Tanılama Yaklaşımları.....	14
3.2.1. Parametrik modeller.....	15
3.2.1.1.Doğrusal modeller.....	15
3.2.1.2.Doğrusal olmayan modeller.....	16
BÖLÜM 4. YAPAY SİNİR AĞLARI	18
4.1. YSA Kullanımının Üstünlükleri.....	18
4.2.Sinir Hücresi Modelleri.....	19
4.3.Aktivasyon Fonksiyonları	21
4.3.1.Eşik fonksiyonu	22
4.3.2.Parçalı doğrusal fonksiyon.....	22
4.3.3.Sigmoid fonksiyonu	22
4.4.Yapay Sinir Ağı Yapıları	23
4.4.1.Tek katmanlı ileri yönlü ağlar.....	23
4.4.2.Çok katmanlı ileri yönlü ağlar.....	24
4.4.3.Geri dönüşümlü ağlar	25
4.4.4.Kafes Yapılı Ağlar.....	27
4.5.Yapay Sinir Ağlarında Öğrenme ve Çeşitleri.....	28
BÖLÜM 5.EŞLEŞTİRMELİ AĞLAR VE ÇOK KATMANLI	
ALGILAYICILAR.....	29
5.1.Ölçü Fonksiyonu ve Yakınsama Doğruluğu.....	29
5.2.Çok Katmanlı Algılayıcılar.....	30
5.2.1.Çok katmanlı ağların özellikleri.....	31
5.3.Geri Yansıtma Algoritması.....	32
5.4.Öğrenme Modları.....	35
5.4.1.Anlık öğrenme modu	35
5.4.2.Grup öğrenme modu.....	36
BÖLÜM 6. DİNAMİK YAPILI YAPAY SİNİR AĞLARI.....	37
6.1.Dinamik Yapay Sinir Ağı Tanımı	37

6.2. Gerçek Zaman Geri Dönüşümlü Ağlar.....	39
6.3.1. Gerçek zaman geçici eğitici öğrenme algoritması.....	41
BÖLÜM 7 YAPAY SİNİR AĞLARI İLE DOĞRUSAL OLMAYAN	
SİSTEM TANILAMA.....	44
7.1. Yapay Sinir Ağları ile Modelleme.....	44
7.1.1. Doğrusal durum.....	45
7.1.2. Doğrusal olmayan durum.....	46
7.2. Doğrusal Olmayan Dinamik Sistem Modelleri.....	47
7.3. Sistem Tanılama Yöntemleri.....	51
7.3.1. Paralel tanılama	51
7.3.2. Seri-Paralel tanılama.....	52
7.4. Model Gerçekleştirme.....	53
7.5. Simülasyon Çalışması.....	54
BÖLÜM 8 YAPAY SİNİR AĞLARI İLE EEG MODELLEME VE	
UYARILMIŞ POTANSİYELLERİN ÇIKARTIMI.....	57
8.1. İşaret Simülasyonu.....	57
8.1.1. EEG modeli.....	58
8.1.2. UP modeli.....	59
8.1.3. Birleşik model.....	60
8.2. Gerçek İşaret Özellikleri	61
8.3. Yapay Sinir Ağı İle Uyarı Sonrası EEG	
Kestirimi ve UP Çıkartımı.....	61
8.4. Sonuçlar.....	63
8.5. Sonuçların Değerlendirilmesi.....	70
KAYNAKLAR.....	72
EKLER.....	75
ÖZGEÇMİŞ.....	79

SİMGELER DİZİNİ ve KISALTMALAR

α_n	: AR katsayısı
β_n	: Rassal madel için harici giriş katsayısı
$f(\cdot)/g(\cdot)$: Doğrusal olmayan fonksiyon
$\hat{y}(n)$: Model çıkışı
w_{ji}	: YSA ağırlık parametre vektörü
$\vartheta(\cdot)$: Doğrusal olmayan aktivasyon fonksiyonu
θ_k	: Eşik değeri
δ_i	: Gradyent vektörü
Δw_{ji}	: Ağırlık değişimi
ε	: Maliyet Fonksiyonu
Z^{-1}	: Birim gecikme elemanı
e_j	: Hata
π_{kl}^j	: Geri dönüşümlü ağ için dinamik değişken
η	: Öğrenme oranı
μ	: Bias
v	: Beyaz gürültü
v_j	: Ağ çıkışı için içsel aktivasyon
Φ_{xy}	: İlişki fonksiyonu
N	: Ayrık zaman işaret boyu
a^T	: AR katsayı vektörü
c^T	: IIR filtre katsayı vektörü
EEG	: Elektroensefalogram
UP	: Uyarılmış Potansiyel
YSA	: Yapay sinir ağı
SUP	: Somatik uyarılmış potansiyel
AUP	: Akustik uyarılmış potansiyel

GUP	: Görsel uyarılmış potansiyel
AR	: Öz bağımlı model
NAR	: Doğrusal olmayan öz bağımlı model
ARX	: Öz bağımlı ekstra girişli model
ARMA	: Öz bağımlı kayan ortalamalı model
ARMAX	: Öz bağımlı, kayan ortalamalı ekstra girişli model



ŞEKİLLER LİSTESİ

Şekil 2.1 Gözün açık ve kapalı olmasına göre EEG' nin eğişiimi.....	6
Şekil 2.2 Uyarma potansiyeli ölçüm sisteminin blok diyagramı.....	8
Şekil 2.3 Çeşitli görsel UP'ler.....	8
Şekil 3.1 Basit tanılama düzeneği.....	13
Şekil 4.1 Sinir hücrelerinin doğrusal olmayan modeli.....	20
Şekil 4.2 Eşik değerinin toplayıcı çıkışına ve toplam aktivasyona etkisi.....	21
Şekil 4.3 Eşik değerinin hücre modeli üzerinde değişik gösterimi.....	22
Şekil 4.4 Genel aktivasyon fonksiyonları.....	24
Şekil 4.5 Tek katmanlı ileri yönlü ağ yapısı.....	24
Şekil 4.6 Bir gizli katmana sahip ve tam bağlantılı ileri yönlü çok katmanlı ağ yapısı.....	25
Şekil 4.7 Tek katmanlı geri dönüşümlü ağ yapısı.....	26
Şekil 4.8 Gizli katmanı olan Geri dönüşümlü ağ yapısı.....	26
Şekil 4.9 Kafes yapılı ağlar.....	27
Şekil 4.1 Yapay sinir ağlarında öğrenmenin sınıflanması.....	28
Şekil 5.1 Hata yüzeyinin tipik şekli.....	30
Şekil 5.2 İki temel işaretin çok katmanlı bir ağ parçası üzerinde gösterilişi.....	31
Şekil 5.3 Üç katmanlı ağın blok diyagram olarak gösterilişi.....	32
Şekil 5.5 Geri yansıtma algoritmasının diyagram olarak gösterilişi.....	33
Şekil 6.1 Statik çok katmanlı ağın blok yapısı.....	37
Şekil 6.2 Doğrusal olmayan kestirimci olarak çok katmanlı ağın kullanılışı.....	38
Şekil 6.3 Gerçek zaman geri dönüşümlü ağ yapısı.....	40
Şekil 7.1 Uyarlanabilir Doğrusal Eleman (ADALINE).....	46
Şekil 7.2 Doğrusal olmayan model olarak çok katmanlı ağ.....	47
Şekil 7.3 Doğrusal olmayan Dinamik model-I	48
Şekil 7.4 Doğrusal olmayan Dinamik model-II	49
Şekil 7.5 Doğrusal olmayan Dinamik model-III	50
Şekil 7.6 Doğrusal olmayan Dinamik model-IV.....	50

Şekil 7.7 Paralel tanılama modeli.....	51
Şekil 7.8 Seri-Paralel tanılama modeli.....	52
Şekil 7.9 Model-IV için tanılama blok şeması.....	53
Şekil 7.10 Sistem için kullanılan işaretler	55
Şekil 7.11 Sistem tanılamanın sonuçları	55
Şekil 7.12 Toplam ortalama karesel ağ hatası.....	56
Şekil 7.13 İlişki testi sonuçları.....	57
Şekil 8.1 EEG ve UP simülatörü blok diyagramı.....	58
Şekil 8.2 EEG simülasyonu için AR model çıkışı.....	59
Şekil 8.3 UP simülasyonu için farklı genlik ve fazdaki ilk üç segment UP işareti.....	60
Şekil 8.4 UP+EEG işareti için birleşik model çıkışı.....	61
Şekil 8.5 Yapay işaret için çok katmanlı ağ ile UP sezinleme sonuçları.....	64
Şekil 8.6 Yapay işaret için gerçek zamanlı ağ ile UP sezinleme sonuçları.....	65
Şekil 8.7 Gerçek işaret için çok katmanlı ağ ile UP sezinleme sonuçları.....	67
Şekil 8.8 Gerçek işaret için gerçek zamnlı ağ ile UP sezinleme sonuçları.....	69
Şekil 8.9 Tek segmentte UP sezinleme için frekans düzlemi sonuçları.....	70

TABLULAR LİSTESİ

Tablo 2.1 EEG işaretlerinin kapsadıkları frekans bantları ve isimleri.....	5
Tablo 8.1 UP çıkartımı için çeşitli yöntemlere göre SNR sonuçları.....	65



BÖLÜM 1. GİRİŞ

İnsan vücudunun en karmaşık organı olan beyin üzerinde yapılan bilimsel çalışmalar, bu kompleks yapı hakkında daha fazla bilgi edinmek amacıyla son yıllarda önemli bir ilerleme kaydetmiştir.

Beynin elektriksel aktivitesini gösteren EEG kafa derisi üzerinden veya beyin içerisinden elektrotlar kullanılarak algılanır. Bu elektriksel aktivite sinir hücreleri arasındaki akım akışı ile ortaya çıkar. İnsan beynindeki sayısı milyarları bulan ve değişik davranış sergileyen hücrelerin varlığı düşünülürse bu işaretin yorumlanması ve işlenmesinin oldukça güç bir işlem olduğu anlaşılabilir. Tıpta ,teşhis ve tedavi amacı ile EEG işaretleri önemli bir yer tutarlar.

Uyarılmış potansiyeller (UP) beş duyu organından herhangi biri ile uyarılan beyin cevabıdır. Yani beynin bir dışsal uyarıya karşılık elektriksel aktivitesini gösterirler. Bu nedenle teşhis ve tedavide oldukça geniş bir kullanım alanı vardır. En çok kullanılan uyarı çeşitleri işitsel , görsel ve somatik olanlarıdır.

Uyarı potansiyelleri genellikle ölçüm sırasında süregelen EEG içerisinde gömülü durumdadır. Bu nedenle UP'lerin bir segmentlik ölçümünde yorumlanması zorlaşmaktadır. En çok bilinen ve kullanılan UP çıkartım yöntemi grup ortalaması alma yöntemidir. Bu yöntemle işaret/gürültü miktarı iyileştirilir. Fakat bunun gerçekleştirilmesi için yüzlerce UP içeren segmentin kullanılması gereklidir. Bu hem bellek ihtiyacını arttırmakta hem de segmentlerde ani değişimlere uğrayabilen UP cevaplarının güvenilirliğini azaltmaktadır. Bu nedenle ortalama sayısını azaltmak ve kısa sürede çözüme ulaşmak için bir çok çalışma gerçekleştirilmiştir. Bir segmentlik UP değişimlerinin çıkartımı için Aunon ve McGillem (1975 , 1977) Zamanla değişmeyen, doğrusal, en küçük ortalama karesel hata filtresini önerdiler. Bu yaklaşımda EEG ve UP'nin öz ilişki karşı ilişkilerinden faydalanılmıştır. Fakat Weerd(1981) zamanla değişmeyen filtreleme yaklaşımlarının bu iş için pek uygun

olmadığını gösterdi. Ayrıca Bromm(1985) zamanla değişen filtre ile UP çıkartımını gerçekleştirdi.

Yukarıda bahsedilen yaklaşımların hepsi uyarı sonrası EEG aktivitesini kullanmaktadırlar. Fakat uyarı öncesi aktivite uyarı sonrası EEG özelliklerini de taşımaktadır. Bu nedenle UP çıkartımında uyarı öncesi EEG bilgisinin kullanılabilmesi için Spreckelsen ve Bromm (1988) Kalman Filtre Yaklaşımı ile UP çıkartımını gerçekleştirdiler. Buna benzer bir yaklaşım Gülçür ve Madani (1994) tarafından Yapay Sinir Ağı ile gerçekleştirildi. Bu yaklaşım ile EEG ve UP ayrı ayrı ele alınmış ve süregelen EEG aktivitesi uyarı öncesi EEG ile eğitilen çok katmanlı ağ ile modellenmiştir. Daha sonra çıkartılan bu model ile uyarı sonrası EEG aktivitesi kestirilmiş ve kalanlar yani hata UP olarak elde edilmiştir. Bahsedilen tüm yaklaşımlarda grup ortalaması almanın üzerinde bir sonuç elde edilememiştir. Fakat bir segmentte UP çıkartımında yeterli ilerlemeler kaydedilmiştir.

Tez çalışmasının amacı, kısa aralıklarda durağan kabul edilen ve her segmentlik ölçümde farklı davranış sergileyen EEG işaretinin dinamik modelinin çıkartılması ve uyarı sonrası EEG'nin bu model ile kestirilmesidir. Uyarı öncesi EEG'den uyarı sonrası kestirmek için, bir çok doğrusal olmayan sistemi veya zaman serisini modelleyebilen yapay sinir ağları kullanımı amaçlanmaktadır. Yapay sinir ağı olarak tez çalışmasında statik yapılı çok katmanlı ağ ve dinamik davranış sergileyen gerçek zaman geri dönüşümlü ağ kullanılacaktır. Yapay sinir ağlarının eğitimi uyarı öncesi EEG ile gerçekleştirilecektir. Eğitimi tamamlanmış ağlar ile her segment için uyarı sonrası EEG kestirilip giriş ile kestirim arasındaki fark UP olarak yorumlanmaktadır. Dinamik ve statik yapılı yapay sinir ağları ile EEG'ye ait NAR (Non-Linear Auto-Regressive) modeli elde edilecektir. Dinamik yapay sinir ağı ile her segmentinde değişik yapıda olabilecek EEG'ler için bir uyum sağlanması düşünülmüştür. Geri dönüşümlü ağların ani değişikliklere sahip olan dinamik sistemlerin modellenmesinde üstünlük göstermektedirler (Narendra 1990).

UP çıkartımı için gerçek işaret ile ağları denemeden önce bir işaret simülasyonu gerçekleştirilmektedir. Simülasyon çalışması ile bilinen işaretler üzerinde olumlu sonuçların alındığı koşullarda gerçek işaret kullanılmaktadır. Gerçekleştirilen işaret

simülasyonu EEG ve UP birbirlerinden bağımsız olarak üretilmekte ve daha sonra bir model altında toplanmaktadır.

Bir segmentlik ölçümde UP çıkartımı için yapılan tez çalışmasında sistem tanılama çerçevesinde ele alınacaktır. Yapay sinir ağlarının doğrusal olmayan dinamik sistemlerin modellemesindeki üstünlüklerinden bu konuda faydalanmak amaçlanmaktadır.

EEG ve UP hakkındaki temel bilgiler bölüm-2'de ele alınmış ve UP'nin elde edilme yöntemlerinden bahsedilmiştir. Bölüm-3'de sistem tanılama hakkında temel bilgiler verilmiştir. Ayrıca bu bölümde konuya genel yaklaşımlardan bahsedilmiş ve çok kullanılan rastsal modellere değinilmiştir. Bölüm-4'de yapay sinir ağlarına giriş yapılmış ve yapay sinir ağlarının klasik yöntemlere göre üstünlüklerinden bahsedilmiştir. Ayrıca bu bölümde temel yapay sinir ağı öğrenme yöntemlerinden bahsedilmiştir. Bölüm-5'de Yapay Sinir Ağlarında önemli yeri olan Çok Katmanlı ağlar ele alınmış ve Geri Yansıtımlı öğrenmeden bahsedilmiştir. Bölüm-6'da dinamik yapay sinir ağları konusu ele alınmış ve statik ağlarla farkları belirtilmiştir. Tez çalışmasında kullanılan dinamik ağ modeli olan Gerçek-Zaman Geri Dönüşümlü ağ ve bunun eğitim algoritmasından bahsedilmiştir. Bölüm-7'de doğrusal olmayan dinamik sistemlerin modellemesinde yapay sinir ağlarının kullanılmasından bahsedilmiş ve bir simülasyon çalışması yapılmıştır. Bölüm-8 tez çalışmasının sonuç bölümünü oluşturmaktadır. Bu bölümde çok katmanlı ve gerçek zaman ağları kullanılarak UP çıkartımı için yapay işaret elde edilmiştir. Yapay işaret ile her iki ağ denendikten sonra gerçek işaret ile UP çıkartımı gerçekleştirilmiştir.

BÖLÜM 2. ELEKTROENSEFALOGRAM VE UYARILMIŞ POTANSİYELLER

EEG işaretlerinin varlığının bilimsel şekilde kesin olarak ortaya konması, ancak elektronikteki gelişmeler sonucunda 1934'de Adrian ve Matthews tarafından, elektrodlarla alınan EEG işaretlerini kuvvetlendirip kaydedilmesi ile sağlandı. P.A. Davis, 1939'da, uyanık bir insanın EEG'sinde, ses uyarılarına karşı olan uyarılan yanıtlarının (Evoked Potential) olduğunu ortaya çıkardı.

İkinci dünya savaşını takip eden yıllarda, elektronik ve bilgisayarlarda meydana gelen büyük gelişmelerle bu alanda büyük ilerlemeler sağlanmıştır. EEG böylece klinik uygulamalarda kullanılmaya başlandı.

Uyarılmış Potansiyeller (UP) alanındaki gelişmeler, bunların genliklerinin EEG'ye nazaran küçük olmaları nedeniyle ancak yeni işaret işleme yöntemlerinin bulunmasıyla mümkün olabildi. İlk olarak 1958'de Clark ve arkadaşları, geliştirdikleri Ortalama yanıt yöntemini; işaretin (UP) gürültüye (EEG) olan oranını yükseltmek amacıyla kullandılar.

2.1. Elektroensefalogram(EEG)

Beynin sinirsel faaliyeti sonucu elde edilen biyoelektrik işaretlere elektroensefalogram (EEG) adı verilir. EEG'nin çok karmaşık bir değişim şekli vardır ve yorumlanması zordur. Yüzeyden ölçülen EEG işaretleri, alttaki birçok noktadan ve serebral korteksin oldukça geniş bir bölgesinden gelen potansiyellerin toplamından oluşur (Yazgan ve Korürek 1996).

Bugüne kadarki deneysel çalışmalar EEG frekansının kişinin zihin faaliyeti ile değiştiğini göstermiştir. Bu durum, beynin tabii ve boşta çalışma frekansı gibi bir senkronizasyon durumunu gösterir. Kişi uyanır uyanmaz veya düşünmeye başlarken alfa ritmi kaybolur ve yerine , genellikle beta bandında bir ritim oluşur. Günümüzde çoğu araştırmalar, beyindeki bu işlemi yapan mekanizmanın bulunması ile ilgilidir. Bu

konudaki çalışmalar, bazı özel durumlarda kişinin kendi EEG'si ile ilgili bilginin, ses veya ışıkla uyarı olarak geri besleme şeklinde kendisine verilmesi durumunda, EEG işaretlerinin bir miktar kontrol edebileceğini göstermiştir.

2.1.1. EEG işaretinin frekans bileşenleri

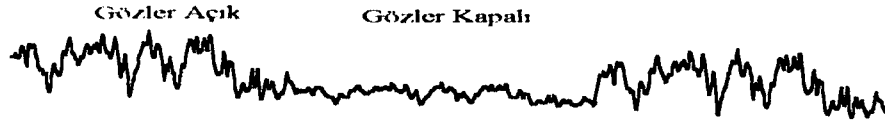
EEG işaretinin frekans bileşenleri son derece önemlidir. Farklı bölgelerden alınan benzer EEG işaretleri arasındaki faz ilişkileri de bulunmaktadır. Bu tip bilgiler, EEG'nin kaynağının incelenmesine yardımcı olur. Kafa üzerinden alınan EEG'lerin genliği tepeden tepeye 1-100 μ V ve frekans bandı ise 0.5-100 Hz'dir. Ölçümlerin beyin üzerinden doğrudan alınması durumunda genlik 10 kat kadar artmaktadır. EEG'de oluşan farklı frekansların oluşum mekanizmalarının anlamı, henüz tam olarak bilinmemektedir. EEG işaretleri periyodik olmadıklarından genlik, faz ve frekansları sürekli değişir. Bu nedenle, anlamlı bir veri elde edebilmek için, ölçümlerin oldukça uzun bir sürede yapılması gerekir. Tablo 2.1'de EEG işaretlerinin kapsadıkları frekans bantları ve bu bantlara verilen özel isimler gösterilmektedir. (Yazgan ve Korürek 1996).

Tablo 2.1 EEG işaretlerinin kapsadıkları frekans bantları ve isimleri

Frekans Bandı ismi	Frekans aralığı(Hz)
Delta(δ)	0.5-3.5
Teta(θ)	4-7
Alfa(α)	8-12
Beta(β)	12-22
Gamma(γ)	22-30

Alfa dalgaları , 8-12 Hz arasındaki beyin dalgalandır. Uyanık, normal ve sakin kimselerde görülüp uyku durumunda yok olurlar. Uyanık kişi dikkatini özel bir faaliyete yöneltirse α dalgaları yerine, daha yüksek frekanslı fakat düşük genlikli EEG işaretleri (β dalgaları) meydana gelir. Şekil 2.1'de, parlak ışıkta gözlerin açılması ve sonra kapatılması durumundaki EEG işaretlerindeki değişim gösterilmiştir.

Beta Dalgaları , frekansları 12 Hz'in üzerindeki beyin dalgalandır. 25 Hz'e ve nadir hallerde de 50 Hz'e kadar uzanırlar. Saçlı derinin parietal (şakak bölgesi) ve frontal



Şekil 2.1 Gözün açık ve kapalı olmasına göre EEG' nin değişimi

(alın bölgesi) bölgelerinde belirgin olarak kaydedilebilir. Beta-I ve Beta-II diye ikiye ayrılırlar. Beta-II dalgalarının frekansı, beta-I'ninkinin iki mislidir ve α dalgalarında olduğu gibi zihinsel aktivitenin artması ile ortadan kalkarlar ve yerlerine düşük genlikli işaretler oluşur. Beta-II dalgaları, merkezi sinir sisteminin kuvvetli aktivasyonunda veya gerginlik hallerinde ortaya çıkar.

Teta Dalgaları, 4-7 Hz arasındaki dalgalardır. Özellikle, çocuklarda parietal ve temporal bölgelerde ortaya çıkarlar. Yetişkinlerde de hayal kırıklığı durumlarında ortaya çıkarlar. Genlikleri, $100 \mu V$ 'dan küçüktür.

Delta Dalgaları, 3.5 Hz'in altındaki beyin dalgalandır. Bazı durumlarda 1Hz'in altına da düşerler. Süt çocuklarında ve ağır organik beyin hastalıklarında görülür. Genlikleri, $100 \mu V$ 'dan küçüktür.

Gamma Dalgaları, genellikle araştırmacılar tarafından kullanılmaktadır. Genlikleri, $2 \mu V$ 'dan daha küçüktür. Kafanın merkezinde, genlikleri daha büyüktür. Uykunun karakteristik belirtisini taşırlar.

2.2. Uyarılmış Potansiyeller

Yapılan araştırmalar, normal bir insanın algıladığı tüm enformasyonun %70'ini görme, %15'ini işitme ve %10'unu da dokunma duyusu ile elde ettiğini göstermiştir. Duyular ile ilgili bozuklukların (Özellikle yeni doğan çocuklarda) mümkün merteye erken belirlenmesi, tedavinin erken başlamasını ve dolayısıyla başarı şansını arttırmaktadır.

Uyarılmış Potansiyellerin ölçülmesiyle, görme, işitme ve dokunma duyularının normallik seviyeleri belirlenebilmektedir. Beynin, görsel, duysal ve elektriksel uyarıya karşı cevabının normal olup olmaması, uyarmanın beyne ulaştığını gösteren UP'deki tepe ve vadilerin oluşma sürelerinin, normal değerlerinden olan farklılıkların ölçülmesiyle belirlenir. UP'lerin ölçülmesiyle, sinir sisteminin fonksiyonları, invazive

(doğrudan) olmayan objektif bir yöntemle belirlenmiş olur. Kanama, tümör gibi sinir sitemleriyle ilgili bozuklukların erken teşhisi mümkün olmaktadır. UP'ler elektrodlar yardımıyla algılanır. Algılanan işaretin gürültü içerisine gömülü olması nedeniyle, İşaret-Gürültü oranının artırılması gereklidir.

UP'lerin özelliklerinin büyük oranda bilinmesine karşın, kaynakları ve oluşum mekanizmaları, beynin elektriksel aktivitesinin oluşumunu açıklayan temel bir teorinin olmaması nedeniyle açık bir şekilde bilinmemektedir. Böyle bir temel teori, beyin anatomisinin karmaşık ve nörofizyolojik olayların doğrusal olmamaları nedeniyle geliştirilememiştir.

2.2.1. Uyarılmış potansiyellerin ölçümü

Şekil 2.2'de UP için bir ölçüm düzeneği gösterilmiştir. Hastanın başına bağlı olan EEG elektronları ile ölçü sistemine giriş sağlanır. Hastadan algılanan işaret kuvvetlendirildikten ve bir önfiltreden geçirildikten sonra bilgisayar sistemine girilir. Bilgisayar, istenilen zaman aralıklarında uyarı düzenlerini tetiklemektedir. Her bir uyarı sonucu oluşan ve beynin uyarıya cevabı olan işaretlerin senkron bir şekilde alınmış toplam değerleri ölçüm sayısına bölünür. UP genellikle Bilgisayar ortamına alındığında EEG ile karışmış olup başka gürültüler de içerebilir. Bu yüzden UP'lerin işaret - gürültü oranını artırılması için genellikle ortalama alma yöntemi kullanılır. Şekil 2.3'de çeşitli ortalama sayıları için UP'ler ifade edilmektedir. Şekilden görüleceği gibi ortalama sayısı arttıkça işaret-gürültü oranı iyileştirilmektedir.

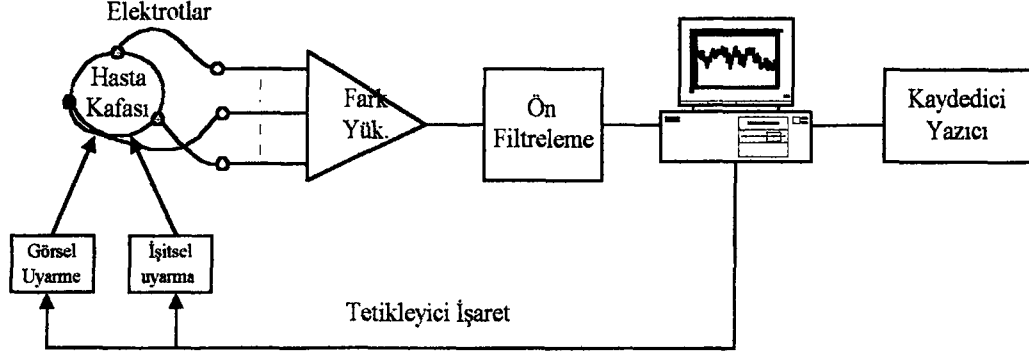
Ortalama alma yöntemi ile UP'deki ani değişikliklerin atılması sorunu ortaya çıkmaktadır. Ortalama sayısını en aza indirmek veya tek segmentte UP tespiti yapmak bu konudaki en önemli uğraşı alanıdır.

2.2.2. Uyarılmış potansiyel Çeşitleri

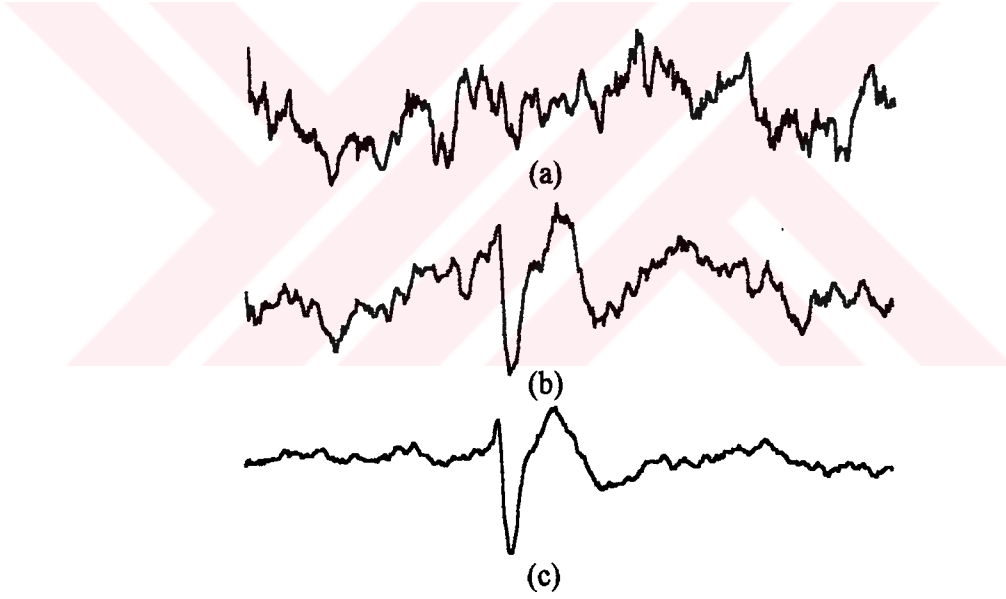
2.2.2.1. Somatik duysal uyarılmış potansiyeller (SUP)

SUP'lerin algılanması, somatik duysal sistemin fonksiyonunun incelenmesinde kullanılır. Bu tür bir sistemin kaynağı, uyarıyı sinirlerde dürtülere dönüştüren reseptörlerdir. Bu dürtü dizisi sinir lifleri üzerinden omuriliğin arka köklerine gelir ve

birtakım işlemlerden sonra da beyin kabuğuna (kortekse) ulaşır. SUP'ler kalın, ilik içeren duyarlı liflerin mekanik ve elektriksel uyarımı sonucu meydana gelirler. Bu lifler, derinin, kasların ve eklemlerin mekanik reseptörlerinden çıkarlar. Uyarı, sinir yolu boyunca olmak şartıyla, vücudun herhangi bir yerine uygulanabilir.



Şekil 2.2 Uyarma potansiyeli ölçüm sisteminin blok diyagramı



Şekil 2.3 Çeşitli Görsel UP' ler

- a) İşlenmemiş işaret
- b) 30 ortalama sonucu
- c) 90 ortalama sonucu

Senkron uyarılmış olan bölgelerin ölçümünde, ölçüm bölgesinin yerine ve büyüklüğüne göre SUP'ın maksimum genlik değeri, birkaç mikrovolt mertebesine ulaşır. Bu SUP'ın, diğer biyolojik işaretler ve ölçüm düzeninin gürültü işaretleri

tarafından bastırılabilceđi anlamına gelir. Ortamda, uyarıdan bađımsız olan ve rasgele bir karaktere sahip olan gürültü işaretleri söz konusu olduđundan işaret-gürültü oranını iyileştirmek amacıyla, senkron uyarılar için SUP'in ortalamasının alınması yöntemi kullanılır.

Bu alandaki en önemli parametre, uyarı ile ortalama SUP bileşenleri arasındaki gecikmelerin ölçülmesiyle elde edilir, Söz edilen parametrelerin belirli bir uyarı ve ölçüm durumu için elde edilen normal değerleriyle yapılan karşılaştırma sonuçlarından yola çıkarak bir tanı konulabilir. Küçük genlikli SUP'lerin algılanmasında, 10^3 - 10^4 oranında bir kuvvetlendirme yapılmalıdır.

SUP ölçümleri, vücudun somatik duysal sisteminin, objektif ve nicel bir değerlendirmesinin yapılmasına imkan verir. Somatik duysal yol boyunca birçok uyarma ve ölçüm yeri seçilerek, gecikme miktarı belirlenebilir; Çevresel ve merkezi uyarı iletim hızları hesaplanabilir. Bir veya birkaç bölgede, uyarı iletim hızında normal değerine göre gözlenen sapma, merkezi sinir sisteminde hastalıklı bir durumun varlığını göstergesidir. Böyle bir durumda hastalıklı bölgenin yeri tespit edilebilir ve ilgili kısım uzunluğu belirlenebilir. Bu şekilde tedavide de daha iyi bir yaklaşım sağlanır.

2.2.2.2. Duysal (akustik) uyarılmış potansiyeller (AUP)

Akustik uyarılmış potansiyeller, işitme duyusunun elektriksel aktivitesi sonucu meydana gelirler. Dış ve orta kulak bölümlerinin, ses uyarısını mekanik titreşimlere dönüştürdüğü akustik işaretlerden elde edilirler. Daha sonra, dalgalar şeklinde ana membran üzerinden salyangoz (cochlea) içinde dağılmış olan iç ve dış saç hücrelerine kadar yayılırlar. Salyangozun (kohleanın) akustik özelliđi nedeniyle, burada bulunan saç hücreleri, ses dalgalarını frekansa çevirir. İşitme sınırı yaklaşık olarak 30000 sinir lifinden oluşur ve beyin sapı, orta beyin ve beyin kabuđu ile birlikte merkezi işitme yolunu oluşturur.

AUP'lerin elde edilmesi için, geniş bandlı ses uyarıları veya darbe genişliđi birkaç 100 s olan ses darbeleri uygulanır. "Click" ses uyarıları olarak bilinen uyarılar, geniş bandlı

oluşları nedeniyle, yüksek genlikli AUP'ler üretirler ve işaret-gürültü oranını iyileştirirler ayrıca bu ölçüm süresinin kısılmasına neden olur.

AUP'lerin büyük band genişliğine (100Hz-2kHz) sahip olmaları ve maksimum genliklerinin 20 nV-20 µV arasında oluşu nedeniyle, ölçüm düzenlerinin de bu özel koşullara uygun olması gerekir. Yeterli bir işaret kalitesini elde edebilmek için yaklaşık 6000 AUP cevabı alınmalıdır.

Duyu organlarıyla ilgilenen fizyologlar, otologlar, nörologlar ve psikiyatristler, akustik uyarılmış potansiyellerden tanı koymaya yönelik bilgiler elde ederler. Tanı koymada, erken, ortalama (normal) ve gecikmiş uyarı cevapları, teşhiste önemlidir. Duyu organlarıyla ilgilenen fizyoloji dalında, işitme sinir hücrelerinin iletim özellikleri sonucu meydana gelen bileşenlerin genlikleri ve gecikmeleri anlam taşımaktadır.

2.2.2.3. Görsel uyarılmış potansiyeller (GUP)

Görsel uyarılmış potansiyeller denince, görme sinirlerinin toplam elektriksel aktivitesi ve görsel bir uyarı karşısında korteksteki görme merkezinde oluşan cevap anlaşılır. Görsel bir uyarı, gözün dış tabakası, mercek veya göz yuvarlağının içini dolduran esas madde üzerinden gözün iç yüzeyini çevreleyen retinaya (ağ tabaka) ulaşır. Retinada, gelen ışığı biyokimyasal yollarla membran gerilimine dönüştüren ve homojen olmayan bir dağılıma sahip koni ve çubuk hücreleri bulunmaktadır. Uyarı sonucu bu hücrelerde oluşan potansiyel değişimlerini gangliyon hücreleri, görme sinirlerine göndermeden önce işlemeye tabi tutarlar. Elde edilen görsel bilgi, görme sinirleri üzerinden korteksteki görme merkezine ulaşır, Diğer duyu kanallarından farklı olarak görme sinirlerinde, işaretin karakterine göre önemli ölçüde farklılıklar gösteren, kompleks bir bilgi taşınması söz konusudur. Bu nedenle, her GUP şekli, kullanılan görsel uyarının özellikleri ile doğrudan ilgilidir.

GUP'lerin elde edilmesinde genellikle, çok kısa süreli parlak ışık uyarıları ve kontrast uyarı olarak da satranç tahtası modeli kullanılır. Kontrast uyarı olarak, yatay veya dikey yönlendirilmiş sinüs dalgaları türünde bir takım işaret katarları da kullanılmaktadır. Kontrast uyarılar hastaya genellikle bir ekran üzerinde sunulur.

GUP'ler, görsel korteks bölgesinde, yukarıda belirtilen türde parlaklık ve kontrast uyarılarına karşı bir cevap olarak oluşur. Elde edilen GUP'un özellikleri aşağıdaki uyarı şartlarına bağlıdır.

- Model üzerindeki açık ve koyu bölgelerin yer değişim frekansı,
- Modelin büyüklüğü,
- Modelin üzerindeki bölgeler arasındaki kontrastlık derecesi,
- Ekranın ışık yoğunluğu şiddeti,
- Deney ortamının aydınlığı,
- Model üzerinde yapılan değişimlerde izlenen yol.

Elektromagnetik ve elektrostatik alanlar, diğer biyolojik kaynaklar, elektrod ve ölçme düzenlerinden v.s. kaynaklanan bozucu faktörlere ek olarak GUP'lerin ölçümünde, yine mümkün olduğunca önlenmesi gereken diğer etkiler de mevcuttur. GUP'lerin genlik değeri, bakış yönüne büyük ölçüde bağlıdır. Bakış yönünün uygunluğunun kontrol edilmesiyle işaret-gürültü oranının iyileştirilmesi sağlanabilir. Bir diğer istenmeyen olay da bilinçsizce gerçekleşen, gözün açılıp kapanma hareketleridir.

GUP'lerin frekans spektrumu, 1-50Hz arasındadır ve maksimum genlik değeri de 10-15 μ V civarındadır. Sağlıklı bir GUP şeklinin elde edilebilmesi için 100 civarında senkron uyarım cevabının ortalaması alınmalıdır.

Elde edilen uyarılmış potansiyeller, birçok negatif ve pozitif dalgalara sahip olabilirler. Tıbbi, tanımına göre pozitif olanlar aşağı doğru; negatif olanlar ise yukarı doğru gösterilir. Tepeden tepeye genlik değerleri uyarı, ile doğrudan ilgilidir.

GUP'ler gözlerin ve optik sinir sisteminin incelenmesi amacıyla kullanılır. Bu şekilde, uygun uyarılar kullanılarak, hastalık veya yaşa bağlı olarak oluşabilen görüş alanı ile ilgili bozukluklar, çift görme ve renk körlüğü gibi durumlar teşhis edilebilir.

BÖLÜM 3. SİSTEM TANILAMA

Sistemlerin davranışlarının anlaşılması ve gelecek durumlarının tahmini üzerine birçok bilimsel çalışma bulunmaktadır. Bunların birçoğu gerçek sistemi temsil edecek olan uygun modelin geliştirilmesi üzerinedir. Bu modeller çok farklı şekilde olabilirler. Örnek vermek gerekirse bir model sembolik veya fiziksel olabilir. İnsan beyninin belirli kısımlarının modellenmesi sembolik modellemedir. Fakat uçağın dentimi fiziksel bir yaklaşımdır.

Nedensel , yani ancak bir giriş uygulandığında çıkış üreten sistemler dinamik ve statik olmak üzere iki durumda ele alınırlar (Ljunk1987). Statik sistemlerde çıkış sadece o an uygulanan girişe bağlıdır. Dinamik sistemlerde ise çıkış giriş ve çıkışın geçmiş değerlerine bağlıdır.

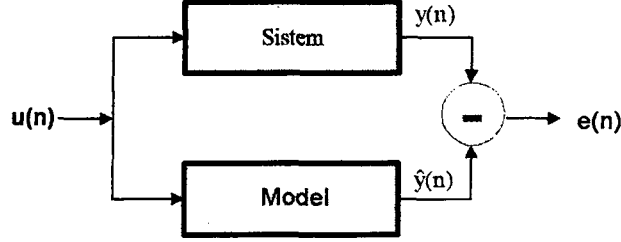
Sistem tanılamasının temel amacı gerçek sistem gibi davranan bir kestirim modeli elde etmektir. Bu tür yaklaşımlar şimdiye kadar bir çok bilim dalında başarılı bir şekilde uygulanmışlardır. Gerçek sistemi tam olarak modelleyebilmek istenen bir durum olmasına rağmen buna her zaman ulaşamayabilir. Bunun nedeni , gerçek sisteme yakınsamada bir takım kestirim bilgilerinin kullanılmasıdır.

3.1. Sistem Tanılama

Esas itibari ile sistem tanılama , ampirik bilgi ve sisteme ait öncül yapı bilgilerden dinamik sistem modelleri elde etme işlevidir. Öncül bilgiden kasıt giriş ve çıkışlardan oluşan sistemin uygun şekilde yapılandırılmasıdır. Bu model daha sonra istatistiksel anlamda ampirik bilgi elde edecek şekilde düzenlenir. Bu işlem genellikle öğrenme olarak adlandırılır. Gözlem altındaki ayrık zaman dinamik sistem için giriş u ($u = [u(n-1), u(n-2), \dots, u(n-M)]$) vektörü $y(n)$ çıkışını elde edecek şekilde ağırlıklarla düzenlenir (Haykin 1996).

Sistem tanılama gerçek sistem çıkışı olan $y(n)$ 'i kestirmek için $\hat{y}(u^T, y)$ modelini bulma ile uğraşır. Şekil 3.1'de basit bir tanılama düzeneği gözükmektedir. Şekilden

açıkça görülebileceği gibi gerçek bir sistem gibi davranan bir model elde edilmeye çalışılmaktadır. Bu model genellikle kestirim modeli olarak adlandırılır. Aşağıda birazdan bahsedilecek olan amaçları yerine getirmelidir.



Şekil 3.1 Basit tanımlama düzeneği

3.1.1.Sistem tanımlamanın amaçları

Sistem tanımlamanın temel amaçları aşağıdaki gibi özet olarak sıralanabilir(Ljunc 1987).

3.1.1.1.Kestirilebilirlik

Bu genellikle en önemli özelliktir. Gözlem bilgisinden , gerçek model çıkışına yakınsama yapacak olan ve $\hat{y}(n)$ 'i oluşturacak model yapısının bulunmasına yardımcı bir özelliktir. Uygun modelden alınan $\hat{y}(n)$ 'ler ile şekil 3.1'deki $e(n)$ hatası sıfıra doğru gider. Hatanın giderek küçülmesi istatistiksel anlamda yorumlanmalıdır. Gerçek sistem çıkışı ile model çıkışı arasındaki farkın beklenen değerinin en küçüğe indirgenmesi modelin gerçek sisteme yakınsaması ile sonuçlanır. Beklenen hata değeri denklem 3.1'deki gibi ifade edilir.

$$J_{\infty} = E[(y(n) - \hat{y}(u, y))^2] \quad (3.1)$$

Beklenen hata değerinin hesabında sonlu sayıda bilgiye sahip olabileceğimiz için beklenen hata zaman ortalaması cinsinden denklem 3.2'deki gibi tanımlanır.

$$J_N = \frac{1}{N} \sum_{n=1}^N (y(n) - \hat{y}(u, y))^2 \quad (3.2)$$

Sistem tanılamada J_N maliyet fonksiyonunun en küçüklenmesi her ne kadar etkili bir yöntem olsa da bazen yeterli sonuca ulaşmayabilmektedir. Sistem davranışının

modellenmesi sistem tanımının temeli olan genelleştirme olarak adlandırılır. Modelin geliştirilmesi seçilen model yapısı (derecesi vs.) ve kullanılan bilgilerin gürültü özellikleri ile yakından ilgilidir.

3.1.1.2.Yorumlanabilirlik

Sistem tanılama çalışılan sistem hakkında bilgi elde etme olarak yorumlanabilir. Tanılama işleminin sonunda sistem hakkında basit matematiksel ifadeler ve hangi girişlerin çıkışı etkilediği modelin doğruluğu hakkında bilgi vermektedir. Tanılama modelinin giriş çıkış ilişkisinin gerçek sistem ile uyum gösterip göstermediği tanılama açısından önemlidir.

3.1.1.3.Uyarlanabilirlik

Bazı uygulamalarda sistemin on-line olması gerekmektedir. Bu durumlarda yeni bilgi geldiğinde sistemin buna uygun cevap vermesi beklenir. Böyle sistemlerin tanılanmasında model yapısı ve gerekli algoritmalar zamanla değişen nitelikte seçilir. Zamanla değişen sistemlerin modellenmesi diğerlerine göre kestirimi daha zor bir durum sergilemektedir.

3.2.Sistem Tanılama Yaklaşımları

Belirli amaçları yerine getirmek için düşünülmüş model yapısı tanılama açısından oldukça önemlidir. Bu nedenle model yapısının sistem tanımada en önemli kriter olduğu düşünülürse bu konuda çeşitli yaklaşımların olabileceği açıktır. Bu yaklaşımlara örnek olarak doğrusal fark modelleri , polinomsal modeller , ve yapay sinir ağı yaklaşımları örnek olarak verilebilir. Bu bahsedilen tüm yaklaşımlar sistem giriş ve çıkışlarının geçmiş değerlerinden oluşan sonlu zaman serileri ile beslenirler. Bu yaklaşımlar , w ağırlık vektörü ile parametrelendirilen genel parametrik yaklaşımlar olarak da adlandırılabilir. Parametrik yaklaşım denklem 3.4 deki gibi ifade edilebilir.

$$y(\mathbf{u}, \mathbf{y}) = \hat{y}(\mathbf{x}(n), \mathbf{w}) \quad (3.4)$$

Burada $\mathbf{x}(n)$ sistem giriş ve çıkışlarının geçmiş değerlerinden oluşan kestirim modelinin giriş vektörüdür ve denklem 3.5'deki gibi açık formda ifade edilebilir.

$$\mathbf{x}(n) = \left[u(n), u(n-1), \dots, u(n-p_u), y(n-1), \dots, y(n-m_y) \right] \quad (3.5)$$

Yukarıda bahsedilen sistem tanılama yaklaşımlarının temel sorunu tanılama modeli yapısının seçimidir. Kestirim modelinin seçiminde yapay sinir ağları daha esnek özellikler sunmaktadır(Chen 1990). Yapay sinir ağları ile modelleme konusu bölüm 7 de ayrıntılı olarak ele alınmıştır. Ayrıca sistem tanılamada Parametrik Olmayan yaklaşımları da bulunmaktadır. Bu konu tez konusunun dışında kaldığı için üzerinde durulmamıştır.

3.2.1. Parametrik modeller

Şimdiye kadar bir çok farklı parametrik model önerilmiştir(Ljung 1987). Parametrik yaklaşımlarda giriş ve çıkışın geçmiş değerlerinden sistem çıkışını kestirim esas alınmıştır. Bu işlem esasen giriş uzayından çıkış uzayına bir eşleştirme işlemidir. Bu açıdan bakıldığında model giriş çıkış bağlantı uzayında bir yüzey tanımlamaktadır. Ampirik bilgi seti bu yüzeyde dağılmış durumdadır. Sistem tanılama bahsedilen uzayda bulunan bu bilgiye bir yüzey uydurma olarak düşünülebilir. Bu düşünce ile aslında çok değişkenli bir fonksiyon yakınsamasından bahsedilmektedir. Bahsedilen yüzeye yakınsama sağlayacak model için o modelin doğrusal olup olmayışı önemlidir.

3.2.1.1. Doğrusal modeller

Doğrusal ayrık zaman sistemlerinin giriş ile çıkışları arasındaki en temel ilişki doğrusal fark eşitlikleri ile kurulabilir. Böyle bir modele ait fark denklemi denklem 3.6'daki gibidir.

$$\hat{y}(n) = -\alpha_1 y(n-1) + \dots + \alpha_P y(n-P) + \beta_1 u(n) + \dots + \beta_M u(n-M) + e(n) \quad (3.6)$$

Burada, $e(n)$ beyaz gürültüdür ve modelleme hatasını gösterir. P ve M modelde kullanılan sırası ile giriş ve çıkışın kaç geçmiş değerinin kullanılacağını belirten derecelerdir. Model derecesi modellenecek sistemin çıkışından alınan işaret ile derece tespit yöntemlerine göre belirlenir(Haykin 1996). Denklemdeki fark eşitliği öz bağlantımlı extra girişli model olarak bilinir(ARX). Bu model denklem 3.7'deki gibi matris formunda da gösterilebilir(Ljung 1987).

$$\hat{y}(n) = \mathbf{X}^T(n)\mathbf{w} \quad (3.7)$$

Burada , parametre ve giriş vektörleri açık biçimde aşağıdaki gibi ifade edilirler.

$$\mathbf{w} = [\alpha_1, \alpha_2, \dots, \alpha_p, \beta_1, \beta_2, \dots, \beta_M]$$

$$\mathbf{x}(n) = [y(n-1), y(n-2), \dots, y(n-P), u(n), u(n-1), \dots, u(n-M)]^T$$

Bu güne kadar, ARX modelden başka çok sayıda model önerilmiştir. Bu konuda ayrıntılı bilgi için ilgili kaynaklar incelenebilir. ARX modeli basit yapısına karşın oldukça etkilidir ve bir çok uygulaması başarılı bir şekilde gerçekleştirilmiştir.

3.2.1.2. Doğrusal olmayan modeller

Doğrusal modeller , doğrusal ve doğrusal olmayan sistemlerin modellenmesinde oldukça iyi sonuçlar vermektedir. Fakat sistemlerin karmaşıklığı veya doğrusal olmayan yapının çok etkili olduğu uygulamalarda yetersiz kalmaktadırlar. Bunun nedeni doğrusal kabuller altında modelin büyük hata yapmasıdır. Daha önce bahsedilen doğrusal fark eşitliği doğrusal olmayan fark eşitliği olarak denklem 3.8'deki gibi yeniden düzenlenebilir(Chen 1990).

$$\hat{y}(n) = f(y(n-1), \dots, y(n-P), \hat{y}(n-1), \dots, \hat{y}(n-K), u(n), \dots, u(n-M)) \quad (3.8)$$

Bir başka gösterim şekli olarak da parametrelerinde işin içine katılması ile doğrusal olmayan ayrık zaman modeli denklem 3.9'daki gibi ifade edilebilir.

$$\hat{y}(n) = f(\mathbf{x}(n), \mathbf{w}) \quad (3.9)$$

Burada $f(\cdot)$; doğrusal olmayan esnek bir fonksiyondur. Denklem 3.8'deki doğrusal olmayan fark eşitliği NARMAX olarak bilinen doğrusal olmayan öz bağımlı kayan ortalamalı ve dış girişli modeli göstermektedir. Sırası ile modelin parçaları olan AR , MA , X , sistem çıkışı , model çıkışı ve harici giriş terimlerinin geçmiş değerleri ile oluşturulur. Bu bahsedilen yapıya karşılık genellikle NARX modeli basitliği nedeni

ile tercih edilmektedir(Ljung 1987). Bu modelde $x(n)$ vektörü sadece giriş ve çıkışın geçmiş değerlerinden oluşan giriş vektörüdür.

Bahsedilen her iki modelde de $f(.)$ fonksiyonu zamanla değişmeyen bir fonksiyondur. Sistem tanılama uygulamalarında model yapısı ve $f(.)$ fonksiyonunu önceden bilmek zordur. Bu yüzden $f(.)$ bir çok sistemi modelleyebilecek şekilde seçilmelidir. Genellikle doğrusal olmayan model denklem 3.10'daki gibi ifade edilir.

$$\hat{y}(n) = \sum_i a_i(x(n)), w_i \quad (3.10)$$

Burada , $a_i(.)$ doğrusal olmayan taban (basis) fonksiyon ve w_i ise model parametreleridir. Denklem deki sonuç model taban fonksiyon vektörünün, parametre vektörü ile doğrusal çarpımıdır. Temel fonksiyon olarak en önemlileri ridge fonksiyonu ve tensor çarpımı verilebilir. Doğrusal olmayan sistemlerin tanınması bölüm 7'de bahsedileceği gibi çok katmanlı yapay sinir ağlarına uygunluk göstermektedirler.

BÖLÜM 4. YAPAY SİNİR AĞLARI

Yapay sinir ağları ile ilgili çalışmalar, insan beyninin klasik bilgisayarlara göre farklı bir hesap yöntemi kullanmasının anlaşılması ile başlamıştır. Beyin işlevlerinin öğrenilmesi ile ilgili ilk çalışmalar Ramon Cagal (1911) ile başlamıştır. Cagal sinir hücrelerinin bütün beyni oluşturan yapısal parçalar olduğunu gösterdi. Sinir hücreleri mantık kapılarından daha yavaşırlar. Silikon yongada bir olay yaklaşık 10^{-9} s alırken bu beyinde 10^{-3} s olmaktadır. Fakat beyin içersinde tahminlere göre 10 milyon hücre ve bunlar arasında 60 trilyon bağlantı göz önüne alınırsa bu yavaşlığın sebebi anlaşılabilir. Bu karmaşık yapı ile mükemmel bir verimlilik sergileyebilmektedir.

Beyin, karmaşık , doğrusal olmayan ve paralel işlem yapan bir mekanizmadır. Beyin kesin hesaplamaları yapabilmek için sinir hücrelerini düzenlemektedir. Bu özelliği ile bilgisayarlardan daha hızlıdır. Örnek olarak bir manzara içinde nesneyi tanımak insan beyni için oldukça kolay bir işlemdir. Fakat bu iş bir bilgisayar için bazen çok uzun süre gerektirebilir.

Sinaplar hücreler arasında bilgi alışverişini sağlayan yapısal ve işlevsel bağlantılardır. En yaygın olanları ise kimyasal olanlardır. Sinaptik işlem anında hücreler arasındaki bağlantıda yayılan iletici madde salgılanır. Bu işlem önsinap olarak adlandırılan elektriksel dürtünün kimyasal işleve çevrilmesidir. Sinaplar hücreler arasında uyarıcı veya yasaklayıcı rol oynayabilirler.

Yapay sinir ağları ile insan beyninin kesin işlevleri yerine getirilebilmektedir. Yapay sinir ağı bahsedilen kesin bir işlevi yerine getirmek için oluşturulmuş bir mekanizma olarak düşünülebilir. Bu işlev genellikle bilgisayar simülasyonu veya elektronik cihazlarla yerine getirilir.

4.1. Yapay Sinir Ağı Kullanımının Üstünlükleri

YSA hesaplamadaki gücünü paralel işlem yapabilme ve öğrenme kabiliyetinden alır. Yapay sinir ağları öğrendikleri bilgileri genelleştirirler. Burada, genelleştirmeden kasıt

Eđitimde karřılařılmayan bir giriře yanıt verebilmesidir. Bu iki özellik ile YSA karmařık hesap iřlemlerinin basit giriř çıkıř iliřkilerine dđnüřtürölmesine izin verir (Haykin 1994).

Sinir hücreleri doğrusal olmayan bir yapıya sahiptir. Sonuç olarak YSA 'nın bu hücrelerin bađlantılarından oluřtuđu düşünölürse doğrusal olmayan bir yapı elde edilmiř olur. Doğrusal olmayan iřlevler ađ içerisinde dađıtılmaktadırlar. Gerçek zaman sistemlerinin bir çođu doğrusal deđildirler. Bu yüzden YSA 'ların kullanımı ile bu sistemlerin çözümleri rahatlıkla gerçekleştirilebilmektedir.

Eđiticili öğrenme olarak adlandırılan örnek bir öğrenme ,bir eğitim setinin ađ giriřin uygulanması ve bu giriře göre ađlıkların düzenlenmesi ile olur. Her bir örnek öğrenme, tek bir giriři ve uygun istenen deđerini göstermektedir. Ađın ađlıkları istenen deđerle giriř arasındaki farkı en aza indirecek řekilde ayarlanır. Eğitim, ađ sürekli hale ulařıncaya kadar devam ettirilir. Ađ, eğitim anında, probleme giriř çıkıř eřleřmesi oluřturur.

Yapay sinir ađı, ađlıklarını deđiřen ortama uyarlanabilecek řekilde ayarlar. Özel iřlevleri yerine getirmek için eğitimlenmiř ađ ortamdaki ufak deđiřimlere uygun cevap verebilecek řekilde ayarlanır. Durađan olmayan bir ortamda gerçek zaman deđiřimleri yapay sinir ađları tarafından izlenebilir. Ađ bu gibi ortamlarda çalıřırken kararlılık kriterleri göz önüne alınarak ne kadar uyarlanabilir bir sistem elde edersek o kadar iyi sonuçlar elde ederiz.

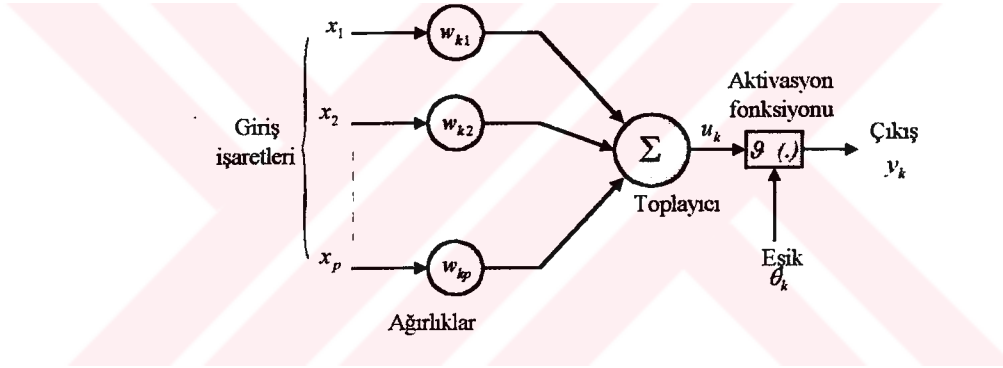
Yapay sinir ađı donanımları hızlı paralel iřlemciler ile belirli görevleri hızlıca yerine getirebilirler ve bu özellikleri VLSI teknolojisi ile üretime uygunluk göstermektedirler. VLSI teknolojisi ile birçok imge tanıma , kontrol ve sinyal iřleme uygulamaları gerçekleştirilebilmektedir.

4.2.Sinir Hücreleri Modelleri

Sinir hücreleri, yapay sinir ađlarının temel iřlevi olan bilgi iřleme ünitesidir. Bir sinir hücreleri için üç temel eleman tanımlayabiliriz.

- Model girişinde her biri ağırlıklarla belirtilen bir grup hat bulunmaktadır. Giriş işareti olan x_j k sinir hücresine bir ağırlıkla bağlıdır. Hücre çıkışı ağırlıklarla girişin doğrusal çarpımı ile ifade edilir. Ağırlıkların w_{kj} şeklinde ifade edilmesinde k bağlantı kurulan hücreyi j ise hangi ağırlıkla çalışıldığını göstermektedir.
- Modelde uygun ağırlıklarla çarpılan girişi toplamak için bir toplayıcı bulunmaktadır. Bahsedilen toplayıcı doğrusal birleştirici olarak anılmaktadır.
- Model çıkışında hücre çıkışını sınırlamak için bir aktivasyon fonksiyonu kullanılmaktadır. Bunun amacı hücre çıkışını belirli sınırlar içine sokmaktır.

Şekil 4.1 de görülen hücre modeli harici eşik değerine (θ_k) sahiptir. Bunun görevi aktivasyon fonksiyonunun giriş değerini azaltmaktır. Benzer şekilde aktivasyon fonksiyonunun girişi arttırılabilir.



Şekil 4.1 Sinir hücresinin doğrusal olmayan modeli

Matematiksel olarak bir sinir hücresi denklem 4.1 ve 4.2'deki gibi tanımlanabilir (Haykin 1994).

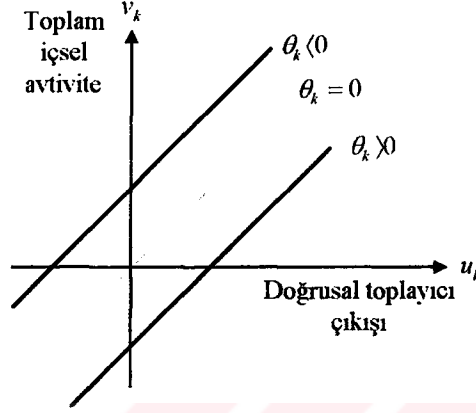
$$u_k = \sum_{j=1}^p w_{kj} x_j \quad (4.1)$$

$$y_k = \varphi(u_k - \theta_k) \quad (4.2)$$

Burada, $x_{k1}, x_{k2}, \dots, x_{kj}$ giriş işaretleri ve $w_{k1}, w_{k2}, \dots, w_{kj}$ ise sinir hücresi k' ya bağlı olan ağırlıklardır. Doğrusal toplayıcının çıkışı u_k , θ_k eşik değeri ve $\varphi(.)$ ise aktivasyon fonksiyonudur. Eşik değerinin hücre çıkışına uygulanması çıkışa, denklem 4.3' deki ayar dönüşümünün uygulanmasını sağlar.

$$v_k = u_k - \theta_k \quad (4.3)$$

Burada , v_k aktivasyon fonksiyonundan önceki ağırlık iç aktivitesidir. Eşik değerinin pozitif veya negatif oluşunun, toplayıcı çıkışı ve toplam içsel aktiviteye etkisi Şekil 4.2 deki gibi görülebilir.



Şekil 4.2 Eşik değerinin toplayıcı çıkışına ve toplam aktivasyona etkisi

Eşik değeri k sinir hücresinin harici bir parametresidir. Denklem 4.2’de eşik değerinin etkisi denklem 4.4’deki gibi gösterilebilir.

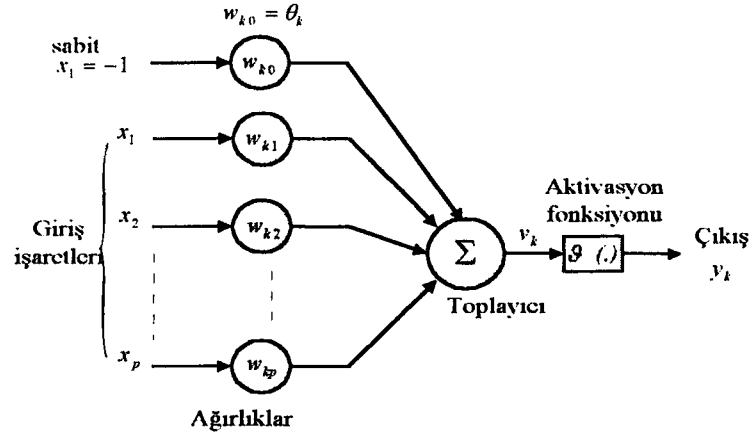
$$v_k = \sum_{j=0}^p w_{kj} x_j \quad (4.4)$$

$$y_k = \varphi(v_k) \quad (4.5)$$

Denklem 4.4 ve 4.5’deki eşik değeri gösterimi Şekil 4.3’de gösterilmiştir. Şekilden görülebileceği gibi eşik değeri sinir hücresi için sabit değerli bir giriştir ve diğer girişler gibi eşik değeri de hücreye bir ağırlık ile bağlanmaktadır.

4.3. Aktivasyon Fonksiyonları

$\varphi(\cdot)$ ile gösterilen aktivasyon fonksiyonu girişin aktivasyon seviyesi cinsinden hücre çıkışını tanımlar. Üç çeşit temel aktivasyon fonksiyonu mevcuttur (Freeman and David 1992).



Şekil 4.3 Eşik değerinin hücre modeli üzerinde değişik gösterimi

4.3.1.Eşik fonksiyonu

Bu aktivasyon fonksiyonu çeşidi şekil 4.4.a'da görüldüğü gibidir ve denklem 4.6'daki gibi tanımlanır.

$$\mathcal{G}(v) = \begin{cases} 1 & v \geq 0 \\ 0 & v < 0 \end{cases} \quad (4.6)$$

Bu ifade ile sinir hücresinin çıkışı eşik fonksiyonu olarak tanımlanmıştır. Burada, v sinir hücresinin içsel aktivitesidir. Bu gibi sinir hücreleri McCulloch Pitts model olarak bilinir.

4.3.2.Parçalı doğrusal fonksiyon

Parçalı doğrusal fonksiyon şekil 4.4.b'de verilmiş olup denklem 4.7'deki gibi ifade edilir.

$$\mathcal{G}(\cdot) = \begin{cases} 1 & v \geq 1/2 \\ v & 1/2 > v > -1/2 \\ 0 & v \leq -1/2 \end{cases} \quad (4.7)$$

Burada doğrusal bölgedeki eğim faktörü birim olarak alınmıştır. Bu çeşit aktivasyon fonksiyonları doğrusal olmayan yükseltece benzetilebilir.

4.3.3.Sigmoid fonksiyonu

Yapay sinir ağlarında kullanılan en yaygın aktivasyon fonksiyonudur. Asimtotik özellik sergileyen bir yapıdadır. Sigmoid fonksiyonuna bir örnek denklem 4.8'deki gibi verilebilir.

$$g(v) = \frac{1}{1 + e^{-av}} \quad (4.8)$$

Burada , a eğim parametresidir. Sigmoid fonksiyonu çeşitli eğim değerleri için Şekil 4.4.c 'de çizilmiştir. Limit değerlerde eğim parametresi sonsuza yaklaşır ve fonksiyon genlik fonksiyonuna dönüşür.

Daha önce bahsedilen aktivasyon fonksiyonları 0-1 aralığında değişmektedir. Bazen sigmoid fonksiyonu orijine göre simetrik olması istenen durumlarda -1 +1 aralığında alınır. Bu durum denklem 4.9'da gösterilmiştir.

$$g(v) = \begin{cases} 1 & v > 0 \\ 0 & v = 0 \\ -1 & v < 0 \end{cases} \quad (4.9)$$

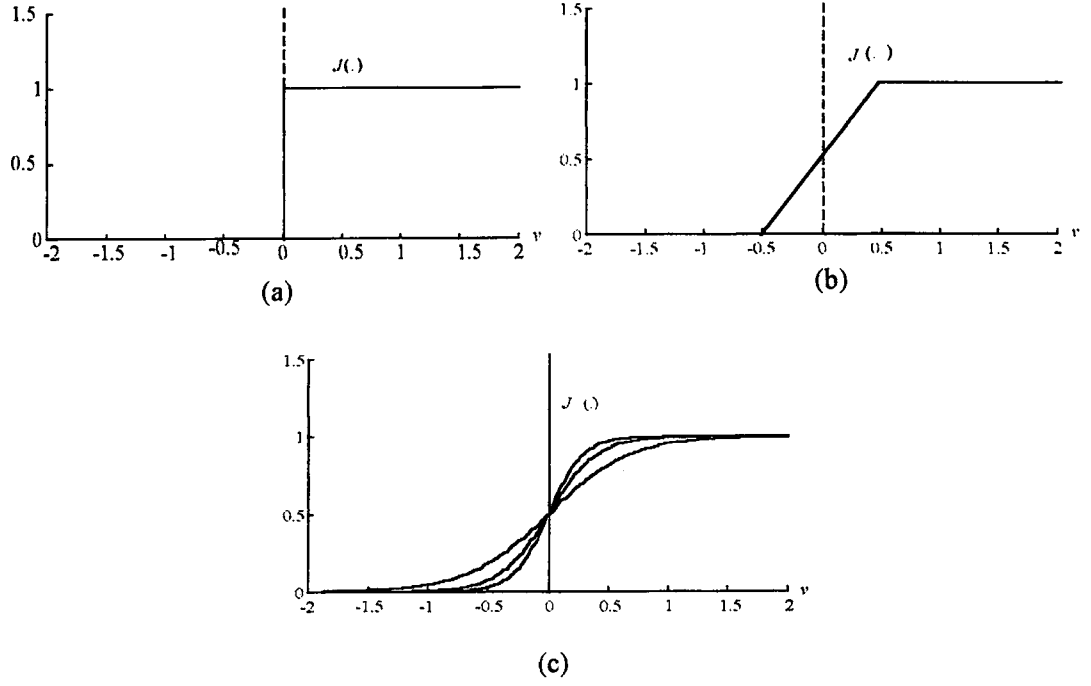
denklem 4.9'daki fonksiyon genellikle signum fonksiyonu olarak adlandırılır.

4.4.Yapay Sinir Ağı Yapıları

Yapay sinir ağında hücrelerin yapılandırılması , ağı eğitmek için kullanılan öğrenme algoritması ile bağlantılıdır. Bu yüzden yapay sinir ağları tasarımında öğrenme algoritmaları göz önüne alınmalıdır. Bu bölümde, temel ağ mimarilerinden bahsedilecektir. En genel anlamda dört değişik ağ mimarisinden bahsedebiliriz.

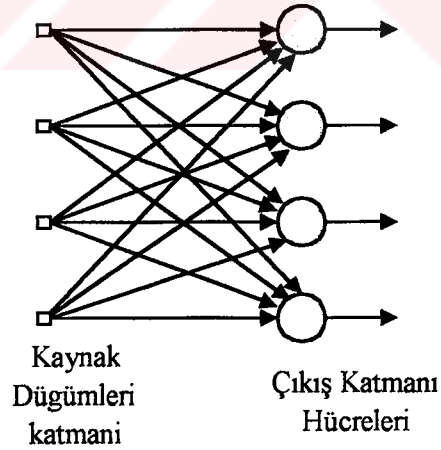
4.4.1.Tek katmanlı ileri yönlü ağlar

Katmanlı YSA sinir hücrelerinin katmanlar halinde düzenlenmesinden oluşur. Bu çeşit ağlar ileri yönlü yapıya sahiptirler. Yapısı şekil 4.5'deki gibidir. Şekilden görüleceği gibi ağ sadece giriş kaynak düğümlerinden ve çıkış hücrelerinden oluşmaktadır. Tek katmanlı ağlara örnek olarak Doğrusal İlişkisel Bellekli Ağ verilebilir. Bu ağın uygulanmasında giriş vektörü ile çıkış arasında ağırlıklarla bir bellek ilişkisi kurulmaktadır.



Şekil 4.4 Genel aktivasyon fonksiyonları

- a) Eşik aktivasyon fonksiyonu
- b) Parçalı doğrusal aktivasyon fonksiyonu
- c) Çeşitli eğim parametreleri için sigmoid fonksiyonu



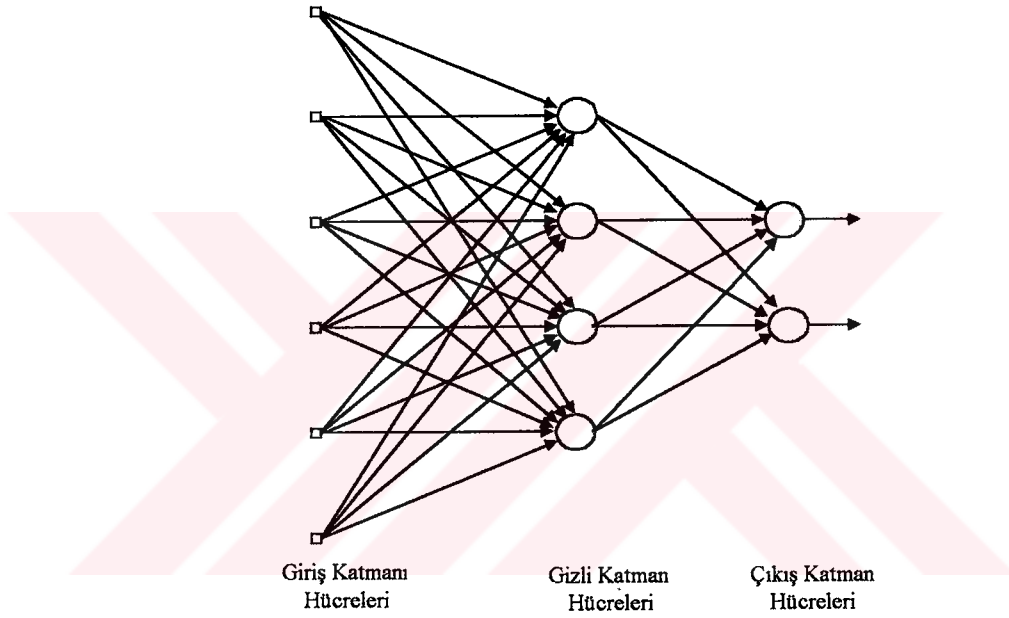
Şekil 4.5 Tek katmanlı ileri yönlü ağ yapısı

4.4.2. Çok katmanlı ileri yönlü ağlar

İleri yönlü ağların ikinci çeşidi olan çok katmanlı ağlar iki veya daha fazla gizli katmandan oluşur. Gizli katmanlar giriş ile çıkış arasında yer almaktadırlar. Gizli

katmanların ağı eklenmesi yüksek dereceli istatistiksel bilgilerin kullanılmasına izin vermektedir. Gizli katmanlar arasındaki ağırlık bağlantılarının kullanımı Churchland ve Sejnowski(1992)'ye göre ağ global bir yapı kazanmaktadır (Khanna 1989).

Giriş düğümleri uygun girişleri ağırlık bağlantıları üzerinden gizli katmana bağlamaya yarar. Sırası ile bir önceki katman sonraki katman için giriş olmaktadır. Ağın en çıkışında girişi temsil eden ve tüm katmanlardan geçmiş çıkış bilgisi elde edilir. Bir tane gizli katman içeren çok katmanlı bir ağ modeli şekil 4.6 deki gibidir.



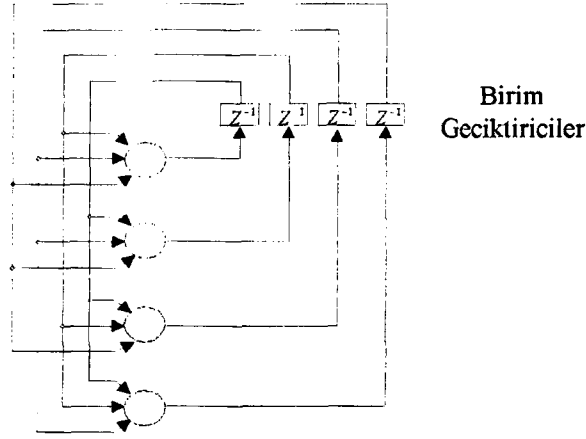
Şekil 4.6 Bir gizli katmana sahip ve tam bağlantılı ileri yönlü çok katmanlı ağ yapısı

Şekildeki çok katmanlı ağ altı giriş düğümüne ile dört gizli katmana ve iki tane de çıkış katmanı sinir hücresine sahiptir. Şekilden görülebileceği gibi her bir düğüm veya hücre kendinden sonra gelen tüm hücrelere bağlantı yapmaktadır. Bu tür bağlantıya sahip ağlara tam bağlı ağlar denir. Bazen bu bağlantıların tam olması gereksizdir. Bu durumda ağ parçalı bağlı olarak bilinir.

4.4.3.Geri dönüşümlü ağlar

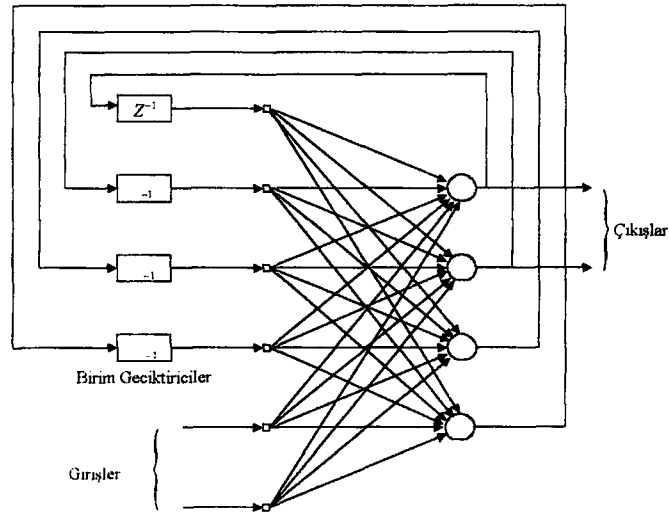
Geri dönüşümlü ağlar ileri yönlü ağlardan en az bir geri yönlü göze sahip oluşları ile ayrılırlar. Örnek olarak bir geri dönüşümlü ağ bir katmandan ve bu katman çıkışının

giriş geri beslendiği bir yapıda olabilir. Böyle bağlantıya sahip bir yapı şekil 4.7’de ki gibidir. Şekilden anlaşılacağı gibi öz geri besleme kullanılmamıştır. Öz geri besleme



Şekil 4.7 Tek katmanlı geri dönüşümlü ağ yapısı

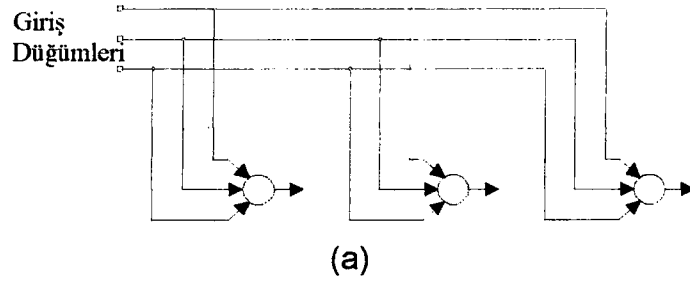
hücre çıkışının yine kendi girişine geri beslenmesidir. Şekil 4.7’deki ağda gizli katman bulunmamaktadır. Şekil 4.8’da ise gizli katman içeren geri dönüşümlü ağ modeli görülmektedir. Gizli katman çıkışları birim geciktiriciler üzerinden girişe geri beslenmektedir. Geri beslemenin öğrenme üzerinde etkisi dinamik ağlar bölümünde detaylı olarak ele alınacaktır.



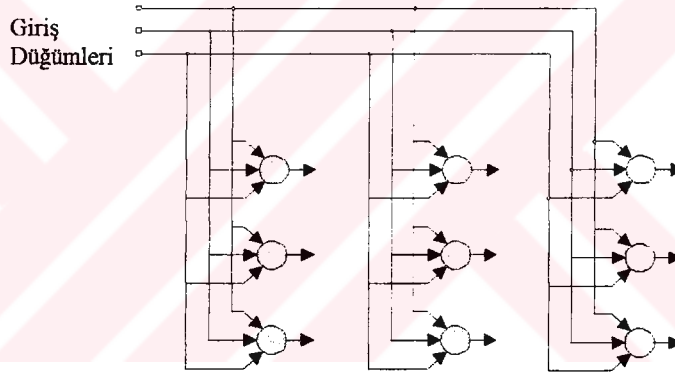
Şekil 4.8 Gizli katmanlı olan Geri dönüşümlü ağ yapısı

4.4.4.Kafes yapılı ağlar

Kafes yapısı , bir veya birden fazla boyutlu sinir hücresinden oluşan diziler ile elde edilir. Diziye giriş sağlayan giriş düğümlerinin sayısı dizi boyutuna uygunluk sağlamaktadır. Kafesin boyutu , ağ grafının bulunduğu uzayda boyutların sayısı olarak ifade edilir. Şekil 4.9.a’da bir boyutlu , b’ de ise iki boyutlu ileri yönlü kafes ağ yapısı görülmekte olup tüm giriş düğümleri her bir hücreye giriş yapmaktadır.



(a)



(b)

Şekil 4.9 Kafes yapılı ağlar

a) Bir boyutlu kafes yapı

b) İki boyutlu kafes yapı

4.5.Yapay Sinir Ağlarında Öğrenme ve Öğrenme Çeşitleri

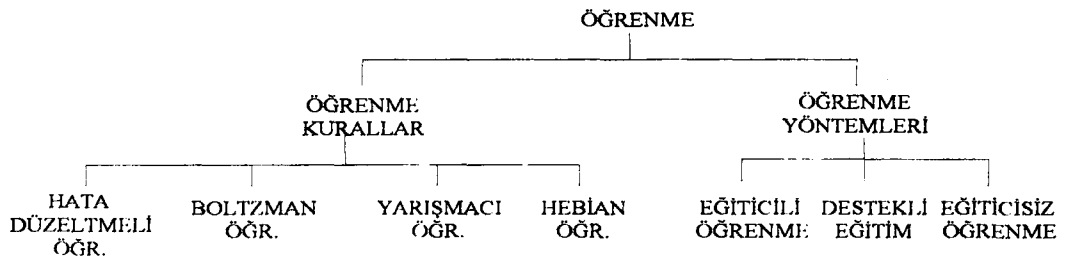
YSA' ların bir çok özellikleri arasında en önemlisi çevresini öğrenebilmesi ve öğrenme aşamasında performansını arttırabilmesidir. Performanstaki artış belirtilen ölçüler içinde zamanla oluşur. YSA ağırlıklara uygulanan iteratif uyarılama ile çevresini öğrenebilmektedir. Her bir iterasyondan sonra ağ çevresi ile ilgili biraz daha bilgi sahibi olmaktadır. Mendel ve McClaren 1970’de YSA kavramı olarak öğrenmeyi aşağıdaki gibi tanımlanmışlardır.

“Öğrenme , ağıın içinde bulunduđu ortam tarafından, devam eden bir süreçte bağımsız ağı parametrelerinin ayarlanması işlemidir.”

Öğrenmenin çeşidi ağıın hangi parametrelerinin deđiştirileceđine göre belirlenir. Öğrenme problemini çözmek için tanımlanmış kuralların tümüne öğrenme algoritması denir. YSA'nın tasarımı için tek bir öğrenme algoritması yoktur. Bu güne kadar çeşitli algoritmalar geliştirilmiştir. Bu algoritmaların her birinin kendine özgü üstünlükleri ve eksiklikleri vardır. Öğrenme algoritmaları ağırlık deđişiminin yapılış şekline göre birbirlerinden ayrılırlar. Yapay sinir ağılarında öğrenme Şekil 4.10'daki gibi sınıflandırılabilir (Haykin 1994).

Dört temel öğrenme kuralından biri olan hata düzeltmeli öğrenme Optimal filtrelemenin bir uzantısı olarak geliştirilmiştir. Hebbian ve Yarışmacı öğrenme ise nörobiyolojik düşünceden kaynaklanmaktadır. Boltzmann öğrenme diğerlerinden oldukça farklı olarak termodinamikten esinlenilerek geliştirilmiştir. Şekil 4.10'dan görüleceđi gibi üç ayrı çeşit öğrenme yöntemi vardır. İsminden anlaşılacağı gibi eğitici öğrenmede bir eğiticiye ihtiyaç duyulmaktadır. Destekli öğrenmede bazı kriterler kullanılarak deneme-hata ilişkisi incelenir. Eğitici öğrenmede bir eğiticiye ihtiyaç yoktur. Ağırlık deđişimleri içsel olarak gerçekleştirilir.

Şekilden görülen öğrenme yöntem ve kurallarının tez çalışması ile ilişkili olan çok katmanlı ağılarda eğitici öğrenme ve hata düzeltmeli öğrenme bölüm 5'de detaylı olarak ele alınmıştır.



Şekil 4.1 Yapay sinir ağılarında öğrenmenin sınıflanması

BÖLÜM 5. EŞLEŞTİRMELİ AĞLAR VE ÇOK KATMANLI

ALGILAYICILAR

Eşleştirmeli ağlar tarafından belirtilen problem, eşleşmeyi ifade eden $f: A \subset \mathbb{R}^n \rightarrow \mathbb{R}^m$ yakınsama işleminin gerçekleştirilmesidir. Burada, n boyutlu bir uzayın alt elemanı olan A kümesinin m boyutlu bir $f(A)$ kümesine yakınsaması ifade edilmektedir (Narendra 1990). Bu işlem giriş çıkış için alınan eşlerin $((x_1, y_1), (x_2, y_2), \dots, (x_k, y_k))$ eğitimi ile gerçekleştirilir.

Özellik ve örnek tabanlı olmak üzere iki temel eşleştirmeli ağ vardır. Bir özellik ağı, fonksiyonel terimlerle ifade edilen giriş çıkış ilişkisini oluşturur. Bu fonksiyonel yapı, gerçekleştirilmesi gereken eşleştirmeye göre, eğitim kuralı kullanarak ağ ağırlıklarının uyarlanmasıyla elde edilir. Özellik tabanlı ağlara en iyi örnek geri yansıtmalı ağıdır.

Bir örnek tabanlı ağ, istatistiksel olarak ağın davranışını belirleyen özel giriş çıkış örnek kümelerinin $((x_1, v_1), (x_2, v_2), \dots, (x_L, v_L))$ yaratılması ile çalışır. Ağ yeni bilinmeyen giriş için bir karşılaştırma yaparak yeni giriş için bir tahminde bulunur. Öz-organizeli eşlem ağı (self-organizing map network) örnek tabanlı ağlar için örnek olarak verilebilir.

Bu bölümde tez çalışmasında kullanılan özellik tabanlı ağlardan olan çok katmanlı algılayıcılar (veya Ağ) ele alınacaktır. Çok katmanlı algılayıcıları eğitmek için kullanılan en popüler metod olan geri yansıtmalı öğrenme algoritmasından bahsedilecektir.

5.1. Ölçü Fonksiyonu ve Yakınsama

YSA girişini x vektörü ve $y(x, w)$ ise çıkışını gösterebilir. Burada w , YSA'nın ağırlık vektörüdür. Ağın yakınsamasının doğruluğunu tespit etmek için ağın gerçek çıkışı ile istenen değer karşılaştırılır. Ayrıca, x_k ve y_k , k 'inci deneme için örnekler olsun.

Burada $y_k=f(x_k)$ 'dir ve yakınsamanın doğruluğu hakkında ölçü olarak denklem 5.1 kullanılabilir(Chen 1990).

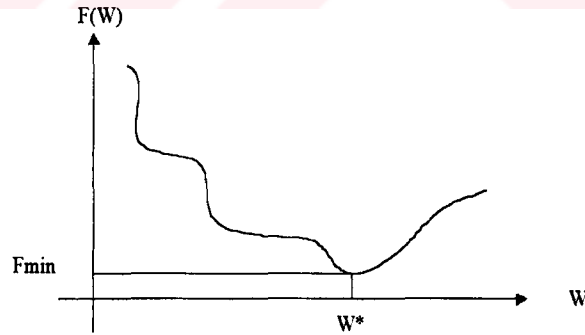
$$F_k(x_k, w) = |f(d_k - y(x_k, w))|^2 \quad (5.1)$$

Burada F_k ; yakınsama hatasının karesidir. Test anında ağırlık vektörünün sabit olduğuna dikkat edilmelidir. Yakınsama için bir ifade elde edildikten sonra ağ için ortalama karesel hata $F(w)$ seçilen x_k girişinin limitinin olması koşulu ile denklem 5.2'deki gibi tanımlanır.

$$F(w) = \lim_{N \rightarrow \infty} \sum_{k=1}^N F_k(x_k, w) \quad (5.2)$$

Ortalama karesel hatanın hata için bir ölçü kullanılması ağırlık değişimlerini karesel olarak değiştirilmesine neden olur. Yani büyük hatada daha büyük değişim küçük hatada ise daha küçük ağırlık değişimi sağlar.

Yukarıda bahsedildiği gibi ortalama karesel hata ağırlıkların bir fonksiyonudur. Her ağırlık değişiminde farklı ortalama karesel hata elde edilir. Bu yüzden ortalama karesel hata ağırlıklar ile değişen bir yüzey olarak düşünülür. Bu yüzey hata yüzeyi olarak adlandırılır. Şekil 5.1 tipik bir hata yüzeyini göstermektedir.



Şekil 5.1 Hata yüzeyinin tipik şekli.

5.2. Çok Katmanlı Algılayıcılar

Çok katmanlı algılayıcılar ileri yönlü YSA modelleri içinde önemli yere sahiptirler. Ağ, girişinde düğümlerden oluşan bir giriş katmanı, bir veya birden fazla gizli katmanı ve bir de çıkış katmanından oluşur. Giriş işareti giriş düğümlerinden ileriye doğru ağ

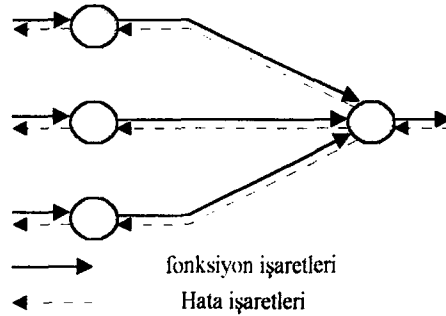
içinde yansıtılır. Bu tür ağlar genellikle çok katmanlı algılayıcılar veya ağlar olarak isimlendirilir.(Multilayer Perceptron) Çok katmanlı algılayıcılar birçok karmaşık probleme çözüm getirebilmektedirler. Eğitimleri , eğiticili eğitim sınıfına girer ve Hata Geri Yansıtma Algoritması ile eğitimleri gerçekleştirilirler. Bu eğitim Hata Düzeltmeli Öğrenme ile benzerlik göstermektedir.

Geri Yansıtımlı Öğrenmede ağ içinde , ileri ve geri olmak üzere iki geçiş vardır. İleri geçişte giriş, tüm katmanlardan geçerek çıkışa ulaşır. Çıkışta ağın gerçek cevabı elde edilir. Ağın çıkışında istenilen değerle gerçek değer arasındaki fark alınarak hata olarak elde edilir. Geri yansıtma anında bu hata değerine göre ileri yönde geçişte sabit olan ağ ağırlıkları değiştirilir. Hata Geri Yansıtımlı Öğrenme ismini buradan almaktadır. Ağırlık değişimleri çıkış değeri ile istenen değer arasındaki farkı en aza indirecek şekilde sürdürülür.

5.2.1. Çok katmanlı ağların özellikleri:

Çok katmanlı ağlar tüm sinir hücrelerinin çıkışlarında doğrusal olmayan , türevi alınabilen bir aktivasyon fonksiyonu içerirler. Bu tür ağlarda genellikle daha önce bahsedilen sigmoid fonksiyonu aktivasyon fonksiyonu olarak kullanılmaktadır. Çok katmanlı bir ağ bir veya birden fazla gizli katman içerir ve ağın hücreleri arasında ağırlıklardan oluşan birçok ağırlık bağlantısı vardır.

Şekil 5.2 çok katmanlı algılayıcının küçük bir kısmını göstermektedir. Bu şekilde , fonksiyon ve hata işaretleri olarak iki çeşit işaretten bahsedilebilir.



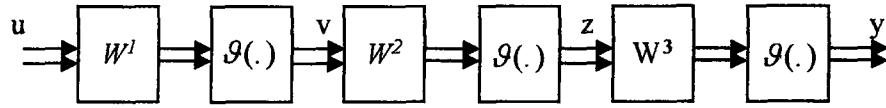
Şekil 5.2 İki temel işaretin çok katmanlı bir ağ parçası üzerinde gösterilişi.

Fonksiyon işaretleri ağın girişine uygulanan giriş işaretidir. Bu işaret daha sonra ağın tüm katmanlarından geçerek çıkış katmanında bir işaret üretir. Bu işarete fonksiyon işaretleri

denmesinin nedeni çıkışın, giriş ve ağırlıkların bir fonksiyonu olmasıdır. Hata işareti ise ağırlık çıkışında elde edilen ve geri yönlü yansıtılan işarettir.

Çok katmanlı algılayıcı , her bir hücre çıkışında görülen fonksiyon işaretine göre hücrelerin sahip olduğu doğrusal olmayan sürekli fonksiyon çıkışını ve geri yansıtma için gerekli olan ani gradiyent değerini hesaplar.

Çok katmanlı tipik bir ağırlık blok yapısı şekil 5.3'deki gibidir. Şekilden görüleceği gibi ağ W^1 , W^2 , W^3 olmak üzere üç ağırlık matrisine ve her ağırlık matrisinin çıkışında doğrusal olmayan $\mathcal{G}(\cdot)$ operatörüne sahiptir.



Şekil 5.3 Üç katmanlı ağırlık blok diyagramı olarak gösterilişi.

Blok diyagramda görülen her bir katman denklem 5.2 deki gibi ifade edilebilir.

$$N_i[u] = \mathcal{G}[W^i u] \quad (5.3)$$

Burada , N_i tek katmanı göstermektedir. Çok katmanlı ağırlık giriş çıkış ilişkisi denklem 5.4 deki gibidir.

$$y = \mathcal{G}[W^3 \mathcal{G}[W^2 \mathcal{G}[W^1 u]]] = N_1 N_2 N_3 [u] \quad (5.4)$$

5.3.Geri Yansıtma Algoritması

Geri Yansıtma öğrenme Şekil 5.4 gösterilmiş olup hem ileri geçiş ve hem de geri yönlü işaret akışı görülebilmektedir(Haykin 1994).

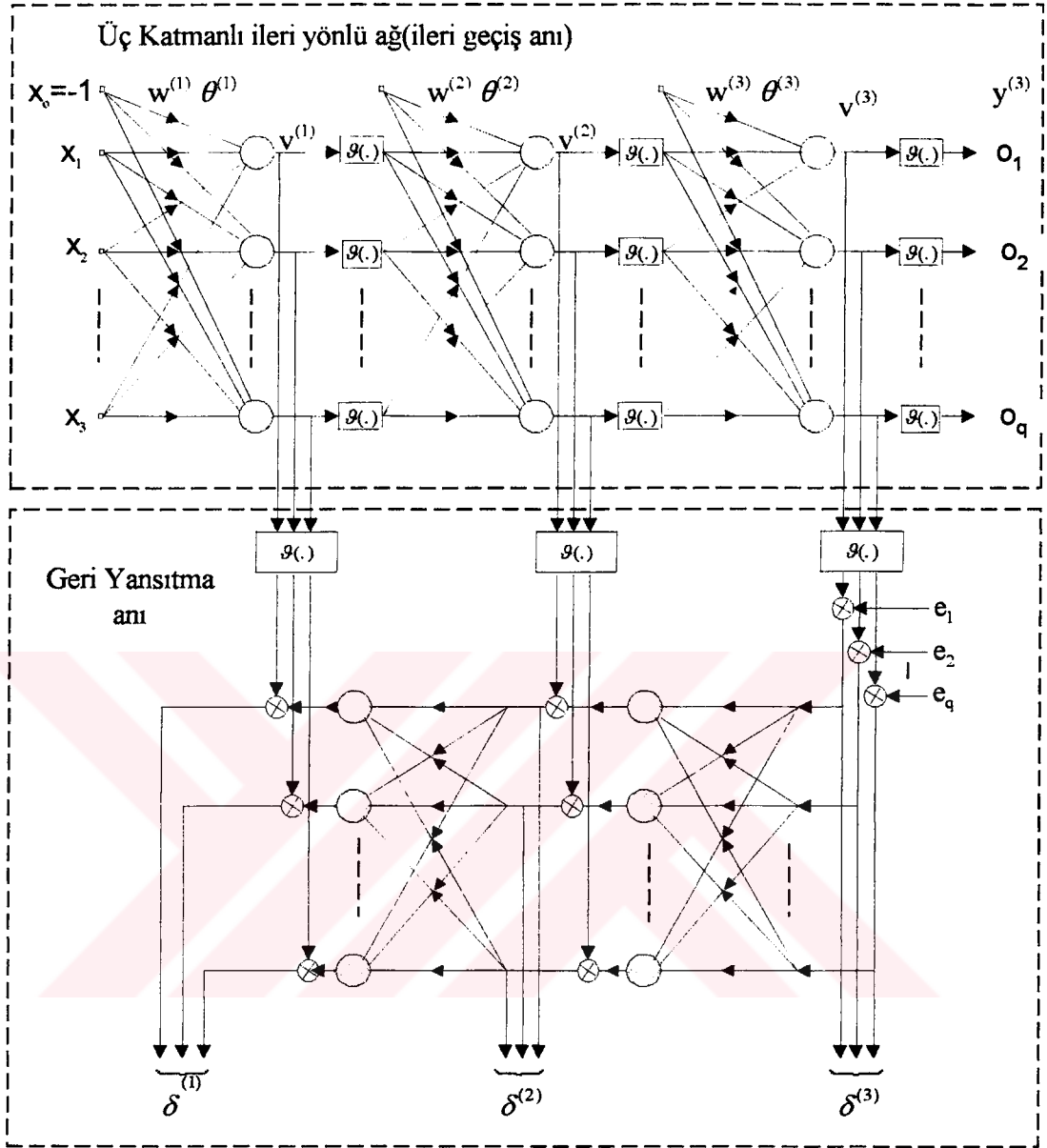
Şekilde kullanılan notasyon aşağıdaki gibidir.

$w^{(l)}$ / katmandaki ağırlıklar

$\theta^{(l)}$ / katmandaki eşik değeri

$v^{(l)}$ / katmandaki hücrelerin aktivasyon değeri

$y^{(l)}$ / katmandaki hücrelerin fonksiyon işareti



Şekil 5.5 Geri yansıtma algoritmasının diyagram olarak gösterilişi

Şekilde ağın derinliği olan $l=3$ olarak alınmıştır. Şekil 5.5'in alt kısmı geri yansıtmayı açıklamaktadır. Bu kısımda yerel gradiyent elde edilmektedir ve kullanılan notasyonlar ise aşağıdaki gibi tanımlanmışlardır.

$\delta^{(l)}$ l katmandaki hücrelerin gradiyent vektörü.

e ise e_1, e_2, \dots, e_q elemanlarından oluşan hata vektörüdür.

Geri yansıtımlı öğrenmede bahsedildiği gibi iki geçiş vardır. İleri yön geçişte ağırlıklar sabit olacak şekilde giriş vektörü çıkışa doğru her bir katmandan geçerek yansıtılır.

Geri yansımada ise her bir katman için gradiyent vektörü hesaplanır. Bulunan gradiyent ifadesi ile ağırlık değişimleri arasında önemli bir ilişki vardır.

Eğitim çifti , giriş bilgisi $x(n)$ ve bu giriş için istenen değer olan $d(n)$ [$x(n)$, $d(n)$] ile gösterilsin. $x(n)$ giriş bilgisi ağın giriş düğümlerinden ağa doğru uygulanır çıkışta ise daha önceden belirlenmiş istenen değer bulunmaktadır. Giriş uygulandıktan sonra her katmanın tüm hücreleri için fonksiyon işareti yada aktivasyon hesaplanır. l 'inci katmandaki j hücresinin aktivasyonu olan $v_j^\ell(n)$ denklem 5.5'deki gibi hesaplanır.

$$v_j^\ell(n) = \sum_{i=0}^p w_{ji}^\ell(n) \cdot y_i^{\ell-1}(n) \quad (5.5)$$

Burada $y_i^{\ell-1}(n)$ bir önceki katmanın i 'inci hücresinden gelen fonksiyon işaretidir. $w_{ji}^\ell(n)$ j hücresine bir önceki katmandan bağlı olan i 'inci hücreden gelen ağırlık bağlantısıdır. $i=0$ için $y_0^{\ell-1}(n) = -1$ ve $w_{j0}^\ell(n) = \theta_j^\ell(n)$ olarak alınır . Burada , $\theta_j^\ell(n)$ eşik değerini göstermektedir. Her katmandaki j 'inci hücre çıkışında doğrusal olmayan sigmoid aktivasyon kullanıldığı kabul edilirse her bir hücrenin çıkışı denklem 5.6'daki gibi bulunur.

$$y_j^\ell(n) = \frac{1}{1 + e^{(-v_j^\ell(n))}} \quad (5.6)$$

Eğer çalışılan katman bir çıkış katmanı ise hücre çıkışı denklem 5.7'deki gibi ifade edilir.

$$y_j^\ell(n) = o_j(n) \quad (5.7)$$

Çıkış katmanının tüm hücreleri için $d_j(n)$ istenen çıkış değerinin bilindiği düşünülürse hata değeri denklem 5.8'deki gibi hesaplanabilir.

$$e_j(n) = d_j(n) - o_j(n) \quad (5.8)$$

Geri yansıtma algoritmasının geri yönlü işleyişinin amacı her katman için yerel gradiyent ifadesinin elde edilmesidir. Gradiyent vektörü, çıkıştaki katmandan geriye

doğru tüm katmanlar için hesaplanır. Gradyent ifadesinin bulunması hangi katman ile çalışıldığına bağlıdır. Çıkış katmanı için gradyent ifadesi denklem 5.9'daki gibi hesaplanır.

$$\delta_j^\ell(n) = e_j^\ell(n) o_j(n) [1 - o_j(n)] \quad (5.9)$$

Bir gizli katman için gradyent ifadesi çıkış için olanından daha farklıdır ve denklem 5.10'daki gibidir.

$$\delta_j^\ell(n) = y_j^\ell(n) [1 - y_j^\ell(n)] \sum_k \delta_j^{\ell+1}(n) w_{kj}(n) \quad (5.10)$$

Gradyent ifadeleri bulunduktan sonra l katmanındaki ağırlıklar genelleştirilmiş delta kuralına göre denklem 5.11'deki gibi elde edilebilir.

$$w_{ji}(n+1) = w_{ji}(n) + \alpha [w_{ji}(n) - w_{ji}(n-1)] + \eta \delta_j^\ell(n) y_j^{\ell-1}(n) \quad (5.11)$$

Burada , η öğrenme oranı α ise momentum sabitidir. Bu sabitler istenen ortalama karesel hata değerine kaç denemede ulaşılacağı ile yakından ilişkilidir. Bu parametreler genellikle uygulama anında deneysel olarak elde edilebilirler.

5.4. Öğrenme Modları

Geri yansıtımlı öğrenme algoritmasının eğitimi, bir eğitim setinin ağ içerisinde defalarca işletilmesi ile sonuçlanır. Eğitim setinin tamamının ağ içerisinde bir defa geçirilmesine deneme (epoch) denir. Denemelerin sayısı istenen ortalama karesel hataya yaklaşımla belirlenir. İstenen en küçük değer elde edildiğinde eğitim son bulur. Ağırlık değişimleri ağ çıkışında elde edilen hataya göre gerçekleştirilir. Ağırlık değişimi için anlık ve grup olmak üzere iki farklı mod bulunmaktadır (Hertz 1991).

5.4.1. Anlık öğrenme modu

Bu eğitim modunda eğitim setinin her bir örneğine karşılık ağırlık değişimi hesaplanmaktadır. N eğitim örneği içeren eğitim seti ele alındığında $([x(1),y(1),\dots,x(N),y(N)])$, ilk eğitim çifti $x(1),y(1)$ 'den son eğitim çifti

$x(N),y(N)$ ' e kadar tüm çiftler için bir hata değişimi hesaplanır. $\Delta w_{ji}(n)$ ağırlık değişimini göstermektedir. Tüm eğitim örnekleri için ağırlık değişimi hesaplandıktan sonra net ağırlık değişimi ortalama olarak denklem 5.12' de görüldüğü gibi elde edilir.

$$\Delta \hat{w}_{ji}(n) = \frac{1}{N} \sum_{n=1}^N \Delta w_{ji}(n) \quad (5.12)$$

5.4.2. Grup öğrenme modu

Grup eğitim modu , geri yansıtmalı öğrenmede tüm eğitim setinin ağdan geçirilmesi sonucunda elde edilen ağırlık değişimi ile gerçekleşir. Ağ için maliyet fonksiyonu grup modda denklem 5.13 ile ifade edilir.

$$\varepsilon_{av} = \frac{1}{2N} \sum_{n=1}^N \sum_j e_j(n)^2 \quad (5.13)$$

Burada, $e_j(n)$ j çıkış hücresinde oluşan hatadır. Tüm çıkışlardaki hata eşitliğin iç tarafında gösterildiği gibi toplanır. Eşitliğin en dış tarafındaki toplam ile tüm eğitim setinin çıkışlarda meydana getirdiği hataların toplamı ifade edilmektedir. Net ağırlık değişimi denklem 5.14'de görüldüğü gibi hesaplanır.

$$\Delta w_{ji}(n) = \frac{\partial \varepsilon_{av}}{\partial w_{ji}} \quad (5.13)$$

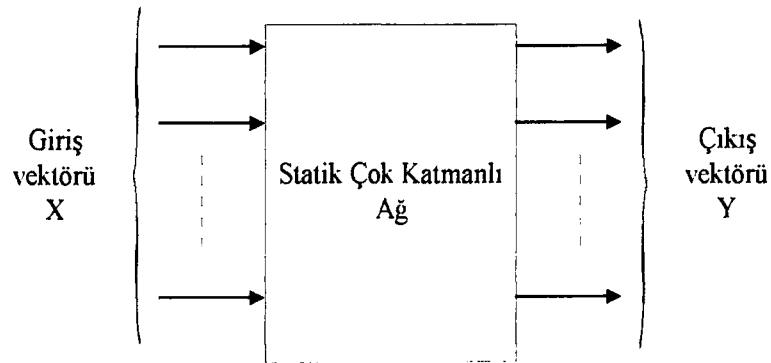
İki öğrenme modu oldukça farklıdır. Anlık öğrenmedeki ağırlık değişimi grup öğrenmede elde edileninkinin kestirimidir. Gerçek zaman öğrenme sistemlerinde anlık öğrenme tercih edilmektedir. Bunun nedeni hafıza ihtiyacının az oluşudur. Hangi öğrenme modunun kullanılacağı eldeki probleme göre farklılık göstermektedir.

BÖLÜM 6. DİNAMİK YAPILI YAPAY SINİR AĞLARI

Geri besleme hatları ile donatılmış yapay sinir ağları ileri yönlü ağlara alternatif olarak geliştirilmişlerdir. Geri beslemeli yapay sinir ağları doğrusal olmayan zaman serilerinin analizi veya doğrusal olmayan sistem tanılamada etkin bir şekilde kullanılabilmektedirler. Geri beslemeli ağlarda ağın bir önceki çıkış bilgisi ağ içerisine geriye yollar. Bundan dolayı karmaşık bir rekürsif yapıya sahiptirler. Gerçek sistemlerin davranışları zamanla değişken bir durum sergilemektedirler. Bu yüzden sistemlerin tanınmasında gerçek zaman değişimlerinin hesaba katılması gereklidir. Dinamik yapay sinir ağları zamanla değişen sistemlerin veya işaretlerin modellenmesinde uygunluk göstermektedirler. Bu bölümde dinamik yapay sinir ağlarının tanımı ve gerekliliği ortaya konduktan sonra gerçek zaman geri dönüşümlü ağından bahsedilecektir.

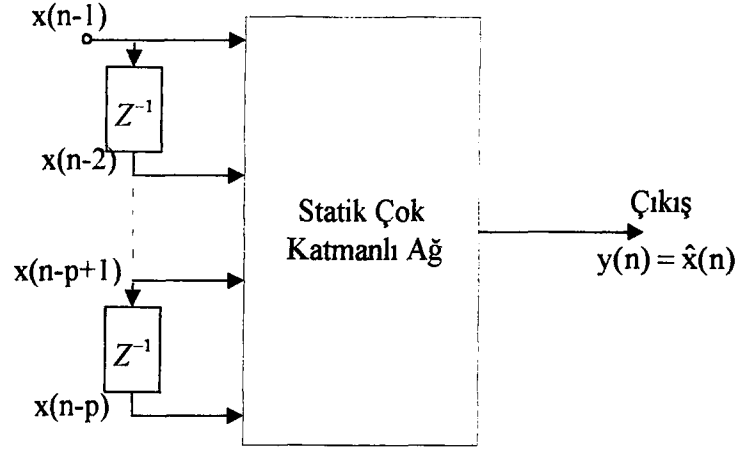
6.1. Dinamik Yapay Sinir Ağı Tanımı

Bölüm 5’de ele alınan geri yansıtma algoritması yapay sinir ağı tasarımında en popüler yaklaşımdır. Fakat bu algoritma giriş ve çıkış eşlerinin statik olması nedeni ile bir çok uygulamada sınırlamalar içermektedir. Çok katmanlı ağlar bu anlamda eğitildiklerinde statik x girişi şekil 6.1’de görüldüğü gibi, yine statik olan y çıkışına eşleştirilmektedir. Bu yapı imge tanıma problemlerini rahatlıkla çözebilmektedir. Çünkü giriş ve çıkış zamandan bağımsız olarak ele alınmaktadır.



Şekil 6.1 Statik çok katmanlı ağın blok yapısı

Standart statik geri yansıtma algoritması durağan, doğrusal olmayan zaman serileri için kestirimci olarak da kullanılabilir. Bir zaman serisinin istatistiksel özellikleri zamanla değişmiyorsa o zaman serisi durağandır denir (Orfanidis 1988) Şekil 6.2’de statik çok katmanlı ağı kestirimci olarak kullanılışı görülmektedir. Şekilde Z^{-1}



birim

Şekil 6.2 Doğrusal olmayan kestirimci olarak çok katmanlı ağı kullanılışı

geciktirme elemanını göstermektedir. Giriş vektörü x 'in geçmiş değerlerinden oluşmaktadır ($x=[x(n-1), x(n-2), \dots, x(n-p)]$) ve p kestirim derecesini göstermektedir. $y(n)$ skaleri çok katmanlı ağı x giriş vektörüne karşı üretmiş olduğu çıkışı göstermektedir. Bu işleme gelecek için bir adımlık kestirim adı verilir. $x(n)$ değeri ağ için istenen değeri göstermektedir.

Bilindiği gibi gerçek uygulamaların çoğunda zaman önemlidir. Örnek olarak ses, işaret ve kontrol ile ilgili uygulamalarının çoğu zamanla değişken yapı göstermektedirler. Bu uygulamalardaki sorun zamanın nasıl temsil edileceğidir. Yani çok katmanlı bir ağı zamanla değişen işaretlerle nasıl uğraşacağı bir sorun teşkil etmektedir. Bu sorun yapay sinir ağına dinamik özellikler ekleyerek ortadan kaldırılabilir.

Bir yapay sinir ağının dinamik olabilmesi için belleğe sahip olması gereklidir (Elman 1990). Bunun bir yolu ağı ağırlık yapısına zaman gecikmeleri ilave etmek ve bunları eğitim anında ayarlamaktır. Yapay sinir ağlarında zaman gecikmelerinin kullanılması

nörobiyolojik açıdan uygun olmaktadır. Bu güne kadar ki nörobiyolojik çalışmalar beyinde bilginin işlenmesi anında zaman gecikmelerinin önemini ortaya koymuşlardır.

Lang (1988) ve Waibel (1989)'nin tanıttıkları ağ yapılarında zaman gecikmeleri kullanılmıştır. Bu ağlar zaman gecikmeli ağlar(Time Delay Neural Networks) olarak bilinir. Zaman gecikmeli yapay sinir ağları bu güne kadar özellikle ses tanıma uygulamalarında başarılı bir şekilde uygulanmıştır.

Zaman gecikmeli yapay sinir ağları gerçekte her bir ağırlığında sonlu dürtü yanıtı olan filtre içermektedir (Wan 1990). Bu yapının keşfedilmesi ile çok katmanlı FIR ağı geliştirilmektedir. Bu ağın eğitimi standart geri yansıtma algoritması ile gerçekleştirilmiştir. Fakat geçici geri yansıtma algoritması ile eğitildiğinde daha iyi sonuçlar verdiği görülmüştür (Wan 1994).

Bir yapay sinir ağını dinamik yapmanın diğer bir yolu da ağı geri dönüşümlü yapmaktır (recurrent). Ağı dinamik bir yapıya sokmak için ağın tasarımına geri beslemeler ilave edilmelidir. Geri dönüşümlü ağı eğitmek için iki temel yaklaşım geliştirilmiştir(Haykin 1994). Bunlar, Zaman Aralıklı Geri Yansıtma Öğrenme ve Gerçek Zaman Geçici Eğiticili Geri Dönüşümlü Öğrenme Algoritmalarıdır. Zaman aralıklı geri yansıtma eğitimde, geri dönüşümlü ağın belirli zaman aralıklarında ileri yönlü ağa benzer davranışta olduğu kabul edilmiştir. Bu düşünce ilk önce Werbos tarafından 1974 yılında doktora çalışması olarak gerçekleştirilmiştir. Gerçek Zaman Geçici Eğiticili Geri Dönüşümlü eğitimin bir türü 1989 yılında Williams ve Zipser tarafından geliştirilmiştir. Algoritma çıkıştan girişe zaman gecikmeli geri beslemeler ile oluşturulmuştur.

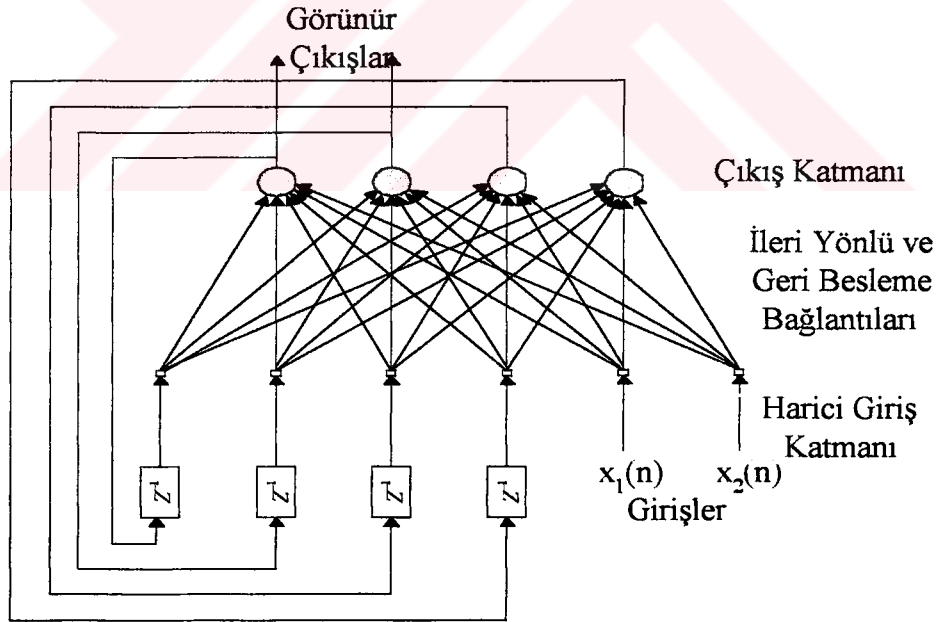
6.2.Gerçek Zaman Geri Dönüşümlü Ağ Yapısı

Bu kısımda geri dönüşümlü ağın eğitimi için farklı bir ağ yapısı ele alınmıştır. Bu ağ yapısında ağ sürekli olarak eğitime devam etmektedir. Bu tür ağlar gerçek zaman geri dönüşümlü ağlar olarak isimlendirilir. Geri dönüşümlü ağların en önemli özellikleri , zamanla değişen girişlere uyum sağlayabilmeleridir.

Toplam N tane çıkış hücresi ve M tane harici girişi olan ağ düşünülürse. n anı için, $x(n)$; $M \times 1$ 'lik giriş ve $n+1$ anın için $y(n+1)$ $N \times 1$ 'lik üretilen çıkış vektörü olsun. Giriş vektörü $x(n)$ ve geciktirilmiş çıkış vektörü $y(n)$; $(M+N) \times 1$ 'lik $u_i(n)$ net giriş vektörünü oluşturur. A , $u_i(n)$ üzerinde harici girişler olan $x_i(n)$ kümesini, B de çıkıştan gelen geciktirilmiş $y_i(n)$ leri gösterirse net giriş ifadesi denklem 6.1'deki gibi elde edilmiş olur.

$$u_i(n) = \begin{cases} x_i(n) & i \in A \\ y_i(n) & i \in B \end{cases} \quad (6.1)$$

Ağda iki ayrı katmanı giriş-çıkış bağlantı ve işlem katmanı olarak ikiye ayırabiliriz. Şekil 6.3'de gerçek zaman geri dönüşümlü ağ görülmektedir. Şekilden görüleceği gibi harici giriş sayısı (M) 2 ve toplam hücre sayısı (N) 4 olarak alınmıştır ve ağ tam bağlantılıdır. Ağda toplam MN tane ileri yönlü ve N^2 tane geri besleme bağlantı bulunmaktadır. Geri beslemelerden N tanesi öz geri beslemedir. Yani hücre çıkışları kendi girişlerine bağlıdır. Ağın ağırlık matrisinin boyutu $N \times (M+N)$ 'dir.



Şekil 6.3 Gerçek zaman geri dönüşümlü ağ yapısı

n anında j çıkış hücresinin içsel aktivitesi denklem 6.2'deki gibidir.

$$v_j(n) = \sum_{i \in A \cup B} w_{ji}(n) u_i(n) \quad (6.2)$$

Burada , $A \cup B$ A ve B 'nin birleşimidir. $n+1$ anında j hücresinin çıkışı içsel aktivitesinin doğrusal olmayan bir fonksiyondan geçirilmesi ile denklem 6.3'deki gibi elde edilir.

$$y_j(n+1) = \mathcal{G}(v_j(n)) \quad (6.3)$$

Denklem 6.2 ve 6.3 ağ dinamiğini oluşturmaktadırlar. $x(n)$ harici girişi $n+1$ anına kadar ağın herhangi bir hücre çıkışını etkilemez. Eldeki bu dinamik eşitliklerle gerçek zaman geçici eğitici öğrenme algoritması en-dik iniş metoduna yaklaşım taban alınarak elde edilmiştir.

6.3.1. Gerçek-zaman geçici-eğitici öğrenme algoritması

Gerçek zaman geri dönüşümlü ağ için $\zeta(n)$ görünür çıkışlar olarak kabul edilirse. Ağdaki diğer hücreler gizlidir. Bu ayırım yapıldıktan sonra çıkışlardaki hata denklem 6.4'deki gibi tanımlanır.

$$e_j(n) = \begin{cases} d_x(n) - y_j(n) & j \in \zeta(n) \\ 0 & \text{Diğer durumlar} \end{cases} \quad (6.4)$$

n anı için klasik geri geri yansıtma algoritmasında olduğu gibi toplam ortalama karesel hata denklem 6.5'deki gibi ifade edilebilir.

$$\varepsilon_{\text{top}} = \frac{1}{2} \sum_n \sum_j e_j^2(n) \quad (6.5)$$

Denklem 6.5 ağ için maliyet fonksiyonunu göstermektedir. Maliyet fonksiyonunun ağırlıklara göre en küçüklenmesi en dik iniş (steepest Descent) metodu ile elde edilmektedir. Bu yöntemde gradiyent matrisine ihtiyaç duyulmaktadır. Gradyent ifadesi denklem 6.6'daki gibi yazılabilir.

$$\nabla_w \varepsilon_{top} = \sum_n \frac{\partial \varepsilon(n)}{\partial w} \quad (6.6)$$

Burada $\nabla_w \varepsilon_{top}$; $\varepsilon(n)$ 'in ağırlıklara göre gradiyentidir. İstenirse bu gradiyent ifadesi ile bir güncelleme eşitliği gerçekleştirilebilir. Fakat gerçek zaman geri dönüşümlü ağ için bir öğrenme algoritması geliştirebilmek amacıyla gradiyentin ani kestirimini kullanılmalıdır. Böylece en-dik iniş algoritması için bir yakınsama elde edilmiş olur. Her hangi bir $w_{kl}(n)$ ağırlığı için ağırlık değişimi denklem 6.7'deki gibidir.

$$\Delta w_{kl}(n) = -\eta \frac{\partial \varepsilon(n)}{\partial w_{kl}(n)} \quad (6.7)$$

Ağırlık değişimlerini hesaplamak için gerekli olan $\partial \varepsilon(n) / \partial w_{kl}(n)$ kısmi türevinin çözümü denklem 6.8' deki eşitlikle sonuçlanmaktadır(Haykin 1994).

$$\pi_{kl}^j(n+1) = \mathcal{G}'(v_j(n)) \left[\sum_i w_{ji}(n) \pi_{kl}^j(n+1) + \delta_{kj} u_i(n) \right] \quad (6.8)$$

Bu eşitlikte, parçalı türev işleminin sonucu olan π_{kl}^j üç indisli dinamik bir değişkendir ve bu indeksler $j \in B$, $k \in B$ ve $l \in A \cup B$ olarak tanımlanırlar. δ_{kj} kroneker deltasıdır. Bu değişken sadece $k=j$ iken 1 diğer durumlarda 0 dir. $\mathcal{G}'(.)$ doğrusal olmayan sigmoidal aktivasyon fonksiyonun türevidir. Bu eşitliğin başlangıç değeri ($\pi_{kl}^j(0) = 0$) sıfır olarak alınır.

Her zaman adımı için ağdaki ağırlık değişimleri dinamik eşitliğin kullanımı ile denklem 6.9'daki gibi elde edilir.

$$\Delta w_{kl}(n) = -\eta \sum_j e_j(n) \cdot \pi_{kl}^j(n) \quad (6.9)$$

Denklem 6.9'daki ağırlık değişimleri elde edildikten sonra ağırlık güncellemesi denklem 6.10'daki gibi olur.

$$w_{kl}(n+1) = w_{kl}(n) + \Delta w_{kl}(n) \quad (6.10)$$

$\nabla_w \varepsilon(n)$ anlık gradiyentinin kullanımı ile Gerçek Zaman Geri Dönüşümlü Öğrenme gerçek gradiyent $\nabla_w \varepsilon_{top}$ 'ı kullanan Gerçek Zaman Olmayan Öğrenmeden farklıdır. Geri dönüşümlü ağı eğitmek için gerçek gradiyent ifadesinin kullanımı negatif en-dik inişi garantilemeyebilir. Bu yüzden anlık gradiyentin kullanımı geri dönüşümlü ağı için uygun düşmektedir. Fakat bu iki gradiyent hesabı uygun öğrenme parametresinin kullanımı ile oldukça yakın hale getirilebilir. Ayrıca gerçek gradiyentin kullanılmayışı ile hata yüzeyindeki en küçüğü arama yörüngesinde sapmalar görülebilir.

Algoritmada tanımlanan $j \in B$, $k \in B$ ve $l \in A \cup B$ indeksleri ile N hücresi, M harici girişi olan ve tam bağlantılı ağı için n anında $N(N^2+NM)$ tane π_{kl}^j dinamik değişkeni elde edilmektedir. Bu dinamik değişken $(N^2+NM) \times N$ boyutunda matris olarak düşünülebilir. Bu matrisin her bir satırı bir ağırlığa ve her bir sütunu da bir sinir hücresini göstermektedir.

BÖLÜM 7. YAPAY SİNİR AĞLARI İLE DOĞRUSAL OLMAYAN SİSTEM

TANILAMA

Bilimde bir çok durumda sistemlerin önceki ölçümlerinden gelecekteki davranışlarının değerlendirilmesi istenir. Bu aslında klasik fiziğin temelidir. Bir örnek vermek gerekirse, bir cisim için önce hareket denklemi oluşturulur daha sonra gelecekteki konumunu bulabilmek için bu hareket denklemin integrali alınır.

Fakat yukarıda bahsedilenler doğrusal olmayan karmaşık sistemlerde başarısız olmaktadır. Buna örnek olarak akışkan hareketleri, ekonomi , fizyolojik işaretler verilebilir. Çok özel durumlar dışında akışkan dinamiğinin çözülebilmesi için gerekli eşitliklerin elde edilmesi oldukça zordur. Çünkü çok boyutlu bir uzayda hareketi izlemek oldukça zordur. Fakat yüksek dereceden dinamik bir sistemin derecesi azaltılabilir. Yani dinamik sistem alt uzaylarda ele alınabilir. Alt uzayı tanımlayacak olan koordinatları belirleyebilirsek sorun basit işlemlerle çözülebilir. Bu ilk planda zormuş gibi gözükse de son yıllardaki bilimsel gelişmeler ile bu konuda önemli adımlar atılmıştır.

Eldeki problemi alt uzaylarda çözüme ulaştırmak için önceki ölçülerden faydalanmak oldukça iyi bir yöntemdir. Takins 1981 yılında, alınan öncül ölçümlerin gelecek için kestirim yapabilmesinin alınan ölçümlerin sayısına bağlı olduğunu gösterdi (Hertz at. al. 1991).

7.1.Yapay Sinir Ağları ile Modelleme

Karmaşık dinamik sistemler için kestirimle çözüme ulaşmak oldukça kolaydır. Takins'in teoremi istenen değere eksterpolasyon hakkında pek fazla bilgi vermemektedir. Bu sorunla birlikte yapay sinir ağları devreye girmektedir. Bu düşünce ile ağ; $x(t)$, $x(t+\Delta)$, $x(t+2\Delta)$ $x(t+(m-1)\Delta)$ değişkenleri ile eğitilir. Bu eğitimde hedef değer $x(t+T)$ olarak alınmaktadır. Bu işlem eksterpolasyon benzeri bir yakınsama fonksiyonudur. Bu fonksiyon daha sonra ağ ağırlıkları ile parametrelendirilir.

Doğrusal optimum filtreleme, ortalama karesel hata kriterinden yola çıkılarak gerçekleştirilmektedir. Durağan ortamda Wiener filtreler, bu kriterin en küçüklenmesi ile gerçekleştirilmektedirler. Bu tür yaklaşımlarda en fazla ikinci derece istatistiksel bilgiler kullanılabilir. Bu sınırlama ile doğrusal filtreleme Gauss olmayan girişlerden bilgi elde edilememektedir. Gauss olmayan girişler işaret işlemede çok karşılaşılan bir durumdur. Bu sorunu aşmak için çözüm yöntemlerinde doğrusal olmayan operatörlerin kullanılması uygundur. Bunun amacı yüksek dereceden istatistiksel bilgileri hesaba katmaktır (Haykin 1996).

Yukarıda bahsedilen nedenlerden dolayı ayrık zaman işaretlerinin veya sistemlerinin parametrelendirilebilmesi için doğrusal ve doğrusal olmayan iki farklı durum düşünülmelidir. Bu iki farklı durumu incelemek için denklem 7.1 deki, daha önce bahsedilen doğrusal olmayan NAR (nonlinear auto-regressive model) ele alınmıştır (Cottrel and Gigard 1995).

$$y(n) = f_w(y(n-1), y(n-2), \dots, y(n-p)) + \varepsilon(n) \quad (7.1)$$

Burada , f bir fonksiyon , w parametre vektörü, $\varepsilon(n)$ Bağımsız , sonlu varyanslı ve sıfır ortalamalı rasgele değişkenlerden oluşmuş işarettir.

7.1.1. Doğrusal durum

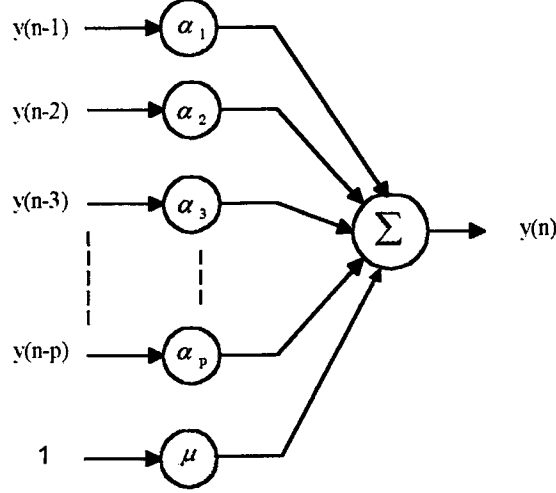
Bu durumda denklem 7.1'deki f_w fonksiyonu parametrelere göre doğrusaldır ve $y(n)$ ise denklem 7.2'deki gibi doğrusal öz bağımlı bir süreçtir.

$$y(n) = \mu + \sum_{i=1}^P \alpha_i y(n-i) + \varepsilon(n) \quad (7.2)$$

Bu modelde yapılan kabuller aşağıdaki gibidir.

- $y(n)$ 'in durağandır
- $E[y(n)]$ sabittir
- $\varepsilon(n)$ sıfır ortalamalı sonlu varyanslı bağımsız dizidir
- Polinomun tüm kökleri birim çemberin içindedir (kararlılık koşulu)

Denklem 7.2 doğrusal yapay sinir ağı olan, p girişli uyarlanabilir doğrusal eleman (ADALINE) ile şekil 7.1'deki gibi gerçekleştirilebilir. Eşitlikteki μ eşik değeridir.



Şekil 7.1 Uyarlanabilir Doğrusal Eleman (ADALINE)

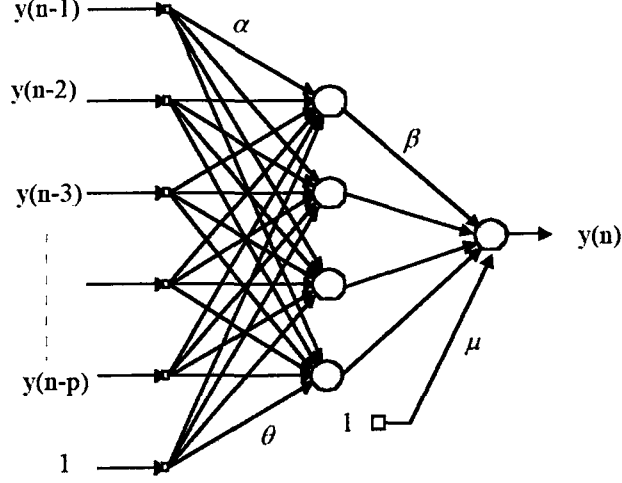
Giriş olarak verilen gözlemden, yani $y(n-1), y(n-2), \dots, y(n-p)$ 'den $y(n)$ 'in kestirimi gerçekleştirilir. Daha sonra Kestirim ile gerçek değer arasındaki fark ortalama karesel anlamda en küçüklendir. En küçükleme işlemi bilinen en küçük kareler (LS) veya en küçük ortalama karesel (LMS) algoritmaları ile gerçekleştirilebilir. Tez çalışmasında doğrusal durumlar esas amaç olmadığı için bu konunun ayrıntılarına girilmemiştir. ayrıntılı bilgi Freeman (1992)'den bulunabilir.

7.1.2. Doğrusal olmayan durum

Denklem 7.1'deki eşitlikte bulunan f_w fonksiyonu çok katmanlı bir ağ tarafından gerçekleştirilebilir. Burada, w ağ parametrelerini ($\mu, \alpha, \beta, \theta$) gösteren vektördür. Ağ şekil 7.2 'deki gibidir. Model parametreleri cinsinden ağ çıkışı denklem 7.3'deki gibi ifade edilir (Cottrel and Gigard 1995).

$$f_w(y(n-1), \dots, y(n-p)) = \mu + \sum_{j=0}^k \beta_j \mathcal{G} \left(\sum_{i=1}^p \alpha_{ji} y(n-i) + \theta_j \right) \quad (7.3)$$

Şekil 7.2'deki notasyona göre $p+1$ tane giriş , sigmoid aktivasyona sahip k adet gizli katman ve bir doğrusal çıkış hücresi bulunmaktadır. Ağın eğitimi ortalama karesel hatanın en küçüklenmesi ile gerçekleştirilir.



Şekil 4.2 Doğrusal olmayan model olarak çok katmanlı ağ

Doğrusal modeller (Aktivasyon fonksiyonu doğrusal olan) doğrusal olmayan modelin alt modelleri olarak düşünülebilir. Bu yüzden doğrusal olmayan çok katmanlı algılayıcı doğrusal modelden daha kötü olamaz.

7.2. Doğrusal Olmayan Dinamik Sistem Modelleri

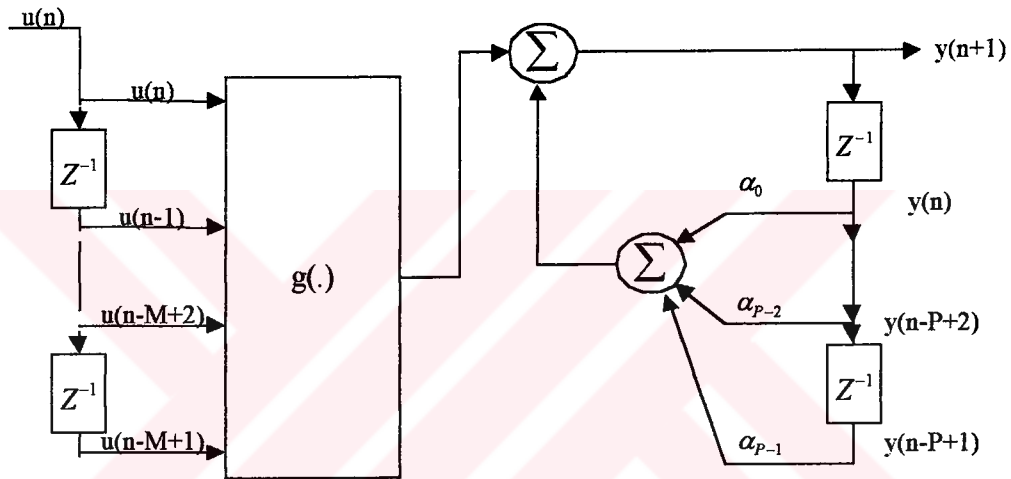
Bahsedilen yapay sinir ağları görüldüğü gibi statik yapıdadırlar. Statik ağırlıklara sahip yapay sinir ağlarında girişler ve çıkışlar zamandan bağımsız olarak ağa sunulmaktadır. Dinamik sistemlerde ise ağı eğitmek için zaman bağımlı giriş ve hedef değerler kullanılmaktadır. Yani eğitimde kullanılan bilgiler bir önceki ağ durumundan etkilenebilmektedirler (Narendra 1990). Çok katmanlı ağlar dinamik sistemleri modellemede kullanılabilirler. Bu işlem aşağıda görüleceği gibi girişlerde, çıkışlarda veya her ikisinde de birim geciktirme elemanları ile gerçekleştirilmektedir.

Doğrusal olmayan dinamik modelleri tanımlayabilmek için dört farklı temel tanımlama modeli düşünülebilir. Bu modeller birbirlerinden , girişin veya çıkışın geçmiş değerlerinden çıkışın nasıl elde edildiğine bağlı olarak ayrılmaktadırlar. Modeller doğrusal olmayan fark eşitlikleri olarak aşağıdaki gibi ifade edilirler.

Model I : Çıkış $y(n+1)$, $n+1$ anında doğrusal olarak çıkışın P geçmiş değeri $y(n-1)$, $y(n-2)$,..... $y(n-P+1)$ ' e ve giriş $u(n)$ 'in doğrusal olmayan M geçmiş değeri $u(n)$, $u(n-1)$ $u(n-M)$ 'e bağlıdır. Bu model denklem 7.4'deki gibi ifade edilebilir ve şekli Şekil 7.3'deki gibidir.

$$y(n+1) = \sum_{i=0}^{P-1} \alpha_i y(n-i) + g(u(n), u(n-1), \dots, u(n-M+1)) \quad (7.4)$$

Burada, $g(.)$ doğrusal olmayan ve kendi parametrelerine göre türevi alınabilen bir fonksiyondur.



Şekil 7.3 Doğrusal olmayan Dinamik model-I

Model II: $y(n+1)$ çıkışı $n+1$ anında çıkışın doğrusal olmayan P geçmiş değeri ve girişin M geçmiş değerine bağlıdır. Bu model denklem 7.5'deki gibi ifade edilir ve şekli Şekil 7.4'deki gibidir.

$$y(n+1) = f(y(n), y(n-1), \dots, y(n-P+1)) + \sum_{i=0}^{M-1} \beta_i u(n-i) \quad (7.5)$$

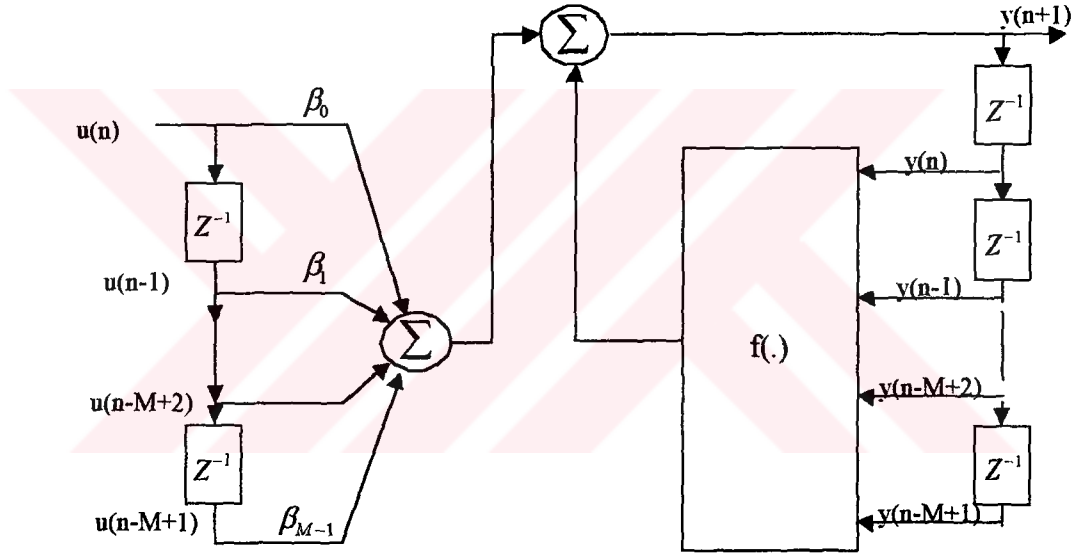
Burada $f(.)$ yine doğrusal olmayan ve kendi parametrelerine göre türevi alınabilen bir fonksiyondur.

Model III: $y(n+1)$ çıkışı $n+1$ anında çıkışın ve girişin doğrusal olmayan sırası ile P , M geçmiş değerinin toplamına bağlıdır. Bu model denklem 7.6'daki gibi ifade edilir ve şekil 7.5'deki gibidir.

$$y(n+1) = f(y(n), \dots, y(n-P+1)) + g(u(n), \dots, u(n-M+1)) \quad (7.6)$$

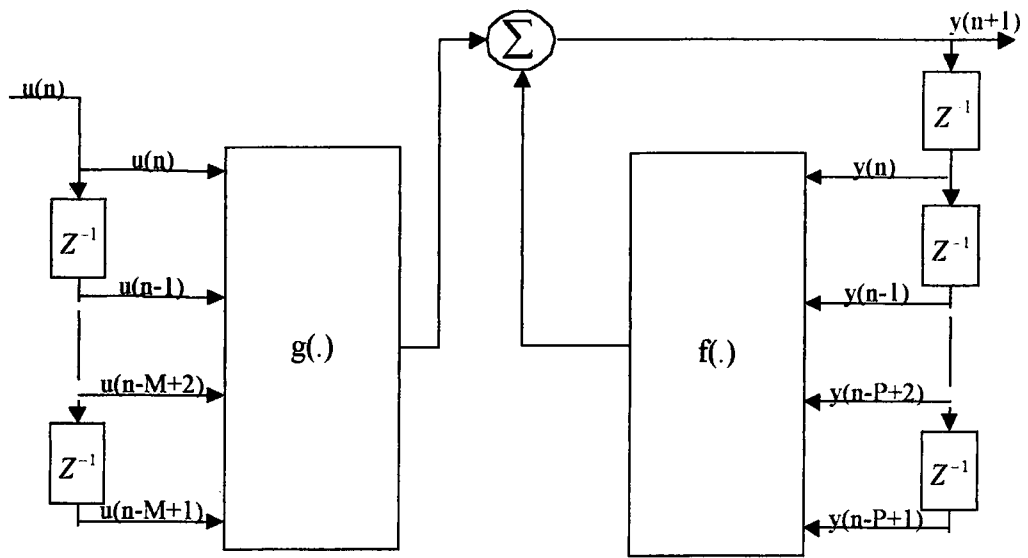
Model IV: $y(n+1)$ çıkışı $n+1$ anında çıkışın ve girişin doğrusal olmayan P , M geçmiş değerinin her ikisine birden bağlıdır. Bu model denklem 7.7'deki gibi ifade edilir ve şekil 7.6'deki gibidir.

$$y(n+1) = f(y(n), \dots, y(n-P+1), u(n), \dots, u(n-M+1)) \quad (7.7)$$

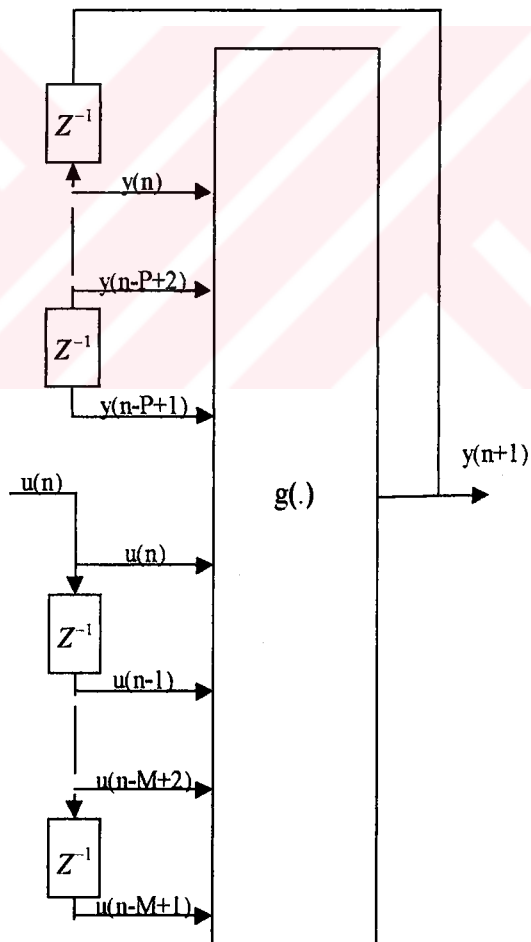


Şekil 7.4 Doğrusal olmayan Dinamik model-II

Model IV diğer tüm modelleri kapsamı nedeni ile en genel olanıdır. Fakat en genel model olmasına karşın matematiksel olarak izlenmesi en zor olanıdır. Tüm modellerdeki f ve g doğrusal olmayan fonksiyonlarının çok katmanlı ağırlık geri yansıtma algoritması ile eğitilmesi giriş çıkış eşleştirmesini gerçekleştirir. Hangi modelin kullanılacağı uygulamaya göre değişiklik göstermektedir.



Şekil 7.5 Doğrusal olmayan Dinamik model-III



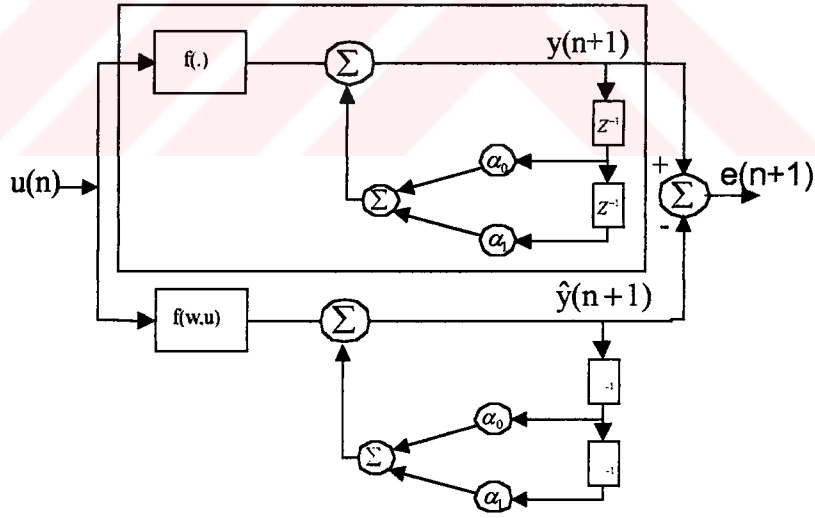
Şekil 7.6 Doğrusal olmayan Dinamik model-IV

7.3. Sistem Tanılama Yöntemleri

Tanılama problemi sistemi temsil edecek olan parametrelerin uygun şekilde ayarlanması işlemidir. Bu parametre ayarlaması performans fonksiyonunu optimize edilmesi ile olur. Performans fonksiyonu tanılama modeli ile gerçek model çıkışı arasındaki farkın fonksiyonudur. Doğrusal olmayan sistemler için tanılama modelinin yapısı uygun seçilmelidir. Yani tanılama modeli ile gerçek model yapıları birbirlerine benzerlik göstermelidirler. Ayrıca tanılama modeli ile gerçek model çıkışları optimizasyon işlemi başladığında aynı değere sahip olmalıdırlar. Tanılama işlemi yapay sinir ağlarının ağırlıklarının ayarlanması sayesinde gerçek modele yakınsama ile son bulur. Doğrusal olmayan dinamik bir modeli tanılamak için paralel ve seri-paralel olmak üzere iki çeşit tanılama modeli bulunmaktadır.

7.3.1.Paralel tanılama

Şekil 7.7’de daha önce şekil 7.3’de belirtilen model-I için paralel tanılama modeli görülmektedir.



Şekil 7.7 Paralel tanılama modeli

Şekil 7.7’de $P=2$ ve $M=1$ alınmıştır. Bir sistemi tanılamak için ilk önce tanılama modelinin denklem 7.8’de görüldüğü gibi yapısı belirtilmelidir.

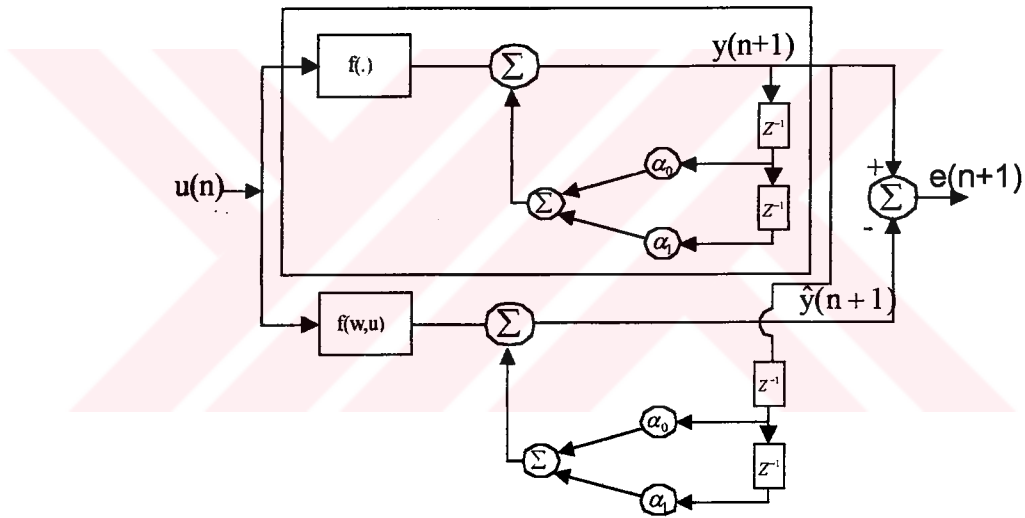
$$\hat{y}(n+1) = \hat{\alpha}_0 \hat{y}(n) + \hat{\alpha}_1 \hat{y}(n-1) + f(w, u(n)) \quad (7.8)$$

Şekil 7.7'deki tanılama $\hat{\alpha}_i$ ve ağ ağırlıklarının kestirimini gerçekleştirir. Bu kestirim daha önce bahsedildiği gibi hatanın kullanımı ile gerçekleştirilir.

Tanılamada paralel yapının kullanılması her zaman kararlı çalışmayı garantileyemez. Yani her zaman hata sifıra doğru yaklaşmayabilmektedir. Bunun nedeni tanılama modelinin çıkışının bir sonraki kestirimini etkilemesidir.

7.3.2.Seri-Paralel tanılama

Paralel modelin tersine seri paralel modelde tanılama modelinin çıkışı yerine sistem çıkışı kestirimde kullanılmaktadır. Seri- paralel tanılama modeli şekil 7.8 de verilmiştir.

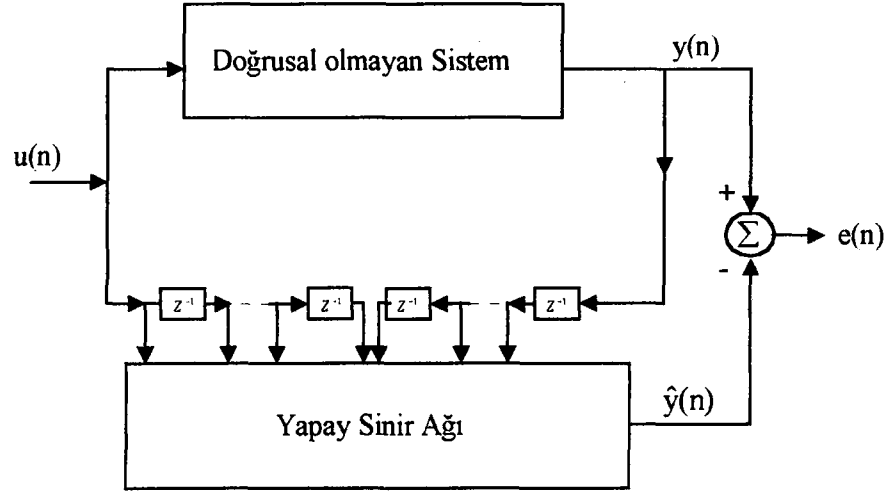


Şekil 7.8 Seri-Paralel tanılama modeli

Seri-paralel tanılama modelinin yapısı denklem 7.9'daki gibidir. Seri-paralel model kararlılık açısından paralel modele göre daha üstündür. Çünkü tanılama modelinde bir geri besleme bulunmamaktadır.

$$\hat{y}(n+1) = \hat{\alpha}_0 y(n) + \hat{\alpha}_1 y(n-1) + f(w, u(n)) \quad (7.9)$$

Seri-paralel tanılama model IV 'e uygulandığında , bilinmeyen sistem için tanılama işlemi şekil 7.9'daki blok şema kullanılabilir. Burada , ağ çıkışı gerçek sistemin giriş , çıkışının geçmiş değerlerinin ve ağ ağırlıklarının fonksiyonu olarak elde edilmektedir.



Şekil 7.9 Model-IV için tanılama blok şeması

Ağ girişinde , sistem girişi ve çıkışının geçmiş değerleri bulunmaktadır. Böyle bir yapı, bilinmeyen modelin NARX ile ifade edileceğini göstermektedir.

7.4. Model Gerçekleştirme

Modelin gerçekleştirilmesi tanılama prosedürünün son basamağı olmalıdır. Model yeterli ise kestirim hatası doğrusal ve doğrusal olmayan giriş ve çıkışın geçmiş değerlerinden bağımsız olacaktır. Bilinen tanılama yöntemleri için kullanılan bu yaklaşım yapay sinir ağları için de geçerlidir. Bir giriş ve bir çıkışlı sistem için model yeterli ise kestirim hatası denklem 7.10'daki koşulları sağlamalıdır (Chen 1990).

$$\begin{aligned}
 \Phi_{ee}(n) &= \delta(n) && \text{bir dürtü fonksiyonu} \\
 \Phi_{eu}(n) &= 0 && \text{Tüm } n \text{ 'ler için} \\
 \Phi_{e(ee)}(n) &= 0 && n \geq 0 \\
 \Phi_{u^2e}(n) &= 0 && \text{Tüm } n \text{ 'ler için} \\
 \Phi_{u^2e^2}(n) &= 0 && \text{Tüm } n \text{ 'ler için}
 \end{aligned} \tag{7.10}$$

Burada, $\Phi_{xy}(n)$ $x(t)$ ile $y(t)$ arasındaki çapraz ilişkiyi göstermektedir.

$\epsilon u(n) = e(n+1)u(n+1)$ ve $u^2(n) = u^2(n) - \overline{u^2(n)}$ olarak hesaplanabilir. Burada , $\overline{u^2(n)}$ $u^2(n)$ 'in zaman ortalaması veya ortalama değeridir. Bu ilişki fonksiyonları $\pm 1.96 / \sqrt{N}$ aralığında sağlanırsa modelin yeterli olduğu söylenebilir.

7.5. Simulasyon Çalışması

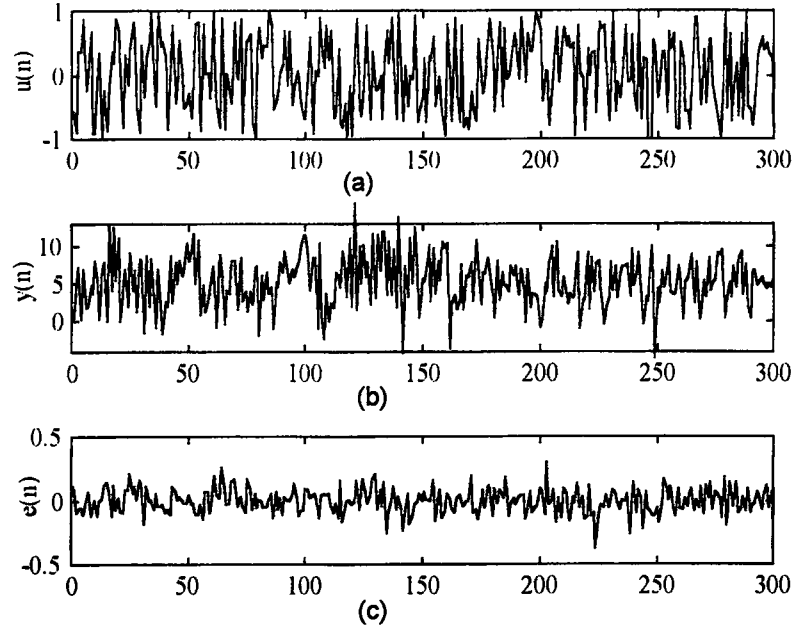
Bu bölümde doğrusal olmayan bir sistemin tanınması için simülasyon çalışması gerçekleştirilmiştir. Simülasyonda denklem 7.10 uyarınca 300 noktalık sistem çıkışı elde edilmiş ve bu zaman serisini ilk 200 noktası ağı eğitmek için kalan 100 noktası ağı test etmek için kullanılmıştır.

$$y(n) = 0.5 + 0.8y(n-2) - 1.5u^2(n-2)y(n-2) + 0.8(u(n-1) - 2)^2 + e(n) \quad (7.11)$$

Sistem girişi olan $u(n)$ $[-1 \ 1]$ aralığında rasgele sayılardan elde edilmiştir. Giriş ile ilişkisiz sistem gürültüsü olarak sıfır ortalamalı varyansı 0.04 olan beyaz gürültü kullanılmıştır. Zaman serisinin tanınması için girişin ve çıkışın üç geçmiş değeri çok katmanlı ağa girilmiştir. Hedef veya istenen değer olarak sistem çıkışından alınan 200 noktalık veri alınmıştır. Ağda bir giriş , bir gizli ve birde çıkış katmanı olmak üzere üç katman kullanılmıştır. Bahsedildiği gibi ağın giriş katmanında 6 tane düğüm , gizli katmanında 10 tane doğrusal olmayan aktivasyona sahip hücre ve çıkış katmanında ise bir tane doğrusal hücre kullanılmıştır. Sistem giriş , çıkış ve gürültüsü şekil 7.10'deki gibidir.

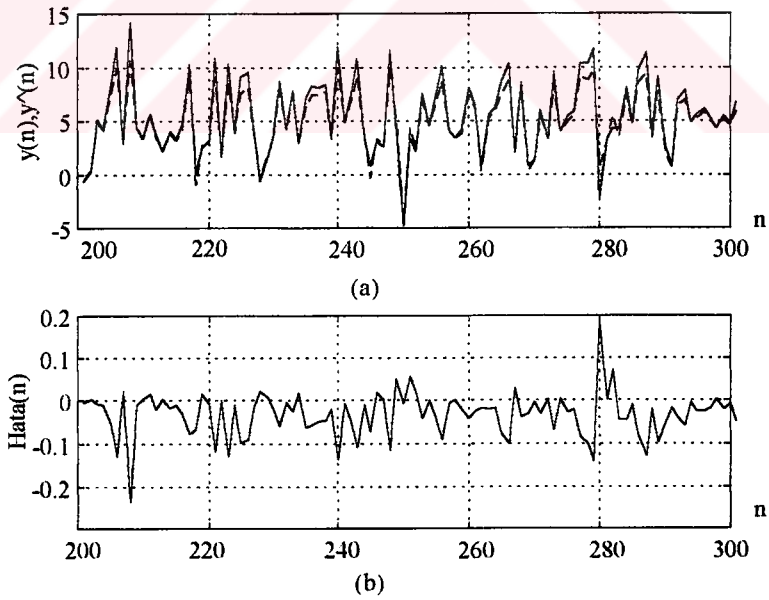
Ağ eğitimi 100 deneme ile gerçekleştirilmiştir. Ağın toplam ortalama karesel hatasının denemelere göre değişimi şekil 7.12'deki gibi elde edilmiştir.

Seçilen ağ parametrelerinin (AR ve X dereceleri ve ağda kullanılan gizli katman sayısı) yeterli olup olmadığını test etmek için alınan sonuçlara ilişki testi uygulanmıştır. Yapay sinir ağı ile gerçekleştirilen doğrusal olmayan modelleme sonuçları şekil 7.11'deki gibidir. Eğitimi tamamlanmış ağın girişine model tarafından üretilen son 100 örnek uygulanmış ve ağ çıkışı ile gerçek değer karşılaştırılmıştır ve ilişki testinin sonuçları şekil 7.13'deki gibidir.



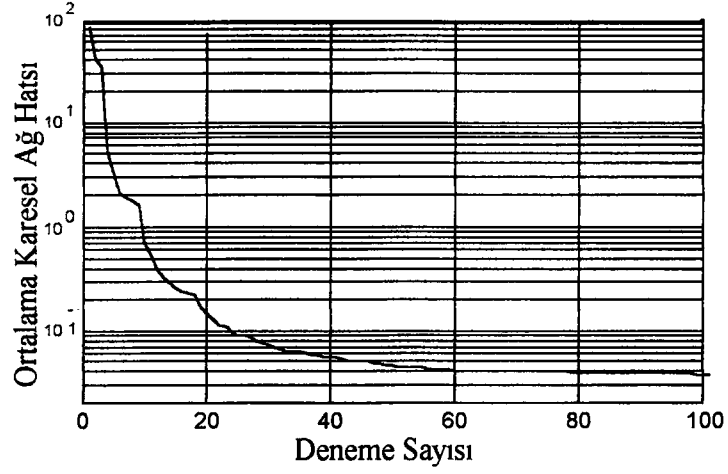
Şekil 7.10 Sistem için kullanılan işaretler

- a) [1-1] aralığında rasgele sayılardan oluşmuş giriş
- b) Sistem çıkışı
- c) Girişle ilişkisiz 0.04 varyanslı beyaz gürültü

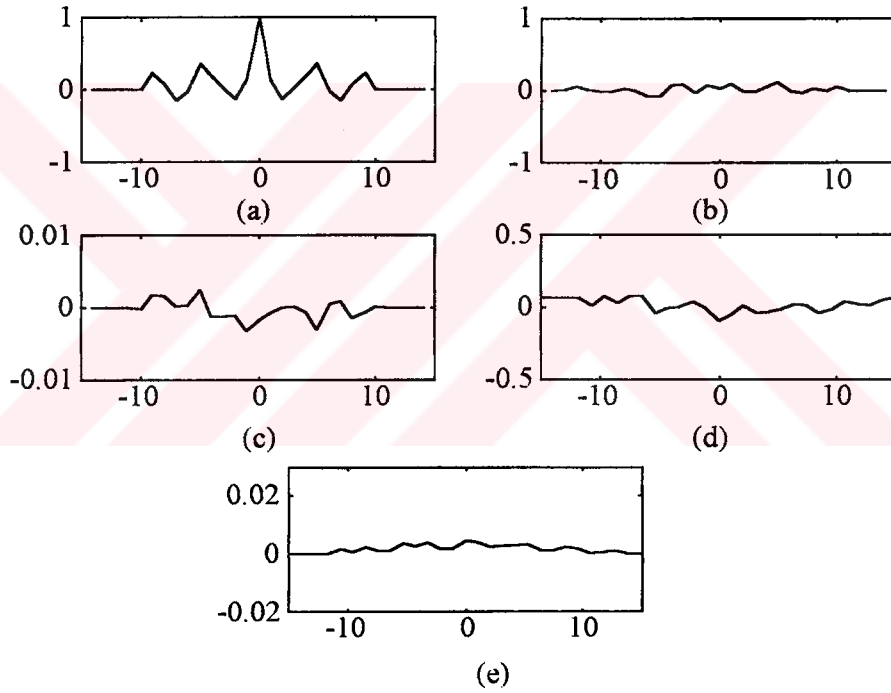


Şekil 7.11 Sistem Tanılamanın sonuçları

- a) sistemin gerçek çıkışı (---) , ağ çıkışı (—)
- b) Kestirim hatası



Şekil 7.12 Toplam ortalama karesel ağ hatası



Şekil 7.13 İlişki testi sonuçları

- a) $\Phi_{\epsilon\epsilon}$
- b) $\Phi_{\epsilon u}$
- c) $\Phi_{\epsilon(\epsilon u)}$
- d) $\Phi_{u2\epsilon}$
- e) $\Phi_{u2\epsilon2}$

BÖLÜM.8 YAPAY SİNİR AĞLARI İLE EEG MODELLEMESİ VE UYARILMIŞ POTANSİYELLERİN ÇIKARTIMI

EEG işaretleri genellikle parametrik işaret işleme yaklaşımları ile analiz edilirler. İşareti matematiksel terimler cinsinden ifade eden parametrik metotlar bir parametre seti ile ifade edilirler. Genellikle EEG, AR veya ARMA olarak beyaz gürültü ile beslenen doğrusal bir modelin çıkışı gibi düşünülerek modellenir. EEG işareti durağan bir işaret değildir. Fakat parametrik yaklaşımlarında belirli aralıklarda (1s gibi) durağan olduğu kabul edilebilirler (Hazarika 1997).

Uyarılmış potansiyellerin tespiti için ortalama alma en çok kullanılan yöntemdir. Bu yöntemde sağlıklı sonuçlar elde edebilmek için en az 80 yada 100 denemeye ihtiyaç vardır. UP'lerin son derece hızlı değişimlerin olduğu ve beyinden alındığı düşünülürse bu kadar uzun bir zamanda yanıtıcı sonuçlar elde edilebilir. Bu güne kadar UP tespitinin daha kısa sürede yapılması için giriş bölümünde bahsedildiği gibi bir çok çalışma yapılmıştır. Şimdiye kadar ki yöntemlerin çoğu UP'yi tespit etmek için uyarı sonrası işareti kullanmaktadırlar. Uyarı öncesi EEG'nin modellenmesi ile uyarı sonrası gürültü karakteristikleri hakkında bilgi elde edilmektedir (Spreccelsen and Brom 1988). Uyarı sonrası periyodu süperpozisyon ile UP ve süregelen EEG'nin toplamı denklem 8.1'deki gibi yazılabilir.

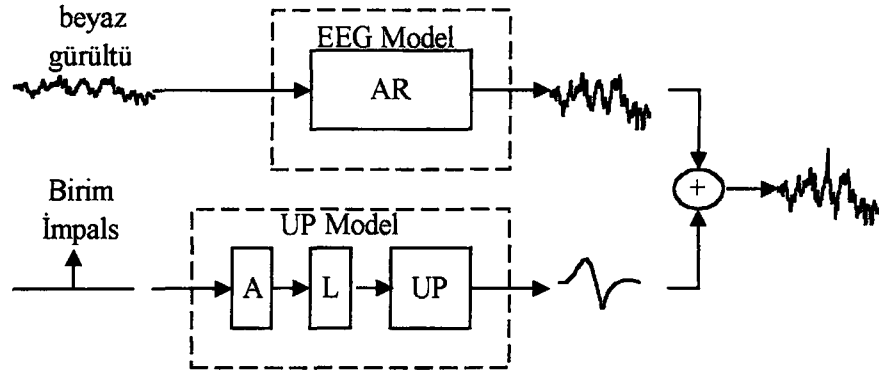
$$s(n)=u(n)+e(n) \quad (8.1)$$

Burada ,e(n) süregelen EEG aktivitesi, u(n) uyarılmış potansiyel, s(n) uyarı sonrası UP+EEG aktivitesi olarak alınmıştır.

8.1.İşaret Simülasyonu

Yapay sinir ağı ile EEG modellenmesi ve UP çıkartımı için , tez çalışmasında bahsedilen uyarı potansiyeli içeren yapay EEG işareti üretilmiştir. Gerçek işaret ile çalışmadan önce böyle bir çalışma yapmak sistemin doğruluğunun test edilmesi açısından önemlidir.

Uyarı sonrası EEG aktivitesi , uyarı ve süregelen EEG'nin toplamı olarak ifade edilebildiği yukarıda açıklanmıştır. Bu nedenle , işaret simülasyonu için iki farklı işaretin üretilmesi gereklidir. Daha sonra ayrı ayrı düşünülen bu iki işaret toplanarak uyarı sonrası toplam işaret elde edilmiştir. Fizyolojik işaret simülatörünün blok şeması Şekil 8.1' deki gibidir.



Şekil 8.1 EEG ve UP simülatörü blok diyagramı

8.1.1.EEG modeli

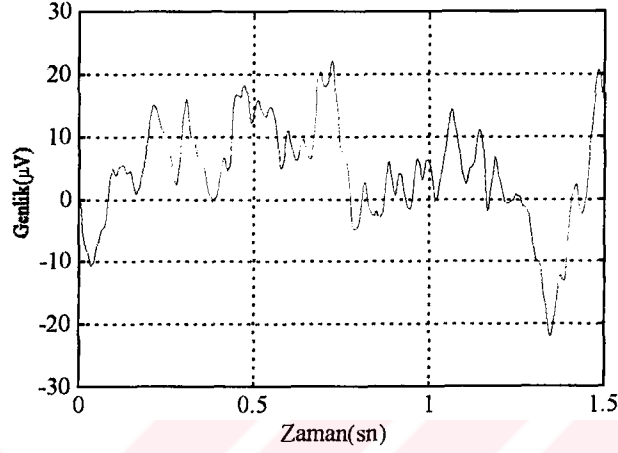
Süregelen EEG rastsal bir süreç olarak düşünülebilir ve bu yüzden beyaz gürültü ile beslenen doğrusal bir sistemin çıkışı olarak elde edilebilir. Yapay EEG üretimi AR rastsal model ile denklem 8.2'deki gibi ifade edilmiştir (Yu at. al.1994).

$$e(n) = \mathbf{a}^T \mathbf{e}(n-1) + v(n) \quad (8.2)$$

Burada , $e(n)$ süregelen EEG , \mathbf{a}^T P'inci dereceden parametre vektörü ve $v(n)$ ise beyaz gürültüdür. EEG modeli bilinen doğrusal parametre kestirimi algoritmalarından biri ile rahatlıkla elde edilebilir. Tez çalışmasında ele alınan fizyolojik verinin ilk 125 örneği alınarak en küçük kareler yöntemi(least squares) ile model elde edilmeye çalışılmıştır ve elde edilen model ile 3750 örneklik işaret elde edilmiştir. Model denklem 8.3'deki gibidir. Modelde gürültü olarak varyansı 6 olan gauss dağılımlı beyaz gürültü kullanılmıştır.

$$e(n) = 0.7889e(n-1) + 0.0496e(n-2) - 0.1950e(n-3) - 0.2310e(n-3) + 0.1105e(n-3) + v(n) \quad (8.3)$$

Yapay işaret için örnekleme frekansı 250Hz olarak alınmıştır ve bir segment ölçümün 1.5 s olduğu kabul edilmiştir. Model çıkışında elde edilen 375 örneklilik yani 1.5s'lik bir segment EEG işareti şekil 8.2 deki gibidir.



Şekil 8.2 EEG simülasyonu için AR model çıkışı

8.1.2. UP modeli

Uyarı potansiyelleri kısa süreli uyarıların cevabı olarak ortaya çıktığı için bir sistemin dürtü yanıtı olarak düşünülebilir. Bu sistem denklem 8.4'deki c parametre vektörü ile elde edilir. Bu sistemin girişi birim dürtü olan $d(n)$ 'dir.

$$u(n) = c^T u(n-1) \quad (8.4)$$

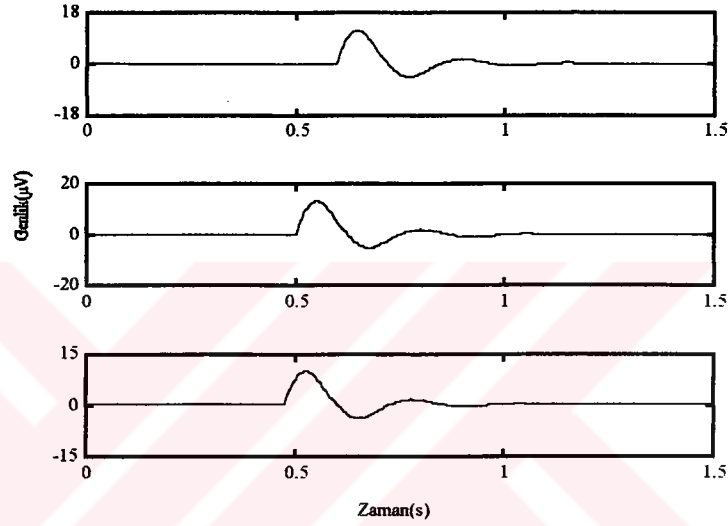
Burada c^T ; IIR filtre katsayı vektörü ($c_1, c_2, \dots, c_m, d_1, d_2, \dots, d_n$) ve $u(n-1)$ ise giriş ve çıkışın geçmiş değerlerinden oluşan giriş vektörüdür. UP üretmek için gerekli parametre vektörü denklem 8.5'deki gibidir.

$$u(n) = d(n) - 0.5d(n-1) - 0.1u(n-1) - 0.3u(n-2) \quad (8.5)$$

Bu şekilde UP'nin deterministik kısmı elde edildikten sonra oluşturulan her segment uyarı için değişkenlik göz önüne alınmalıdır. Bunun için UP işaretine rastsal genlik ve faz bilgileri ilave edilmiştir. Bu durum denklem 8.6'daki gibi ifade edilebilir (Spreckelsen and Bromm 1988).

$$u_j(n) = A_j u(n - \ell_j) \quad (8.6)$$

Burada , A_j ve ℓ_j sırası ile rastsal genlik ve faz deęişimlerini göstermektedir. Simülayon çalışması için sistem çıkışından 100 segmentlik farklı UP elde edilmiştir. İlk üç segment için elde UP'ler şekil 8.3'de gözükmektedir. Uyarı potansiyelleri 1.5 saniyelik periyotlarda ve periyodun üçte birlik kısmında başlayacak şekilde elde edilmişlerdir.



Şekil 8.3 UP simülasyonu için farklı genlik ve fazdaki ilk üç segment UP işareti

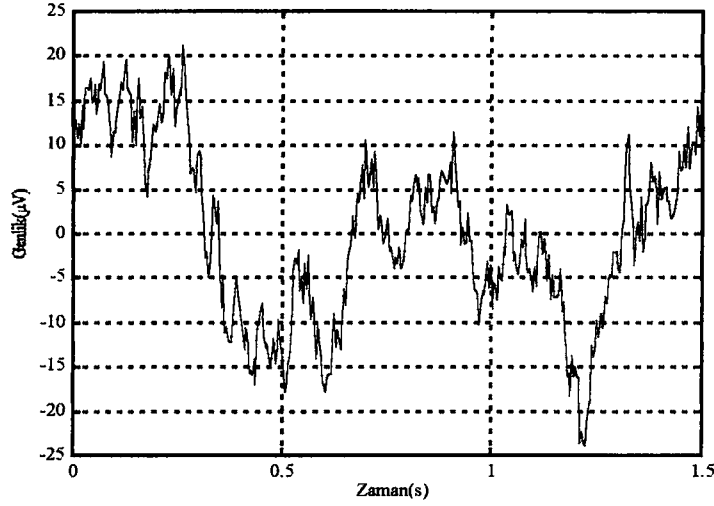
8.1.3. Birleşik model

EEG ve UP ayrı ayrı elde edildikten sonra her segment için toplam işaret oluşturulmaktadır. İşaretin örnekleme frekansı 250Hz olarak alınmış ve her bir segment için 375 örnekten oluşan işaret elde edilmiştir . Birleşik model çıkışına ait bir segmentlik işaret Şekil 8.4 deki gibidir.

8.2. Gerçek İşaret Özellikleri

UP çıkartımı için kullanılan YSA modellerinin gerçek ortamda nasıl çalışabileceğinin test edilmesi için gerçek işaret kullanılmıştır. UP içeren işaret Pz (parietal region) elektrodundan yani başın en üstünün biraz altından yüzey kaydı ile alınmıştır. Kayıta örnekleme frekansı 250 Hz olarak seçilmiş ve her segment için 1.5 saniyelik kayıt yapılmıştır. Uyarı öncesi 0.5 saniye ve uyarı sonrası ise 1 saniye kayıt yapılmıştır. 100

segmet işaret bu şartlar altında kayıt edilmiştir. Uyarı potansiyeli 126 örneğe karşılık gelecek şekilde santranc tahtası ile uyarılarak gerçekleştirilmiştir.



Şekil 8.4 UP+EEG işareti için birleşik model çıkışı

8.3. Yapay Sinir Ağları İle Uyarı Sonrası EEG Kestirimi ve UP Çıkartımı

Bu çalışmada, uyarı öncesi süregelen EEG ile uyarı sonrası EEG aktivitesi arasındaki ilişkiden yola çıkılarak UP'lerin tespiti ele alınmıştır. Yukarıda özelliklerinden bahsedilen işaretlerden UP'lerin sezinlenmesi için karışık EEG+UP işaretinden modellenen EEG çıkartılarak kalanlar UP olarak elde edilmiştir. Uyarı öncesi EEG , Çok Katmanlı ve Gerçek Zaman Ağı ile modellenmiştir. Daha sonra uyarı sonrası EEG'yi kestirmek için bu model kullanılmıştır. Model için gerekli olan ağırlıklar her bir ağ için sırası ile Geri Yansıtma ve Geçici Eğitici Gerçek Zaman Geri Dönüşümlü Öğrenme Algoritmaları ile elde edilmişlerdir. Elde edilen modellerle uyarı sonrası EEG kestirilmeye çalışılmış ve kestirim hatası UP olarak yorumlanmıştır.

Çalışmada her iki ağ eğitmek için yapay ve gerçek işaretin uyarı öncesi bilgiler kullanılmıştır. Bu yüzden uyarı öncesi EEG'nin işlem öncesinde kullanılması gereklidir. EEG modellemek için kullanılan çok katmanlı ağ bir gizli katmana sahiptir ve gizli katmandaki hücre sayısı 10 olarak alınmıştır. Bu çok katmanlı ağın giriş düğüm sayısı 4 taneden fazla alındığında yeterli olduğu gözlemlenmiştir. Çok katmanlı ağ eğitmek için geri yansıtma algoritması kullanılmıştır. Algoritmada öğrenme oranı 0.001 ve momentum katsayısı 0.95 olarak alınmıştır. Gerçek zaman geri dönüşümlü ağ için ise toplam 16 çıkış hücresi kullanılmış ve bunların sadece bir tanesi görünür çıkış

olarak kullanılmış. Diğerleri ise geri besleme hatlarına çıkış üretmek amacıyla kullanılmıştır. Ağ için girişin üç geçmiş değerinin kullanılmasının yeterli olduğu görülmüştür. Geri dönüşümlü ağ için öğrenme oranı 0.1 olarak ve ağın başlangıç şartları bölüm 6'da bahsedildiği gibi alınmıştır.

Ağların her ikisinin eğitiminde bir adımlık ileri doğru kestirim işlemini gerçekleştirmektedir. Yani işaretin geçmiş değerlerinden bir sonraki durumunun kestirimi söz konusudur. Böylelikle uyarı öncesi EEG ait NAR model YSA ile elde edilmiştir.

YSA modellerinin her ikisi ile tek segmentte UP çıkartımı işlemi gerçekleştirildikten sonra performans testi için SNR çalışması gerçekleştirilmiştir. SNR hesabında UP gücü EEG+UP işareti gücüne oranlanmıştır (Christopher 1989). Karşılaştırma yapabilmek amacı ile yapay ve gerçek işareten alınan 100 segmentlik işaretin ortalaması alınmış ve tek segmente göre SNR hesaplanmıştır. SNR hesaplaması için kullanılan eşitlik denklem 8.7'deki gibidir.

$$SNR = 10 \log \left(\frac{E[UP^2]}{E[(EEG + UP)^2]} \right) \quad (8.7)$$

Bilindiği gibi sayısal işaret işlemede frekans domeni analizleri önemlidir. Tez çalışmasında, yapay sinir ağları kullanarak tek segmentlik UP çıkartım sonuçları frekans domeninde ortalama alma ile karşılaştırılmıştır. Her iki yapay sinir ağı ile gerçekleştirilen UP çıkartımı sonuçları için Ortalama Peryodogram Yöntemi ile güç spektrumu kestirilmiştir. Peryodogram, FFT tabanlı en çok kullanılan ve elde edilmesi en kolay spectrum kestirim yöntemlerinden biridir. Ortalama peryodogram metodunda, spektrumu elde edilmek istenen zaman serisi birbirinden bağımsız ve eşit uzunlukta parçalara bölünündükten sonra her parça için bilinen FFT algoritması uygulanır. Her segment için FFT hesaplandıktan sonra segmentlerin ortalanması ile ortalama peryodogram elde edilmiş olur. Ortalama peryodogram ifadesi denklem 8.8'deki gibi dir (Akay 1996).

$$\hat{P}_{\text{parav}}(f) = \frac{1}{L} \left[\sum_{n=0}^{M-1} y(n) e^{j 2\pi f n} \right]^2 \quad (8.8)$$

Burada, L, işaretten alınan segment sayısını M, ise segmentlerdeki örnek sayısını göstermektedir.

8.4.Sonuçlar

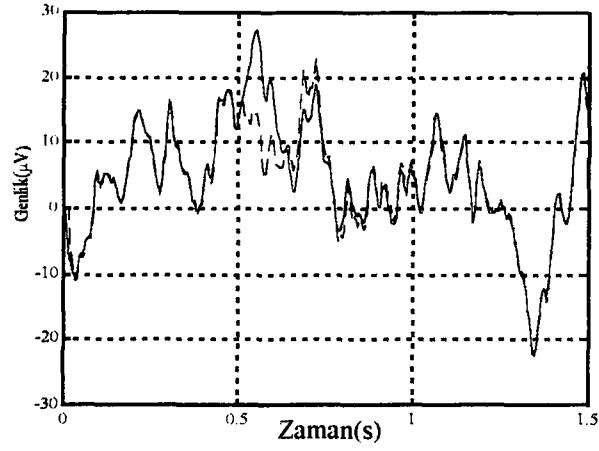
Yapay ve gerçek işaretin her iki ağ girişine uygulanması ile elde edilen sonuçlar bu bölümde ele alınmıştır. İlk önce yapay işaret için UP çıkartımı her iki ağ ile gerçekleştirilmiştir. 100 segmentten oluşan gerçek işaretin sadece ilk iki segmenti için UP çıkartımı sonuçları gösterilmiştir. Her segment işaret için uyarı sonrası EEG aktivitesi kestirilmiştir ve işaretin kestirimi ile arasındaki fark UP olarak elde edilmiş. Ortalama alma ile karşılaştırma yapabilmek amacı ile 10 segment gerçek işaret ortalaması ile 3 segment hata (UP) ortalaması karşılaştırılmıştır.

Yukarıda bahsedildiği gibi, alınan sonuçlar üzerinde, yapay ve gerçek işaret için bir SNR çalışması gerçekleştirilmiştir. SNR çalışmasının sonuçları tablo 8.1'deki gibidir. Tabloda görüleceği gibi klasik yöntemle yapay sinir ağı yaklaşımları karşılaştırılmıştır. Ortalama alma yönteminde UP olarak 100 segment işaretinin ortalaması kullanılmıştır.

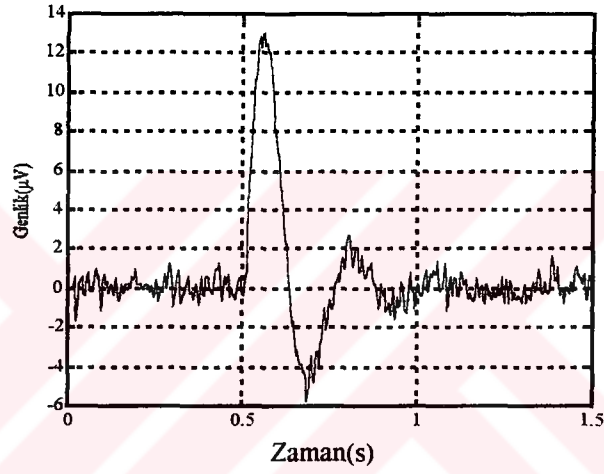
Tablo 8.1 UP çıkartımı için çeşitli yöntemlere göre SNR sonuçları

YÖNTEM	Yapay işaret için SNR(dB)	Gerçek işaret için SNR(dB)
Ortalama Alma	-9.4164	-10.4244
Çok katmanlı Ağ	-11.7608	-11.2559
Gerçek Zaman Geri Dönüşümlü Ağ	-9.5911	-9.5906

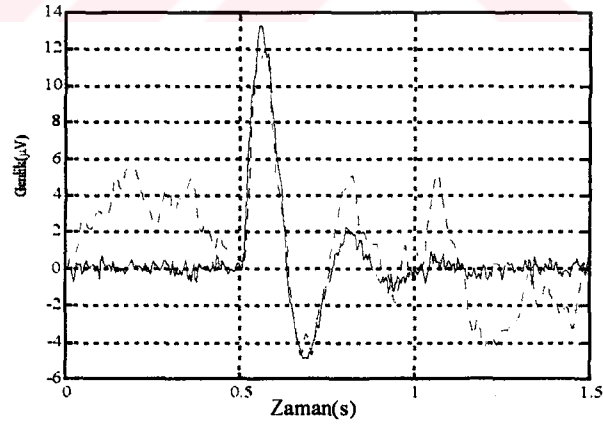
Öncelikle, yapay işaret için her iki ağ ile şekil 8.5 ve 8.6'daki sonuçlar alınmıştır. Şekil 8.7 ve 8.8 de ise gerçek işaret için tek segmentte UP çıkartım sonuçlarını göstermektedir. UP çıkartımlarının frekans domenindeki analizi her iki ağ için Şekil 8.9'daki gibi elde edilmişti. Şekil 8.9'da klasik ortalama alma ve yapay sinir ağları ile elde edilen sonuçlar karşılaştırılmıştır.



(a)



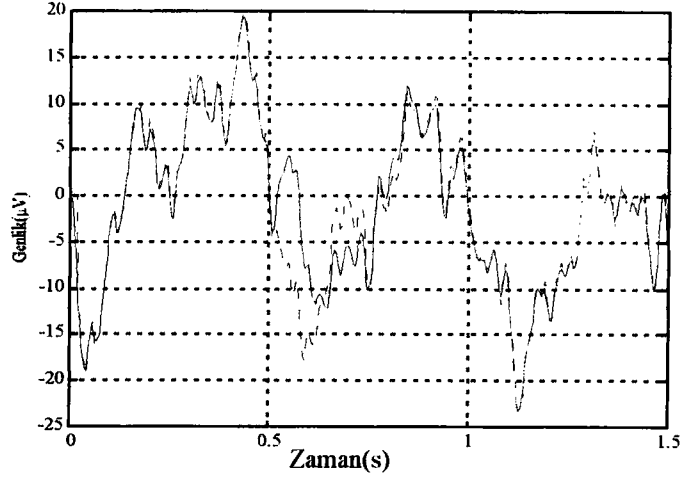
(b)



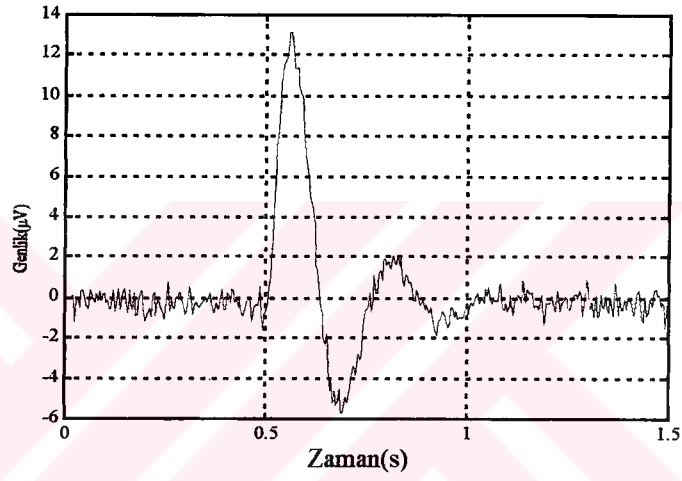
(c)

Şekil 8.5 Yapay işaret için çok katmanlı ağ ile UP çıkartımı sonuçları

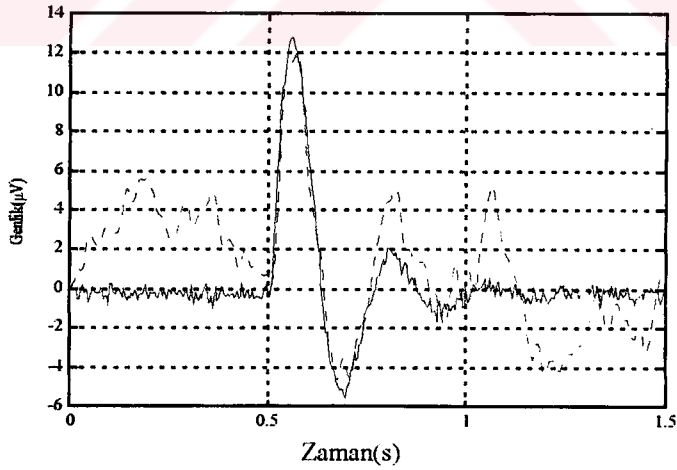
- a) Bir segment için giriş işareti (düz çizgi) ve onun kestirimi(kesikli çizgi)
- b) Bir segment için hata (UP)
- c) İlk 20 segment işaret ortalaması (kesikli çizgi) , 3 hata ortalaması (düz çizgi)



(a)



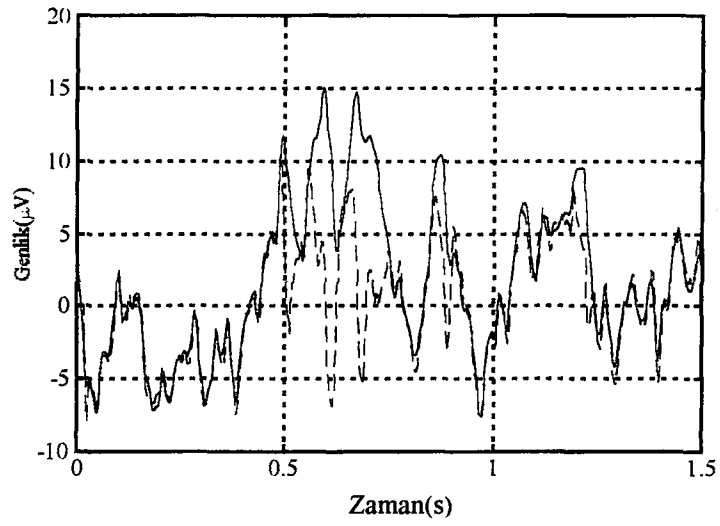
(b)



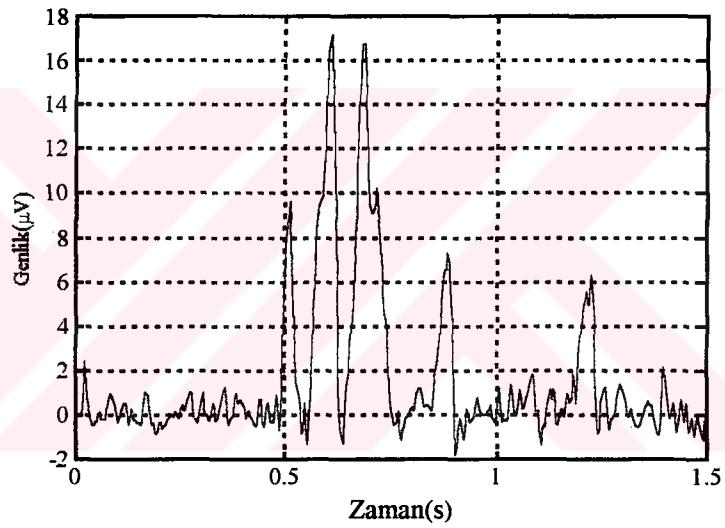
(c)

Şekil 8.6 Yapay işaret için gerçek zamanlı ağ ile UP çıkartımı sonuçları

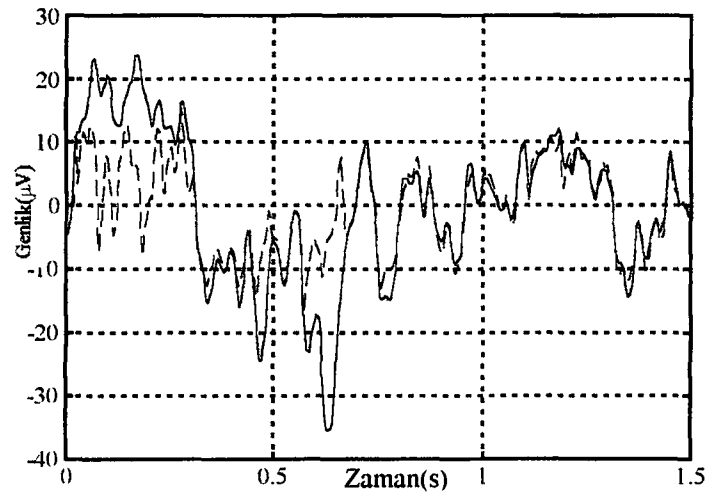
- a) Bir segment için giriş işareti (düz çizgi) ve onun kestirimi(kesikli çizgi)
- b) Bir segment için hata (UP)
- c) İlk 20 segment işaret ortalaması (kesikli çizgi) , 3 hata ortalaması (düz çizgi)



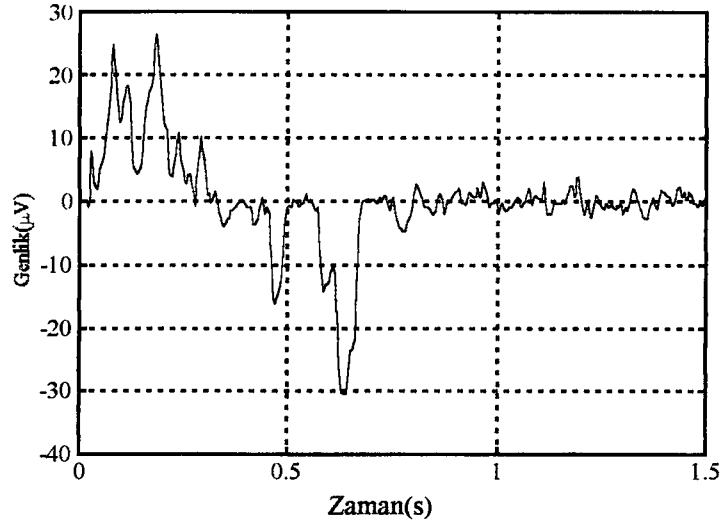
(a)



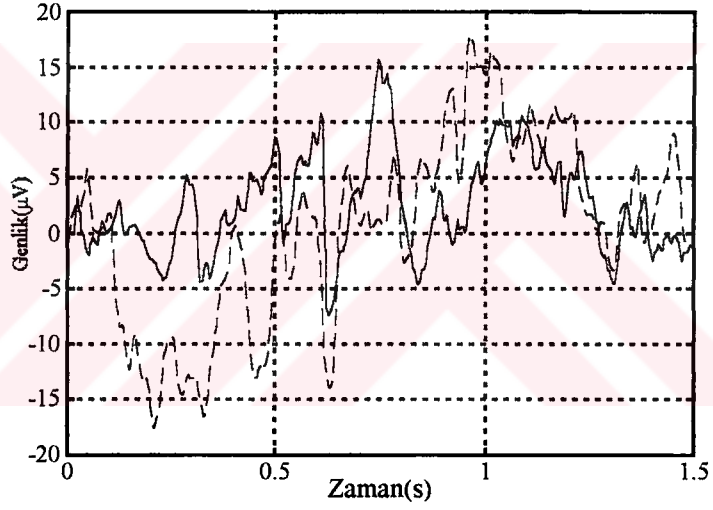
(b)



(c)



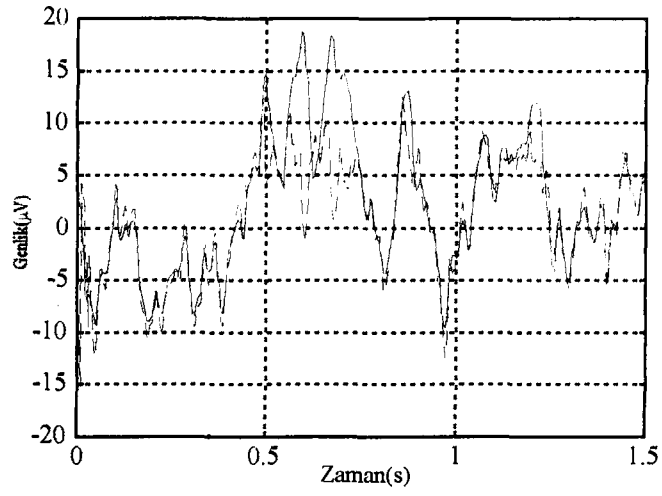
(d)



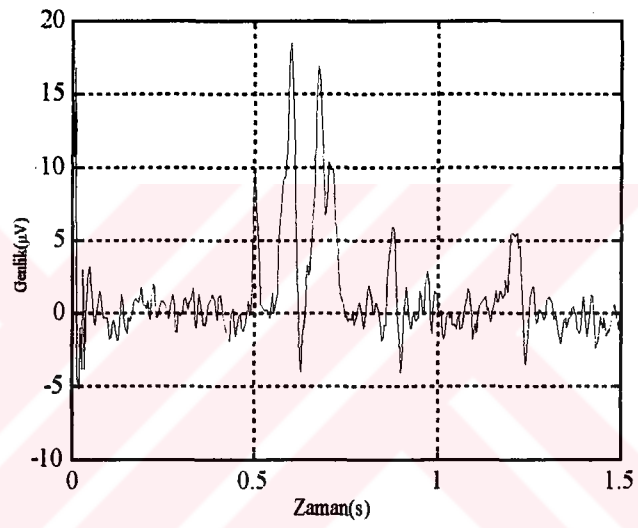
(e)

Şekil 8.7 Gerçek işaret için çok katmanlı ağ ile UP çıkartımı sonuçları

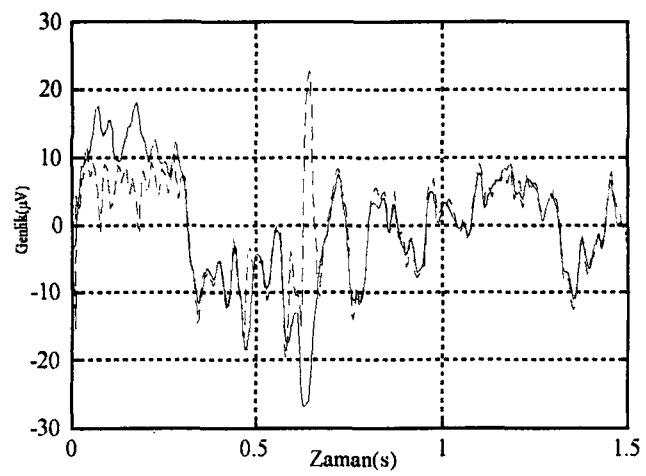
- a) 1. segment için giriş işareti (düz çizgi) ve onun kestirimi(kesikli çizgi)
- b) 2.segment için hata (UP)
- c) 2. segment için giriş işareti (düz çizgi) ve onun kestirimi (kesikli çizgi)
- d) 2.segment için hata (UP)
- e) İlk 10 segment işaret ortalaması (kesikli çizgi) , 3 hata ortalaması (düz çizgi)



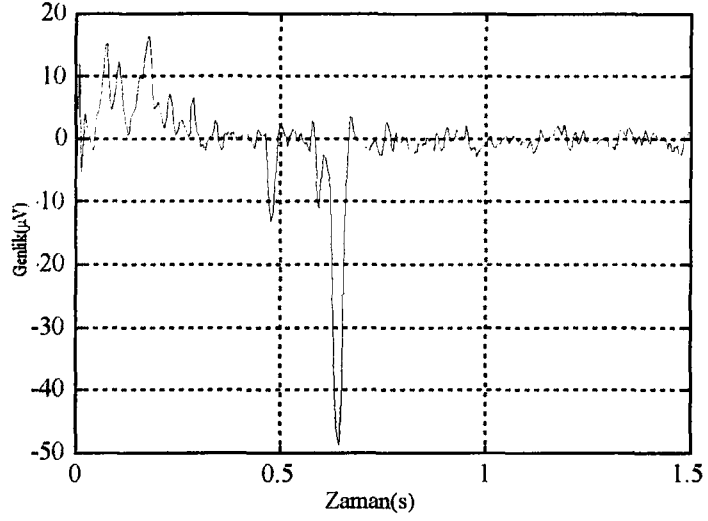
(a)



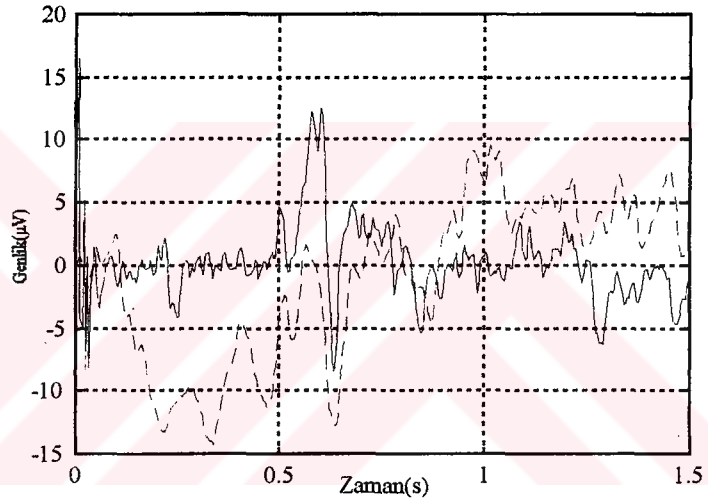
(b)



(c)



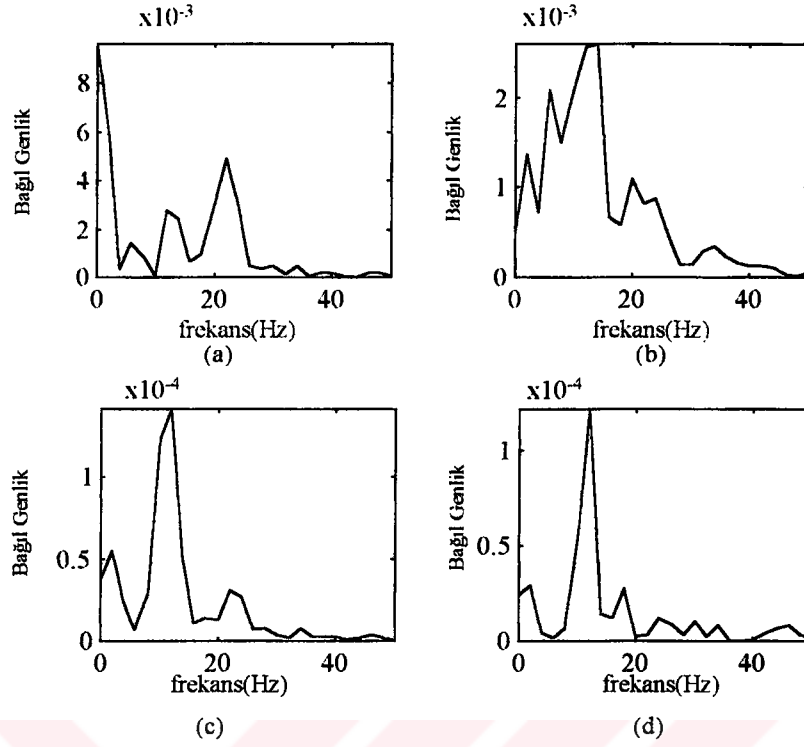
(d)



(e)

Şekil 8.8 Gerçek işaret için gerçek zamanlı ağ ile UP çıkartımı sonuçları

1. segment için giriş işareti (düz çizgi) ve onun kestirimi(kesikli çizgi)
- 1.segment için hata (UP)
2. segment için giriş işareti (düz çizgi) ve onun kestirimi (kesikli çizgi)
- 2.segment için hata (UP)
- İlk 10 segment işaret ortalaması (kesikli çizgi) , 3 hata ortalaması (düz çizgi)



Şekil 8.9 Tek segmentte UP çıkartımı için frekans düzlemi sonuçları

- Bir segment işlenmemiş işaretin periyodogramı
- 100 segment işaret ortalamasının periyodogramı
- Çok katmanlı ağ ile bir segment UP çıkartımı için periyodogram
- Gerçek zaman geri dönüşümlü ağ ile bir segment UP çıkartımı için periyodogram

8.5. Sonuçların Değerlendirilmesi

Tez Çalışmasında ,uyarı öncesi EEG işareti kullanılarak eğitilen ağlar ile EEG'ye ait NAR modeli elde edilmiş ve bu model ile uyarı sonrası EEG aktivitesi kestirilmiştir. Şekil 8.5, 8.6, 8.7 ve 8.8'de görülebileceği gibi uyarı sonrası EEG her iki yapay sinir ağı ile başarılı bir şekilde kestirilmiş ve kestirim hataları UP olarak elde edilmiştir. EEG'nin modellenmesinde kullanılan Gerçek Zaman Geri Dönüşümlü Ağ dinamik yapısından dolayı , Çok Katmanlı Ağa göre ani değişimleri daha iyi takip etmiştir. Her iki ağın kullanımı ile elde edilen sonuçlar klasik ortalama alma ile karşılaştırılmıştır ve ortalama almaya göre oldukça iyi sonuçlar gözlenmiştir. Bir segmentlik veri içerisindeki UP cevabı gecikmeler ve genlik değerleri açısından değerlendirildiğinde

pek fazla bilgi içermediği görülmüş ve bir kaç segment çıkartım sonucunun grup ortalamasının alınmasının faydalı olabileceği düşünülmüştür.

Şekil 8.9.a'da görüldüğü gibi EEG işareti içine karışmış olan UP spectrumda bağıl genlik açısından oldukça zayıf durumdadırlar. Şekil 8.9 a,b,c'de çeşitli yöntemlere göre UP çıkartım sonuçlarının peryodogramları elde edilmiştir. 10-15Hz civarındaki UP'lerin kullanılan YSA yaklaşımları ile bir segment için daha belirgin bir olarak ortaya çıkartılmıştır. Zaman düzleminde olduğu gibi Gerçek Zamanlı YSA yaklaşımı şekil 8.9.c'de belirgin bir şekilde görülebileceği gibi süregelen EEG aktivitesini oldukça zayıflatmıştır.



KAYNAKLAR

1. Akay, M.,1994.Biomedical Signal Processing. ACADEMIC PRESS, 90-11, New York
2. Chen, S.,Billings, S.A., Grant,P.M.,1990. Non-Linear System Identification Using Neural Networks. International Journal Of Control,Vol.51,No. 6, 1191-1214
3. Chistopher, A.,Nitish,V. and Thador, V.,1989. Adaptive Fourier Estimation of Time Varying Evoked Potentials. IEEE Transaction on Biomedical Engineering, Vol.36, No. 4, 448-455
4. Cottrel, M.,Gigard, B.,Gigard, Y., Mangeas, M.and Muller, C.,1995 Neural Modeling for Time-Series:A Statistical Stepwise Method for Eight Elimination. IEEE Transaction on Neural Networks, Vol.6, No.6,1335-1364
5. Freeman, J.A. And Skapura, D.M., 1992. Neural Networks: Algorithms, Application, Programming Techniques. Addison Wesley Pub. Comp. ,45-125,New York
6. Gülçür,H.Ö. ve Masoud, M.,1994. Uyarılmış Beyin Potansiyeli Kestiriminde Yapay Sinir Ağı Kullanımı. BİYOMUD-94 (Biyomedikal Mühendisliği ulusal toplantısı bildiriler Kitabı),7-11,Bebek,İstanbul
7. Haykin, S.,1994. Neural Networks a Comprehensive Foundation. Macmilman Collage Pub. Camp.,138-205, New York
8. Haykin,S.,1996.Neural Networks Expand SP's Horizons. IEEE Signal Processing Magazine, March 1053-5088
9. Haykin, S.,1996. Adaptive Filter Theory, Prentice Hall Pub. Comp, 40-155,New York
- 10.Hazaraki, N. Tsoi,A.C. And Sergejev, A.A, 1997. Nonlinear Considerations in EEG Signal Clasification. IEEE Transaction on Signal Processing , Vol.45, No.4, 829-836
- 11.Hertz, J., Krong, A. And Palmer G.L.,1991,- Introduction to the theory of neural computation. Eddison Wesley Pub. Comp.,New York

12. Khanna, T., 1989. Foundations of Neural Networks. Eddison Wesley Pub. Comp., 48-65, New York
13. Lang, K.J. and Hilton, G.E., 1988. The Development of The Time-Delay Neural Network Architecture for Speech Recognition, Technical Report. 88,152, Carnegie Mellon University, Pittsburg
14. Lennart, L., 1987. System Identification Theory for user. Prentice Hall Pub. Comp., 1-150, New Jersey
15. McDonnell, J. and Waagen, D., 1994. Evolving Recurrents for Time-Series Modeling. IEEE Transaction On Neural Networks, Vol.5, No.1, 24-38
16. Mcgillem, C.D. and Aunon. D.G., 1977. Measurements of Signal Components in single Visually Evoked Brain Potential. IEEE Transaction on Biomedical Engineering, Vol.9, 233-241
17. Narendra, K. And Parthasarathy, K., 1990. Identification and Control of Dynamical Systems Using Neural Networks. IEEE Transaction On Neural Networks, Vol.1, No.1, 1-25
18. Orfanidis, J., 1988. Optimum Signal Processing. McGraw HILL Pub. Comp., 68-76, New York.
19. Waibel, A., Hanazava, T., Hilton, G., Shikano, K. and Lang, K.J., 1989. Phoneme Recognition Using Time-Delay Neural Networks . IEEE Transaction On Acoustics Speech And Signal Processing , 37, 328-339.
20. Wan, E.A., 1990. Temporal Backprobagation for FIR Neural Networks. IEEE International Joint Conference on Neural Networks, Vol. 1, 575-580, San Diego
21. Williams, R.J., Zipser, D., 1989 A Learning Algorithm for Continually Running Fully Recurrent Neural Networks. Neural Computation, 1, 490-501
22. Yazgan, E. ve Korürek, M., 1996. Tıp Elektroniği, İstanbul Teknik Ün Yay., Sayı no. 1574, İstanbul
23. Yu, X., He, Z. And Zhang, Y., 1994. Time Varying Adaptive Filters for Evoked Potential Estimation. IEEE Transaction on Biomedical Engineering , Vol.41, No.11, 1062-1071

24. Zhang, J. And Zheng, C.,1997. Extracting Evoked Potential with the Singularity Detdection Tecnique. IEEE Engineering in Medicine and Biology, 0739-5175,155-160



EK A: PRAGRAM LİSTELERİ

Bu ekte, UP çıkartımı için MATLAB ile yazılmış çok katmanlı ve gerçek-zaman geri-dönüşümlü ağ için eğitim programı listeleri gözükmetedir.

Çok katmanlı ağ için geri yansıtma algoritması

```
clear
clg;
initalize='Y';
load cn51.pz;
y=zeros(375,1);
for i=0:99
    y=y+cn51(375*i+1:375*i+375);
end
y=y/3000;
plot(y);
epoch=240;
N=size(y,1);
D_Y=24;
D_E=3;
lr=0.05;
gizli_boyut=60;
giris_boyut = D_Y+D_E;
[data_set]=ornekle(y,D_Y+1,D_Y);
L=size(data_set,2);
target = data_set(1,:);
data_inp = data_set(2:D_Y+1,:);
clear data_set;
SON=120-D_Y;
est=zeros(L,1);
error=zeros(L+D_E,1);
x=zeros(giris_boyut,1);
if upper(initalize)=='Y'
    ['Initialize 1 hidden layer']
    kat=0.3;
    w1=(rand(gizli_boyut,giris_boyut)-0.5)*kat;
    w2=(rand(1,gizli_boyut)-0.5)*kat;
    b1=(rand(gizli_boyut,1)-0.5)*kat;
    b2=(rand(1,1)-0.5)*kat;
end
['Trainbp 1 hidden layer']
for j=1:epoch
    for i=1:SON
        x(1:D_Y,1) = data_inp(:,i);
        if D_E>0
            x(D_Y+1:giris_boyut,1)=error(i:D_Y+1);
        end
        [gizli_v,cikis_v,gizli_y,cikis_y] = ileri(w1,w2,b1,b2,x);
        desired = target(i);
        [cikis_delta,gizli_delta]=geriyans(gizli_v,cikis_v,gizli_y,cikis_y,w2,desired);
        [w1,w2,b1,b2] = guncelle(x,gizli_y,w1,w2,b1,b2,cikis_delta,gizli_delta,lr);
        [gizli_v,cikis_v,gizli_y,cikis_y] = ileri(w1,w2,b1,b2,x);
        est(i) = cikis_y;
        error(i:D_Y+1) = desired-cikis_y;
    end
end
```



```

if rem(j,5)==0
    close (figure(1));
    figure(1)
    plot(est(1:SON),'r');
    hold on
    plot(target(1:SON));
for i=1:SON
    x(1:D_Y,1) = data_inp(:,i);
    if D_E>0
        x(D_Y+1:giris_bohut,1)=error(i:D_E-1);
    end
    [gizli_v,cikis_v,gizli_y,cikis_y] = ileri(w1,w2,b1,b2,x);
    est(i) = cikis_y;
% error(i:D_E) = desired-cikis_y;
end
close (figure(1));
figure(1)
plot(est(1:SON),'r');
hold on
plot(target(1:SON));
pause
end
end
for i=1:L
    x(1:D_Y,1) = data_inp(:,i);
    if D_E>0
        x(D_Y+1:giris_bohut,1)=error(i:D_E-1);
    end
    [gizli_v,cikis_v,gizli_y,cikis_y] = ileri(w1,w2,b1,b2,x);
    desired = target(i);
    est(i) = cikis_y;
    error(i:D_E) = desired-cikis_y;
end
figure
plot(target)
hold on
plot(est,'r')
figure
semilogy(error(D_E+1:size(error,1)).^2+1)
*****
function [data_set]=ornekle(y,gecikme,R)
N=size(y,1);
data_set=zeros(R+1,N-gecikme+1);
for i=0:R
    data_set(i+1,:)=y(gecikme-i:N-i)';
end
*****
function [gizli_v,cikis_v,gizli_y,cikis_y] = ileri(w1,w2,bias1,bias2,x)
gizli_v = w1*x-bias1;
gizli_y = 2./(1+exp(-2*gizli_v))-1;
cikis_v = w2*gizli_y-bias2;
cikis_y = 2./(1+exp(-2*cikis_v))-1;
*****
function [cikis_delta,gizli_delta]=geriyans(gizli_v,cikis_v,gizli_y,cikis_y,w2,desired)
cikis_delta = (desired-cikis_y).*(4*exp(-2*cikis_v))./(1+exp(-2*cikis_v)).^2);
gizli_delta = (w2*cikis_delta).*(4*exp(-2*gizli_v))./(1+exp(-2*gizli_v)).^2);
*****

```

```

function [w1,w2,b1,b2] = guncelle(x,gizli_y,w1,w2,b1,b2,cikis_delta,gizli_delta,lr):
w1 = w1 + lr*(gizli_delta*x');
w2 = w2 + lr*(cikis_delta*gizli_y');
b1 = b1 - lr*gizli_delta;
b2 = b2 - lr*cikis_delta;
*****

Gerçek zaman geri-dönüşümlü ağ için eğitim algoritması
*****

clear
load eeg_data;
plot(target)
inp=zeros(size(data_inp,1)+1,size(data_inp,2));
inp(1:size(data_inp,1),:)=data_inp;
inp(size(data_inp,1)+1,:)=ones(1,size(inp,2));
H=16;
lr=0.1;
epoch=1;
figure
plot(target);
M = size(inp,1);
[cikis_boyut,no_patt] = size(target);
N = cikis_boyut + H;
est=zeros(size(target));
[w] = sifirla(N,N+M,0.1);
for ep=1:epoch
    del_w = zeros(size(w));
    PIM = zeros(N,N*(N+M));
    y_h=zeros(N,1);
    for i=1:no_patt
        u=[y_h
            inp(:,i)];
        [v,y_h] = ileri(w,u);
        e = target(:,i) - y_h(1:cikis_boyut);
        [PIM]=geriyans(PIM,w,v,u);
        [del2_w] = guncelle(w,e,PIM,lr);
        w=w+del2_w;
        est(:,i) = y_h(1:cikis_boyut);
    end
end
if rem(ep,1)==0
hold on
    plot(est,'r');
    hold on
    plot(target-est,'w');
end
end
*****

function [w] = sifirla(K,L,katsayi)
w = (rand(K,L)-0.5)*2*katsayi;
*****

function [data_set]=ornekle(y,gecikme,R)
N=size(y,1);
data_set=zeros(R+1,N-gecikme+1);
for i=0:R
    data_set(i+1,:)=y(gecikme-i:N-i)';
end
*****

function [v,y_h] = ileri(w,u)

```

```

v = w*u;
y_h = 2./(1+exp(-2*v))-1;
*****
function [PIM2]=geriyans(PIM1,w,v,u)
N=size(w,1);
M=size(w,2)-N;
PIM2=zeros(size(PIM1));
turev_v = 4*exp(-2*v)/((1+exp(-2*v)).^2);
for j=1:N
    for k=1:N
        for l=1:N+M
            kl = (k-1)*(N+M)+l;
            ara = w(j,1:N)*PIM1(:,kl);
            if j==k
                ara = ara + u(l);
            end
            PIM2(j,kl) = ara*turev_v(j);
        end
    end
end
*****
function [del_w] = guncelle(w,e,PIM,lr);
N = size(w,1);
M = size(w,2) - N;
K = size(e,1);
del_w = zeros(1,size(PIM,2));
for i=1:K
    del_w = del_w + e(i)*PIM(i,:);
end
del_w = (lr/K)*reshape(del_w,N+M,N);
*****

```

EC. YÖK
 İZMİR ÜNİVERSİTESİ
 İZMİR

ÖZGEÇMİŞ

1972 yılında Karamürsel’de doğdu. İlk ve orta öğrenimini Karamürsel’de tamamladı. 1990 yılında İzmit Teknik Lisesi, Elektronik bölümünden mezun oldu. 1994 yılında Gazi Üniversitesi, Teknik Eğitim Fakültesi, Elektronik-Bilgisayar Bölümü, Elektronik Ana Bilim dalından mezun oldu. 1995 yılında Kocaeli Üniversitesi, Fen Bilimleri Enstitüsü, Elektronik Bilgisayar Eğitimi ana bilim Dalında Yüksek Lisans öğrenimine başladı.

1995 yılından beri Kocaeli Üniversitesi ,Teknik Eğitim Fakültesi , Elektronik-Bilgisayar Eğitimi Bölümünde Araştırma Görevlisi olarak görev yapmaktadır.

T.C. YÜKSEKÖĞRETİM KURULU
DOKÜMAN YÖNETİM MERKEZİ