

**T.C
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

KAOTİK YAPAY SİNİR AĞLARININ ANALİZİ VE SİSTEM MODELLEME

YÜKSEK LİSANS TEZİ

Nida KAVAK

Anabilim Dalı: Elektronik-Bilgisayar Eğitimi

Programı: Kontrol Eğitimi

ARALIK-2009

**T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

KAOTİK YAPAY SİNİR AĞLARININ ANALİZİ VE SİSTEM MODELLEME

YÜKSEK LİSANS TEZİ

Nida KAVAK

(06131108)

Tezin Enstitüye Verildiği Tarih : 14 Aralık 2009

Tezin Savunulduğu Tarih : 25 Aralık 2009

Tez Danışmanı : Prof. Dr. Muammer GÖKBULUT

Diğer Jüri Üyeleri : Prof. Dr. Z. Hakan AKPOLAT

Yrd. Doç. Dr. Beşir DANDİL

ARALIK-2009

ÖNSÖZ

Bu çalışmanın hazırlanmasında büyük payı olan seminer yöneticim Sayın Prof. Dr. Muammer GÖKBULUT'a çalışmalarında yardımlarını esirgemeyen Öğr. Gör. Cafer BAL' a, manevi desteğini benden esirgemeyen arkadaşlarıma ve aileme teşekkür ederim.

Nida KAVAK
ELAZIĞ – 2009

İÇİNDEKİLER

	<u>Sayfa No</u>
ÖNSÖZ	I
İÇİNDEKİLER.....	II
ÖZET	IV
SUMMARY	V
ŞEKİLLER LİSTESİ	VI
SİMGELER LİSTESİ	IX
SEMBOLLER LİSTESİ	X
KISALTMALAR LİSTESİ	X
1. GİRİŞ.....	1
2. KAOS.....	5
2.1 Kaosu Ortaya Çıkaran Yöntemler	6
2.1.1 Yörüngelerin İzlenmesi Yöntemi	8
2.1.2 Faz uzayının izlenmesi yöntemi	9
2.1.3 Lyapunov üstleri	11
2.1.4 Poincare haritalama yöntemi	12
2.1.5 Güç spektrumu ile gösterim yöntemi.....	14
2.1.6 Çatallaşma diyagram yöntemi	15
2.2 Kaos Kontrolü	16
3. KAOTİK YAPAY SİNİR AĞLARI	18
3.1. Yapay Sinir Ağları	18
3.1.1 Yapay Hücre Modeli	19
3.1.2 Yapay Sinir Ağ Yapıları	20
3.1.2.1 İleri Beslemeli Yapay Sinir Ağları (İBYSA)	21
3.1.2.2. Geri Beslemeli Yapay Sinir Ağları (GBYSA)	22
3.1.3 Geriye Yayılım Öğrenme Algoritması	24
3.2 Kaotik Hücre.....	27
3.2.1. Birinci Kaotik Hücrenin Kaotik Davranışının Analizi	29
3.2.1.1 Yörüngelerin İzlenmesi Yöntemi	30
3.2.1.2 Lyapunov üstleri	32
3.2.1.3 Poincare haritalama yöntemi	33
3.2.1.4 Güç spektrumu ile gösterim yöntemi.....	34
3.2.1.5 Çatallaşma diyagram yöntemi	35
3.2.2 İkinci Kaotik Hücrenin Kaotik Davranışının Analizi.....	36
3.2.2.1 Yörüngelerin İzlenmesi Yöntemi	38

	<u>Sayfa No</u>
3.2.2.2 Lyapunov üstleri	39
3.2.2.3 Poincare haritalama yöntemi	40
3.2.2.4 Güç spektrumu ile gösterim yöntemi.....	40
3.2.2.5 Çatallaşma diyagram yöntemi	41
3.3 Kaotik Yapay Sinir Ağları.....	42
4. YAPAY SİNİR AĞLARI İLE SİSTEM MODELLEME	43
4.1 Düz modelleme.....	43
4.2 YSA ile sistem modellemede kullanılan örnek sistemler	45
4.2.1 Tek Giriş - Tek Çıkışlı Sistem Örneği	45
4.2.2 Çok Giriş - Çok Çıkışlı Sistem Örneği.....	46
4.3.1 Sistem modellemede kullanılan YSA yapıları	48
4.3.1.1 İleri beslemeli YSA	48
4.3.2 Öz Geri Beslemeli Yapay Sinir Ağları	51
4.3.3 Tam Geri Beslemeli Yapay Sinir Ağı	55
4.3.4 Kaotik Yapay Sinir Ağı	59
5. SİSTEM MODELLEME İLE İLGİLİ SONUÇLAR.....	64
5.1 Tek Giriş- Tek Çıkışlı Sistem Örneğinin YSA ile Modellenme Sonuçları	64
5.1.1 Düz İleri Beslemeli YSA İle Modelleme Sonuçları	64
5.1.2 Öz Geri Beslemeli YSA İle Sistem Modelleme Sonuçları.....	67
5.1.3. Tam Geri Beslemeli Yapay Sinir Ağı İle Modellenmesi	69
5.1.4 Kaotik Yapay Sinir Ağı İle Modellenmesi	71
5.2 Çok Giriş- Çok Çıkışlı Sistem Örneğinin YSA ile Modellenme Sonuçları	74
5.2.1 Düz İleri Beslemeli YSA İle Sistem Modelleme Sonuçları	74
5.2.2 Öz Geri Beslemeli YSA İle Sistem Modelleme Sonuçları.....	77
5.2.3. Tam Geri Beslemeli Yapay Sinir Ağı İle Modellenmesi	78
5.2.4 Kaotik Yapay Sinir Ağı İle Modellenmesi	80
6. SONUÇLAR.....	84
KAYNAKLAR.....	85

ÖZET

Yüksek Lisans Tezi

KAOTİK YAPAY SİNİR AĞLARININ ANALİZİ VE SİSTEM MODELLEME

Nida KAVAK

Fırat Üniversitesi

Fen Bilimleri Enstitüsü

Elektronik-Bilgisayar Eğitimi Anabilim Dalı

2009, Sayfa: 99

Kaos, doğrusal olmayan dinamik sistemlerde ortaya çıkan karmaşık ve zamanla tahmin edilemeyen davranışlar olarak tanımlanabilir. Biyolojik hücreler genellikle kaotik karakteristiklere sahiptir ve son yıllarda biyolojik hücrelerin kaotik davranışları modellenmeye çalışılmaktadır. Kaotik yapay hücrelerin geçici rejim karakteristikleri, yerel minimum probleminin çözümü için yararlı sonuçlar üretebilmektedir. Kaotik yapay sinir ağlarının, sistem modelleme ve denetimindeki performansı incelenmeye başlanmış olmakla birlikte kaotik YSA modelleri ve eğitim algoritmaları halen karmaşıktır ve ağın dinamik karakteristikleri ile öğrenme algoritmalarının daha ayrıntılı analizine ihtiyaç duyulmaktadır.

Bu tez çalışmasında, kaotik hücrelerin çeşitli yöntemlerle kaotik davranışları incelenmiştir. Tek girişli - tek çıkışlı ve çok girişli - çok çıkışlı olmak üzere iki farklı doğrusal olmayan dinamik sistemin düz ileri beslemeli, öz geri beslemeli, tam geri beslemeli ve kaotik yapay sinir ağları ile modelleme performansı belirlenmiştir. Kaotik YSA modellerin, ileri beslemeli YSA modellere göre daha az girişe ihtiyaç duymakla birlikte sistem modellemede etkili olduğu belirlenmiştir.

Anahatar Kelimeler: Kaotik sistemler, kaotik hücre, kaotik yapay sinir ağları, sistem modelleme.

SUMMARY
Master Thesis

**ANALYSIS OF CHAOTIC NEURAL NETWORKS AND
SYSTEM IDENTIFICATION**

Nida KAVAK

Firat University
Graduate School of Natural Applied Sciences
Department of Electronics and Computer Education

2009, page: 99

Chaos can be defined as a complex and unpredictable phenomenon which occurs in nonlinear dynamic systems. Biological neurons generally have chaotic characteristics and, in recent years, the chaotic response of chaotic neurons has been studied. Transient response of chaotic neurons could be beneficial to overcome the local minimum problem. In the last decade, identification and control of nonlinear dynamic systems using chaotic neural networks were studied however, chaotic neural network models are highly complex and the detailed analysis of dynamic characteristic of the network and training algorithms is required.

In this thesis, chaotic responses of the chaotic neurons are determined using the various methods. Identification performance of the various neural network models such as feedforward, self feedback, fully recurrent and chaotic neural network is determined for two different nonlinear dynamic systems which are single input-single output and multiple inputs-multiple outputs. Chaotic neural network model needs less input than the feed forward neural model and it is shown that they can be effectively used for system modeling.

Keywords: Chaotic systems, chaotic neuron, chaotic neural network, system identification.

ŞEKİLLER

	<u>Sayfa No</u>
Şekil 2.1 Basit sarkacın resmi.....	6
Şekil 2.2 Sarkacın Periyodik davranışını yörüngenin izlenmesi yöntemi ile gösteren ω -t grafiği	8
Şekil 2.3 Sarkacın kaotik davranışını yörüngenin izlenmesi yöntemi ile gösteren ω -t grafiği	9
Şekil 2.4 Sarkacın periyodik davranışını faz uzayının izlenmesi yöntemi ile gösteren θ - ω grafiği.....	10
Şekil 2.5 Sarkacın kaotik davranışını faz uzayının izlenmesi yöntemi ile gösteren θ - ω grafiği	10
Şekil 2.6 Sönümlü ve sinüzoidal olarak sürülen sarkaçta $g=1$ için 3 tane Lyapunov üstlerinin grafiği.....	11
Şekil 2.7 Sönümlü ve sinüzoidal olarak sürülen sarkaçta $g=1.2$ için 3 tane Lyapunov üstlerinin grafiği.....	12
Şekil 2.8 Sönümlü ve sinüzoidal olarak sürülen sarkaçta $g=1$ için poicare grafiği	13
Şekil 2.9 Sönümlü ve sinüzoidal olarak sürülen sarkaçta $g=1.2$ için poicare grafiği	13
Şekil 2.10 Sönümlü ve sinüzoidal olarak sürülen sarkaçta $g=1$ için güç spektrumu grafiği.....	14
Şekil 2.11 Sönümlü ve sinüzoidal olarak sürülen sarkaçta $g=1.2$ için güç spektrumu grafiği.....	15
Şekil 2.12 Sönümlü ve sinüzoidal olarak sürülen sarkacın çatallaşma diyagramı grafiği.....	16
Şekil 2.13 Henon haritanın kaos kontrolü	17
Şekil 2.14 Kaos kontrol sisteminin diyagramı.....	17
Şekil 3.1 Yapay Hücre Modeli	19
Şekil 3.2 İleri beslemeli 3 katmanlı YSA	21
Şekil 3.3 Geri beslemeli iki katmanlı YSA	23
Şekil 3.4 İleri beslemeli 3 katmanlı YSA sinyal akış şeması	24
Şekil 3.5 Kaotik nöron birimi	28
Şekil 3.6 Kaotik dinamik hücre birimi	29
Şekil 3.7 (a) $A=0.2$ (b) $A=0.5$ (c) $A=0.7$ (d) $=0.9$ $x-n$ grafiği	31

Şekil 3.8 (a) $A=0.2$ (b) $A=0.5$ (c) $A=0.7$ (d) $A=0.9$ $y-n$ grafiği	32
Şekil 3.9 (a) $A=0.2$ (b) $A=0.5$ (c) $A=0.7$ (d) $A=0.9$ Lyapunov üstlerinin grafiği	33
Şekil 3.10 (a) $A=0.2$ (b) $A=0.5$ (c) $A=0.7$ (d) $A=0.9$ Poincare grafiği	34
Şekil 3.11 (a) $A=0.2$ (b) $A=0.5$ (c) $A=0.7$ (d) $A=0.9$ Güç spektrumu grafiği.....	35
Şekil 3.12 Kaotik hücrenin çatallaşma diyagramı grafiği	36
Şekil 3.13 Kaotik hücre birimi.....	36
Şekil 3.14 (a) $a=1$ (b) $a=2$ $x-n$ grafiği	38
Şekil 3.15 (a) $a=1$ (b) $a=2$ $y-n$ grafiği	39
Şekil 3.16 (a) $a=1$ (b) $a=2$ Lyapunov üstlerinin grafiği	39
Şekil 3.17 (a) $a=1$ (b) $a=2$ Poincare grafiği	40
Şekil 3.18 (a) $a=1$ (b) $a=2$ Güç spektrumu grafiği.....	41
Şekil 3.19 Kaotik hücrenin çatallaşma diyagramı grafiği	41
Şekil 4.1 Yapay sinir ağı ile düz modelleme(NARX model)	44
Şekil 4.2 Yapay sinir ağı ile düz modelleme (NOE model)	44
Şekil 4.3 Birinci örnek sistemin grafiği	45
Şekil 4.4 YSA ile örnek bir sistemin modellemesi	46
Şekil 4.5 İkinci örnek sistemin grafiği.....	47
Şekil 4.6 YSA ile örnek bir sistemin modellemesi	47
Şekil 4.7 Düz ileri beslemeli yapay sinir ağının yapısı	48
Şekil 4.8 Öz geri beslemeli yapay sinir ağının yapısı.....	52
Şekil 4.9 Tam geri beslemeli yapay sinir ağının yapısı	55
Şekil 4.10 Kaotik yapay sinir ağının yapısı	59
Şekil 5.1 (a) örneksel (b) toplu öğrenme yöntemi ile eğitilen düz ileri beslemeli YSA'nın sistem modelleme performansı	65
Şekil 5.2 (a) örneksel (b) toplu öğrenme yöntemi ile eğitilen düz ileri beslemeli YSA'nın sistem modellenmesinin toplam karesel hatanın değişimi	66
Şekil 5.3 (a) örneksel (b) toplu öğrenme yöntemi ile eğitilmiş düz ileri beslemeli YSA'nın kare dalga giriş sinyali için test performansı	66
Şekil 5.4 (a) örneksel (b) toplu öğrenme yöntemi ile eğitilen öz geri beslemeli YSA'nın sistem modelleme performansı	67
Şekil 5.5 (a) örneksel (b) toplu öğrenme yöntemi ile eğitilen öz geri beslemeli YSA'nın sistem modellenmesinin toplam karesel hatanın değişimi	68

Şekil 5.6 (a) örneksel (b) toplu öğrenme yöntemi ile eğitilmiş öz geri beslemeli YSA'nın	
$u = \sin(2.k.\pi / 25) + \cos(3.k.\pi / 15)$ giriş sinyali için test performansı.....	69
Şekil 5.7 (a) örneksel (b) toplu öğrenme yöntemi ile eğitilen tam geri beslemeli YSA'nın	
sistem modelleme performansı	70
Şekil 5.8 (a) örneksel (b) toplu öğrenme yöntemi ile eğitilen tam geri YSA'nın sistem	
modellenmesinin toplam karesel hatanın değişimi	71
Şekil 5.9 (a) örneksel (b) toplu öğrenme yöntemi ile eğitilen kaotik YSA'nın sistem	
modelleme performansı	72
Şekil 5.10 (a) örneksel (b) toplu öğrenme yöntemi ile eğitilen kaotik YSA'nın sistem	
modellenmesinin toplam karesel hatanın değişimi	73
Şekil 5.11 (a) toplu öğrenme 1 iterasyon (b) toplu öğrenme 10 iterasyon için eğitilmiş	
kaotik YSA'nın kare dalga giriş sinyali için test performansı	74
Şekil 5.12 (a) örneksel (b) toplu öğrenme yöntemi ile eğitilen düz ileri beslemeli YSA'nın	
sistem modelleme performansı	75
Şekil 5.13 (a) örneksel (b) toplu öğrenme yöntemi ile eğitilen düz ileri beslemeli YSA'nın	
sistem modellenmesinin toplam karesel hatanın değişimi	76
Şekil 5.14 (a) örneksel (b) toplu öğrenme yöntemi ile eğitilmiş düz ileri beslemeli	
YSA'nın kare dalga giriş sinyali için test performansı	76
Şekil 5.15 (a) örneksel (b) toplu öğrenme yöntemi ile eğitilen öz geri beslemeli YSA'nın	
sistem modelleme performansı	77
Şekil 5.16 (a) örneksel (b) toplu öğrenme yöntemi ile eğitilen öz geri beslemeli YSA'nın	
sistem modellenmesinin toplam karesel hatanın değişimi	78
Şekil 5.17 (a) örneksel (b) toplu öğrenme yöntemi ile eğitilen tam geri beslemeli YSA'nın	
sistem modelleme performansı	79
Şekil 5.18 (a) örneksel (b) toplu öğrenme yöntemi ile eğitilen tam geri beslemeli YSA'nın	
sistem modellenmesinin toplam karesel hatanın değişimi	80
Şekil 5.19 (a) örneksel (b) toplu öğrenme yöntemi ile eğitilen kaotik YSA'nın sistem	
modelleme performansı	81
Şekil 5.20 (a) örneksel (b) toplu öğrenme yöntemi ile eğitilen kaotik YSA'nın sistem	
modellenmesinin toplam karesel hatanın değişimi	82
Şekil 5.21 (a) toplu öğrenme 1 iterasyon (b) toplu öğrenme 200 iterasyon için eğitilmiş	
kaotik YSA'nın kare dalga giriş sinyali için test performansı	83

SEMBOLLER LİSTESİ

m	: Sarkacın kütleini	KG
l	: Sarkacın uzunluğunu	M
W	: Sarkacın ağırlığını	G
ω_D	: Süren kuvvetin açısal hızını	Rad/s
θ	: Sarkacın açısal konumunu	Radyan
ω	: Sarkacın açısal hızını göstermektedir	Radyan/sn
q	: Sönüm faktörü	N.sn/m
g	: Süren kuvvetin genliği	M
ω_D	: Sürme frekansdır	Radyan
W	: Hücrenin ağırlıklar matrisi	
y	: Hücre çıkışı	
r	: Giriş vektör sayısı	
p	: Ağırlık çıkışından gelen gecikmişlerin sayısı	
ΔW	: Ağırlıklara uygulanacak düzeltme miktarı	
y_d	: Problemin çıktısı	
e	: Hata	
K	: İnatçı parametre	
A	: Aktivasyon fonksiyonu	
ε	: Sigmoid fonksiyonunun eğimi	
$S^H(k)$	: Orta katmandaki girişlerin toplam ağırlığıdır	
W^I	: Giriş katmanının ağırlık vektörü	
$f_N(.)$	: Doğrusal olmayan sigmoid fonksiyon	
W^O	: Çıkış katmanının ağırlık vektörü	
S_j^O	: Çıkış katmandaki girişlerin toplam ağırlığıdır	
u	: Hücre girişi	
W^{IR}	: Orta katmandaki hücreler arası ağırlık vektörü	
S_j^H	: Orta katmandaki girişlerin toplam ağırlığıdır	

W^O	: Çıkış katmanını ağırlık vektörü
W^{OR}	: Çıkış katmanındaki hücreler arası ağırlık vektörü
S_j^O	: Çıkış katmandaki girişlerin toplam ağırlığıdır
a	: Öğrenme oranı.
∇_w	: Ağırlıklara uygulanacak düzeltme miktarı
$E(k)$	: Hata indeksi

KISALTMALAR LİSTESİ

KYSA : Kaotik yapay sinir ağı
YSA : Yapay sinir ağı

1. GİRİŞ

Doğrusal olmayan sistemlerin zamanla tahmin edilemeyen davranış göstermesi *kaos* olarak adlandırılmıştır. Kaos teorisi bilimsel bir disiplin olarak 1960'lı yıllarda Edward Lorenz'in hava tahmini için topladığı verileri kullanarak meteorolojik sistemleri Klasik Lorenz Eşitlikleri ile bilgisayar ortamında modellemeye çalışması ile ortaya çıkmıştır [1]. Edward Lorenz, bilgisayarın bulduğu sıcaklık değerlerini bir grafikte gösteren hava tahmini programı hazırlamıştır. Programında birbirine çok yakın farklı başlangıç değerlerinin, zamanla birbirinden uzaklaştığını fark etmiştir. Başlangıç şartlarının önemini gösteren, ilk bilimsel çalışmayı Lorenz yapmıştır.

Zamanla tahmin edilemeyen davranış gösteren sistemler, *kaotik sistemler* olarak adlandırılmıştır. Kaotik sistem, ilk koşullara büyük hassasiyetle bağımlı bir sistemdir [2]. Birbirine çok yakın iki noktadan başlayan yörüngeler, daha sonra aynı yönde gitmeleri gerekirken, kaotik sistemlerde bu yörüngeler zamanla uzaklaşmakta ve kestirimi imkânsız hale gelmektedir. Bir sistemin kaotik bir davranış göstermesi dış etkenlere bağlı değildir, sistemin kendi iç dinamiği ve başlangıç şartları ile ilgilidir. Kaotik sistemler ve kaotik işaretler;

- Zaman boyutunda düzensiz davranış gösterirler,
- Başlangıç şartlarına hassas bir duyarlılık gösterirler,
- Sınırsız sayıda değişik periyodik salınımlar ihtiva ederler,
- Gürültü benzeri geniş güç spektrumuna sahiptirler,
- Genliği ve frekansı tespit edilemeyen ancak sınırlı bir alanda değişen işaretler ihtiva ederler [3].

Kaosun en belirgin özelliği olan başlangıç şartlarına bağımlılığı, ilk olarak Poincaré ortaya atmıştır. Henri Poincaré bu olguyu şöyle açıklamıştır: '...başlangıç şartlarındaki çok küçük farklar, sonuç olgusunda çok büyük farklar meydana getirmektedir. Başlangıçtaki küçük hata, daha sonra devasa bir hatayı meydana getirmektedir. Kestirim imkânsız hale gelmekte ve rastlantısal bir olguya sahip olmaktadır [4].

Kaos, literatürde karmaşa ve kargaşalık olarak kullanılmasına rağmen, kaotik sistemler karmaşıklık veya kargaşa olan sistemler değildirler. Karmaşık olan sistemler hiçbir durumda davranışları kontrol altına alınamayan sistemlerdir, çünkü başlangıç koşullarına bağlı değildirler. Halbuki kaotik sistemler kontrol altına alınabilen sistemlerdir.

Bilim dünyasında yüzyıllarca doğanın determinist (gerekirci) olduğu düşüncesi yaygındı. Eğer bir doğa olayı matematiksel olarak modellenilebilirse, basit bir neden sonuç ilişkisi ile gelecekteki durumunun her zaman tahmin edilebileceği düşünülüyordu. Kaotik sistemlerde ise sistemin gelecekteki durumu tahmin edilemez. Karmaşık sistemlerin doğada ve günlük hayatta örnekleri mevcuttur. Doğada karmakarışık davranışın gözlemleri, güneş sisteminde uyduların dinamiği, gökle ilgili maddelerin manyetik alanının zaman gelişimi, çevrebilimde nüfus büyümesi, hareket potansiyellerinin dinamiği ve moleküler titreşimleri kapsar. Karmakarışık sistemlerin günlük örnekleri hava ve iklimi kapsar [5].

Günümüzde bilim adamları, kaotik davranışın, bilimin tüm alanlarındaki deneylerde ve bilgisayar modellerinde gözlenebileceğini kavramışlardır. Gerekli olan, sistemin doğrusal olmamasıdır. Şimdiye kadar yapılmış olan deneylerdeki gürültü veya hata olarak adlandırılan davranışların, kaos terimi ortaya çıktıktan sonra tekrar değerlendirilmesi günümüzde yaygın bir şekilde yapılmaktadır [6]. Doğrusal olmayan sistemlerde kaosun aranması için bazı şartlar vardır. Sürekli zamanlı bir sistemde bu şart sistemin derecesinin üç olması ve sistemin hareket denklemlerinde doğrusal olmayan eleman içermesidir. Ayrık zamanlı bir sistemde ise sistemin doğrusal olmaması kaosun aranması için yeterlidir. Şartlar sağlansa bile sistemin kaotik davranış göstereceği kesin değildir. Sistem kaotik davranış gösterebilir veya göstermeyebilir. Kaosun gösterilmesi için farklı yöntemler vardır [7].

İnsan beyninin çalışma prensibini yapay olarak modellemeyi amaçlayan Yapay Sinir Ağları (YSA); nesne/görüntü tanıma, sinyal işleme, arıza analizi ve tespiti, sistem tanılama (modelleme) ve denetimi ve bunun gibi çeşitli alanlarda yaygın olarak kullanılmaya başlanmış ve kullanıldığı alanlardaki problemlerin çözümüne yeni yaklaşımlar getirmiştir [8]. Hayvanlar üzerinde yapılan elektro fizyolojik deneyler, kaos dinamiğinin gerçek nöronlarda var olduğunu ve sinirsel ağların nöron faaliyetinde önemli bir rolü oynadığını kanıtlamıştır [9]. Kaotik dinamiklerin nöronlar ve yapay sinir ağlarında var olduğu ve belirli işlevler oynadığı bildirilmiştir [10].

Genellikle biyolojik hücreler geri beslemelidir ve geçici veya kalıcı rejimde kaotik karakteristiklere sahiptir. Kaotik nöronların bir araya getirilmesiyle oluşturulan yapay sinir ağları *Kaotik Yapay Sinir Ağları* olarak adlandırılmıştır. Kaotik nöronlar, harici girişler ile diğer kaotik nöronlardan gelen geri besleme girişlerinin toplamının aktivasyon fonksiyonundan çıktısıdır. Bazı değişiklikler yapılsa bile önceden hazırlanan bu kaotik

nöron modellerini sinir ağına uygulamak hala karmaşıktır ve nöron kendisi ve öğrenme algoritmasında daha fazla dinamik karakteristiğe ihtiyaç duymaktadır [11]. Son yıllarda sinir ağlarının kaotik dinamiğinin incelenmesi etkin bir alan olmuştur. Bu incelemelerin birçoğu modellerin dinamik yapısının tanımlanmasıyla ilgilidir [12,13,14]. Diğerleri ise kaotik dinamiklerin uygulamaları ve rolleri ile ilgilidir [15,16,17].

Kaotik hücrelerin ve kaotik hücrelerle oluşturulan kaotik yapay sinir ağlarının (KYSA) kaotik karakteristiklerinin, genellikle optimizasyon problemlerinin çözümü üzerinde olumsuz bir etkisi vardır. Ancak, kaotik hücrelerin geçici rejim karakteristikleri, yerel minimum probleminin çözümü için yararlı sonuçlar üretebilmektedir [18,19]. Buna rağmen, kaotik karakteristiklerin etkisi henüz analitik olarak doğrulanmamıştır. Bazı optimizasyon problemlerinin çözümü ile sistem modelleme ve sistemin adaptif kontrolüne yönelik bazı geliştirme çalışmaları yapılsa da kaotik hücre ve YSA modelleri halen karmaşıktır, bu nedenle hücrenin dinamik karakteristikleri ile öğrenme algoritmalarının daha ayrıntılı analizine ihtiyaç duyulmaktadır [11]. Diğer taraftan, KYSA'lardan uygulamada yararlanılabilmesi için ağı kaotik yörüngesinin uygun bir yöntemle kontrol edilmesi ve bir denge noktasında kararlı hale getirilmesi önemlidir. Son zamanlarda yapılan çalışmalar kaosu kontrolü üzerinedir. Kaos sisteminin kontrolü, kaotik sistemin parametrelerinde az bir değişiklik yaparak kaotik sistemdeki kararsız periyodik yörüngelerin dengelenmesiyle gerçekleştirilmektedir [20]. Kaos kontrolü üzerine çeşitli araştırmalar yapılmış ve düşük boyutlu kaotik sistemler için kaos senkronizasyonu ve geciktirilmiş geri beslemeli kontrol gibi kaos kontrol yöntemleri geliştirilmiştir [21,22,23]. Daha karmaşık sistemler için ise kaotik pinning kontrol yöntemi geliştirilmiştir [24].

Bu çalışmada, sürekli zamanlı bir sistemde kaotik durum için gerekli şartların ve ayrık zamanlı bir sistemdeki kaotik durum için gerekli şartların neler olduğu incelenmiştir. Bir sürekli zaman dinamik sistem örneği (Doğrusal olmayan sarkaç) kullanılarak sistemin dinamik davranışını (periyodik, kaotik vs) gösteren yöntemler açıklanmıştır. Yapay sinir ağlarının yapısı, çeşitleri ve eğitimi hakkında bilgi verilmiştir. Kaotik nöronların yapısı incelenmiş ve matematiksel denklemleri çıkartılmıştır. Sistemin davranışını gösteren yöntemlerle kaotik nöronların davranışları incelenmiştir. Kaotik hücrenin parametrelerinde değişiklik yapılarak kaotik davranış kontrol edilmiş, böylelikle kaotik nöronlar kontrol altına alınmıştır. Düz ileri beslemeli, öz geri beslemeli, tam geri beslemeli ve kaotik yapay sinir ağlarının yapısı incelenmiş, matematiksel denklemleri çıkartılmıştır. Bu yapay sinir

ağları tek giriş - tek çıkışlı ve çok giriş - çok çıkışlı olmak üzere iki farklı doğrusal olmayan dinamik sistem ile modellenme performansları gösterilmiştir.

2. KAOS

Doğrusal olmayan sistemlerin zamanla tahmin edilemeyen davranış göstermesi *Kaos* olarak adlandırılmıştır. Kaosun temel karakteristiği, sistemin geçmiş davranışını tekrar etmemesidir [25]. Dinamik sistemlerin 5 farklı durumları bulunmaktadır.

- (a) Sistemin davranışı bir denge noktasına gidebilir,
- (b) sistem periyodik bir davranış gösterebilir,
- (c) sistem periyodumsu (quasiperiodic) bir davranış gösterebilir,
- (d) sistemin cevabı sonsuza gidebilir,
- (e) sistem kaotik davranış gösterebilir [4].

Kaosun en ayırt edici özelliği, sistem aralarında çok küçük fark bulunan iki komşu noktadan başlatıldığında kolaylıkla görülebilir. Başlangıç şartlarındaki bu çok küçük farkın sadece bir ölçüm hatası meydana getireceği düşünülebilir. Örneğin, kaotik olmayan bir sistemde bu belirsizlik, sonucun tahmininde zamanla doğrusal olarak artan bir hataya neden olur. Oysa kaotik sistemler için ise bu fark, zamanla üstel olarak artar ve çok kısa bir zaman sonra sistemin durumu belirlenemez. Etkili denklemlerin doğrusal olmadığı zaman meydana gelen bu olgu başlangıç şartlarına bağımlılık olarak bilinir [4].

Doğrusal olmayan sistemlerin kaotik davranış göstermesi az rastlanılan bir durum değildir. Bir sistemin kaotik davranış göstermesi için bazı şartlar bulunmaktadır. Sürekli zamanlı bir sistemin kaotik davranış gösterebilmesi için;

- (a) sistem en az üç tane bağımsız dinamik değişkene sahip olması,
- (b) hareket denklemleri doğrusal olmayan bir terim içermesi gerekmektedir.

Denklemler genelde aşağıda verilen biçimde ifade edilirler [4].

$$dx_1/dt = F_1(x_1, x_2, \dots, x_n)$$

$$dx_2/dt = F_2(x_1, x_2, \dots, x_n)$$

.

.

$$dx_n/dt = F_n(x_1, x_2, \dots, x_n)$$

n en az 3 olmalıdır.

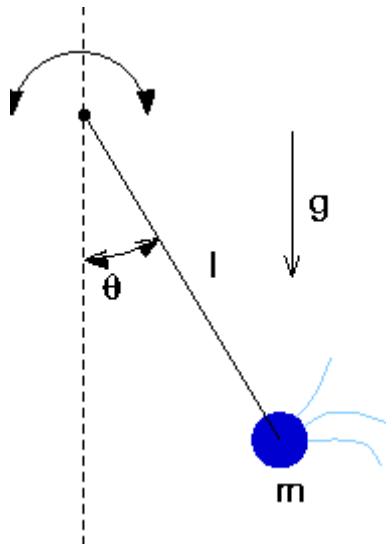
Ayrık zamanlı bir sistemde ise sistemin doğrusal olmaması şartı yeterlidir. Birinci derecede olan ayrık zamanlı bir sistemde bile kaos aranabilir. Sistemin kaos şartlarını sağlamış olması, sistemin kesin kaotik davranış göstereceği anlamına gelmeyip sadece sistemde kaotik davranış arayabileceğimizi ifade eder.

2.1 Kaosu Ortaya Çıkaran Yöntemler

Bir sistemin Kaotik olup olmadığını anlamaya yarayan birçok yöntem bulunmaktadır. Doğrusal olmayan sistemlerde kaos analizi yapmak için en çok kullanılan yöntemler şunlardır:

- Yörüngenin izlenmesi (zaman serileri)
- Faz uzayının incelenmesi (phase portrait)
- Lyapunov üstleri
- Poincare haritalama
- Güç spektrumu
- Çatallaşma (Bifurcation) diyagramı.

Kaosu ortaya çıkaran yöntemleri anlatmak için sönümlü ve sinüzoidal sarkaçtan yararlanılacaktır.



- m Sarkacın kütlesini,
- l Sarkacın uzunluğunu,
- W Sarkacın ağırlığını,
- ω_D Süren kuvvetin açısal hızını,
- θ Sarkacın açısal konumunu,
- ω Sarkacın açısal hızını göstermektedir.

Şekil 2.1 Basit sarkacın resmi

Sönümlü ve sinüzoidal sarkaca ait denklemler 2.1' de verilmiştir [4].

$$ml \frac{d^2\theta}{dt^2} + \gamma \frac{d\theta}{dt} + W \sin \theta = A \cos(\omega_D t) \quad (2.1)$$

Parametre sayısını azaltmak için denklem tekrar yazılırsa;

$$\frac{d^2\theta}{dt^2} + \left(\frac{1}{q}\right) \frac{d\theta}{dt} + \sin \theta = g \cos(\omega_D t) \quad \text{elde edilir.} \quad (2.2)$$

q Sönüm faktörü,

g Süren kuvvetin genliği,

ω_D ise sürme frekansdır.

Sarkacın düşük genlikli doğal açısal frekansı birim olarak alınmıştır.

$$\begin{aligned} d\omega/dt &= -(1/q)\omega - \sin \theta + g \cos \phi \\ d\theta/dt &= \omega \\ d\phi/dt &= \omega_D \end{aligned} \quad (2.3)$$

Denklem birinci dereceden yazılıp, kaosu aranma şartları tekrar gözden geçirilir. Sistemimiz sürekli zamanlı bir dinamik sistem olduğundan dolayı kaosu aranabilmesi için gerekli şartlar;

- (a) sistemde en az üç tane dinamik bağımsız değişken olması,
- (b) hareket denklemlerinde en az bir tane doğrusal olmayan terim bulunmasıdır.

- Sistemin üç tane bağımsız değişkeni ω, θ, ϕ 'dir.
- Sistemin doğrusal olmayan terimleri ise $\sin \theta$ ve $g \cos \phi$ 'dir
- Sistemin kaosa girip girmeyeceğini belirleyen parametreleri ise g, ω_D ve q 'dur.

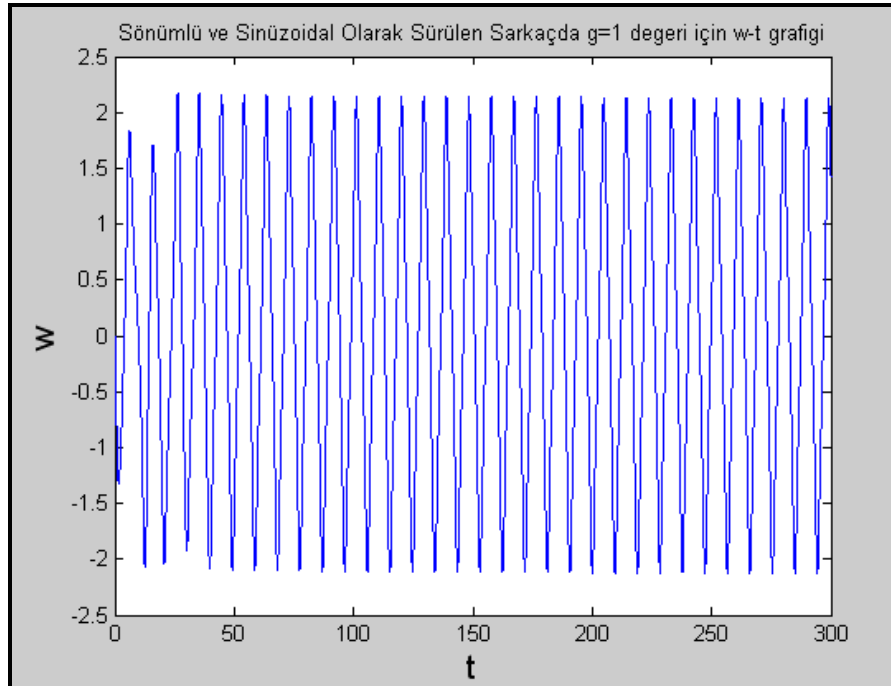
Sistem incelenirken 2 parametre sabit olarak ($\omega_D=2/3$, $q=2$) alınıp 3. parametredeki (g) değişimin, sistemin davranışını nasıl değiştirdiği incelenecektir. Denklem sistemini çözdürürken $g=1$ değeri için sistemin periyodik davranış gösterdiği,

$g=1.2$ değeri için ise kaotik davranış gösterdiği bilinmektedir [25]. Yörüngelerin izlenmesi yöntemine göre $g=1$ ve $g=1.2$ için ω - t grafiği çizilir.

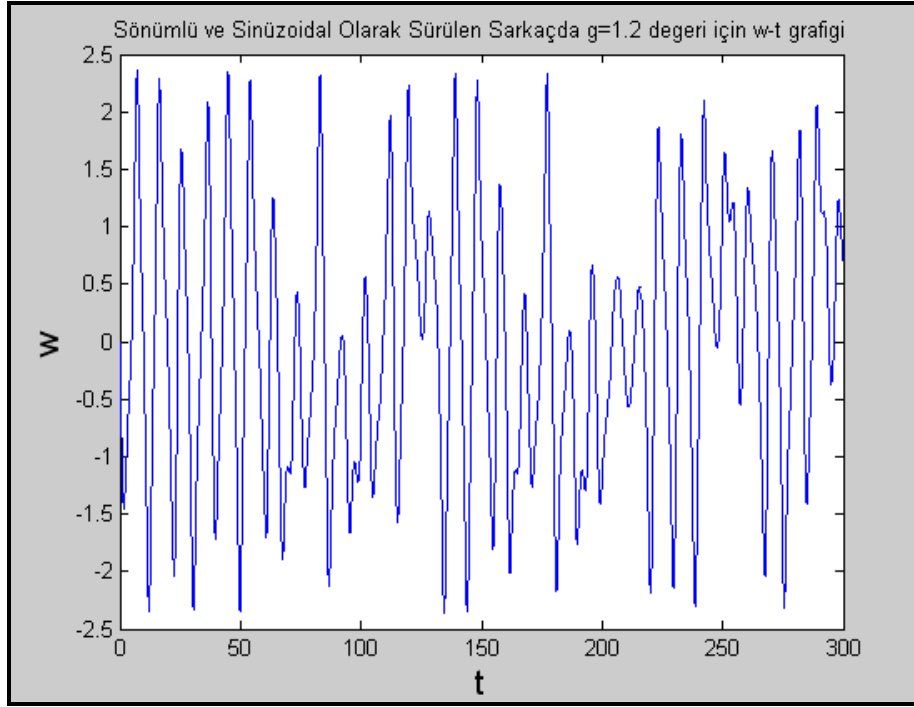
2.1.1 Yörüngelerin İzlenmesi Yöntemi

x eksenine zaman, y eksenine durum değişkenlerinden biri girilerek çizilen grafikte, sistemin kaosa girip girmediğine bakılabilir.

Şekil 2.2’de $g=1$ değeri için değişkenlerden ω ’nın zaman ekseninde grafiği çizilmiş periyodik davranışı gösterilmiştir. Şekil 2.3’de ($g=1.2$) değeri için değişkenlerden ω ’nın zaman ekseninde grafiği çizilmiş ve kaotik bir dalga şekli gösterilmiştir.



Şekil 2.2 Sarkacın Periyodik davranışını yörüngenin izlenmesi yöntemi ile gösteren ω - t grafiği

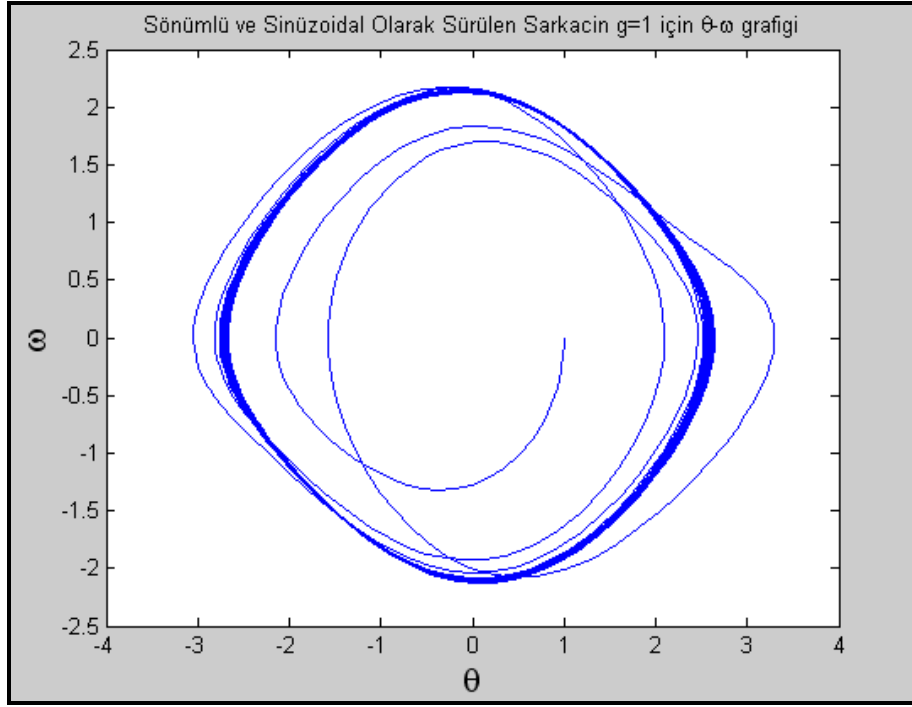


Şekil 2.3 Sarkacın kaotik davranışını yörüngenin izlenmesi yöntemi ile gösteren ω -t grafiği

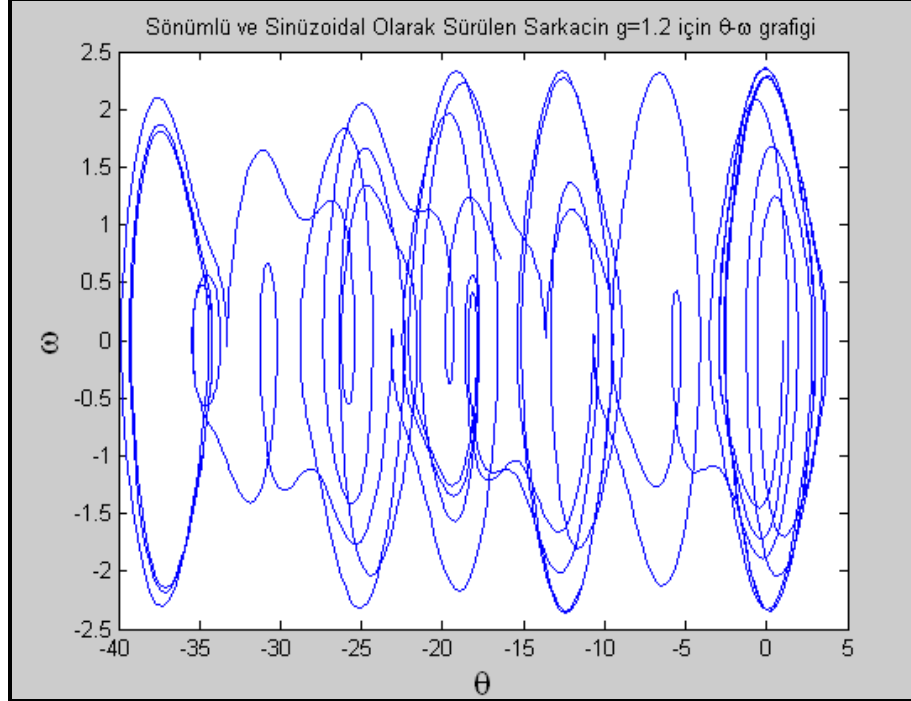
2.1.2 Faz uzayının izlenmesi yöntemi

Durum değişkenlerinin birbirine göre davranışları çizdirilir. Eğer sistem periyodik ise, grafikte kapalı bir çevrim görülür. Buna limit çevrimi de denir ve sistemin periyodik davranış gösterdiğini belirtir. Karmaşık bir şekil ortaya çıkması ise sistemin kaotik olduğunu göstermektedir.

Şekil 2.4’de sarkacın periyodik davranışını faz uzayının izlenmesi yöntemi ile gösteren ω -t grafiği izlenmektedir. Şekilde görüldüğü gibi sarkaç, periyodik bir davranış gösterdiğini belirten kapalı bir çevrim çizmiştir. Şekil 2.5’de sarkacın kaotik davranışını faz uzayının izlenmesi davranışına göre gösteren ω -t grafiği izlenmektedir. Şekilde görüldüğü gibi sistem karmaşık bir şekil çizmiştir, yani kaotik davranış göstermiştir.



Şekil 2.4 Sarkacın periyodik davranışını faz uzayının izlenmesi yöntemi ile gösteren θ - ω grafiği



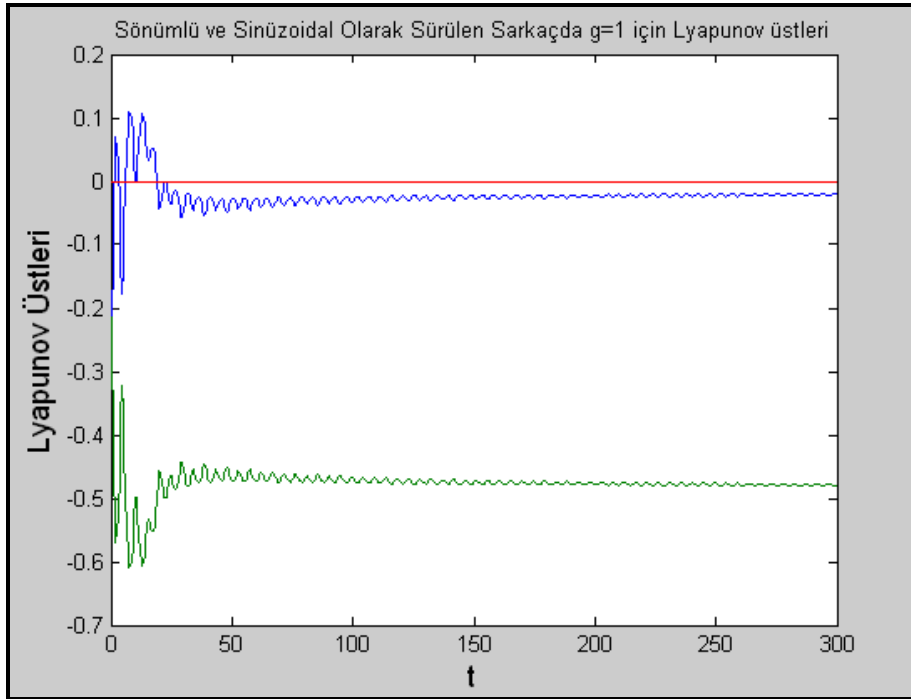
Şekil 2.5 Sarkacın kaotik davranışını faz uzayının izlenmesi yöntemi ile gösteren θ - ω grafiği

2.1.3 Lyapunov üstleri

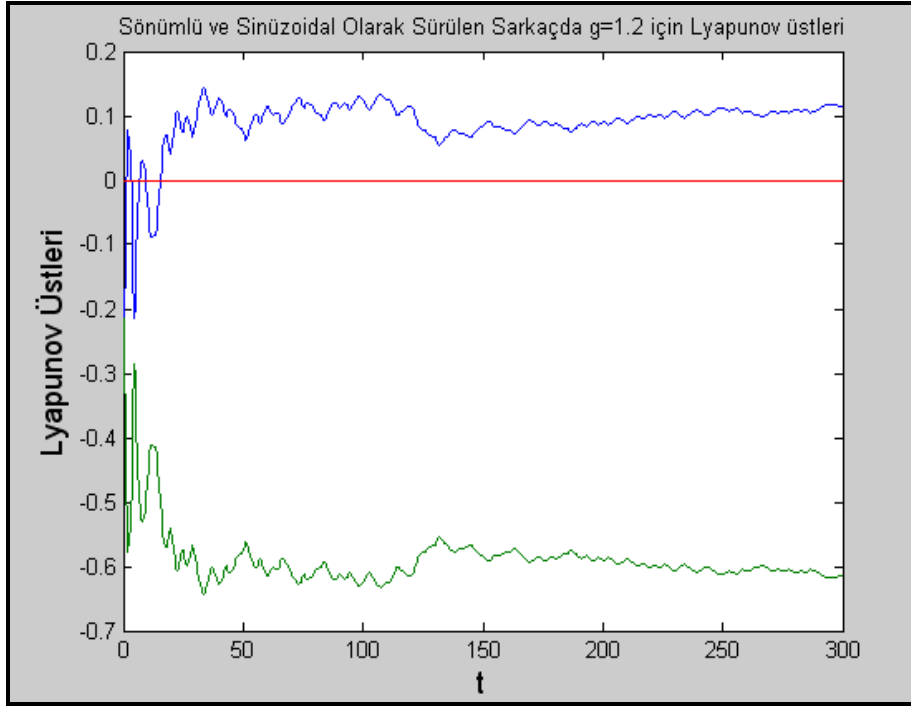
Bu yöntem, doğrusal sistemlerde kullanılan öz değerlere benzetilir ve literatürde de öz değerlerin doğrusal olmayan sistemlerdeki karşılığı olarak ifade edilmektedir [26].

Yöntem temelde, birbirine çok yakın iki noktadan başlayan yörüngelerin, birbirine hangi yönde ne kadar yaklaşacağını ve birbirinden ne kadar uzaklaşacağını niceliksel olarak gösteren bir yöntemdir. Lyapunov üstlerinin hepsi negatif yörüngede ise sistem kaotik olmayan bir davranış gösterecektir. Lyapunov üstlerinden biri pozitif yörüngede ise sistem kaotik bir davranış gösterecektir.

Şekil 2.6'da sönümlü ve sinüzoidal olarak sürülen sarkaçta $g=1$ için Lyapunov üstlerinin 3 tane grafiği çizilmiştir. Şekilde Lyapunov üstlerinin hepsi negatiftir ve sistem periyodik davranış göstermektedir. Şekil 2.7'de sönümlü ve sinüzoidal olarak sürülen sarkaçta $g=1.2$ için Lyapunov üstlerinin 3 tane grafiği çizilmiştir. Şekilde Lyapunov üstlerinden bir tanesi pozitif yörüngededir ve sistem kaotik davranış göstermektedir. Grafiğin çizilmesinde *Matlab Lynch_2007* kullanılmıştır.



Şekil 2.6 Sönümlü ve sinüzoidal olarak sürülen sarkaçta $g=1$ için 3 tane Lyapunov üstleri grafiği



Şekil 2.7 Sönümlü ve sinüzoidal olarak sürülen sarkaçta $g=1.2$ için 3 tane Lyapunov üstleri grafiği

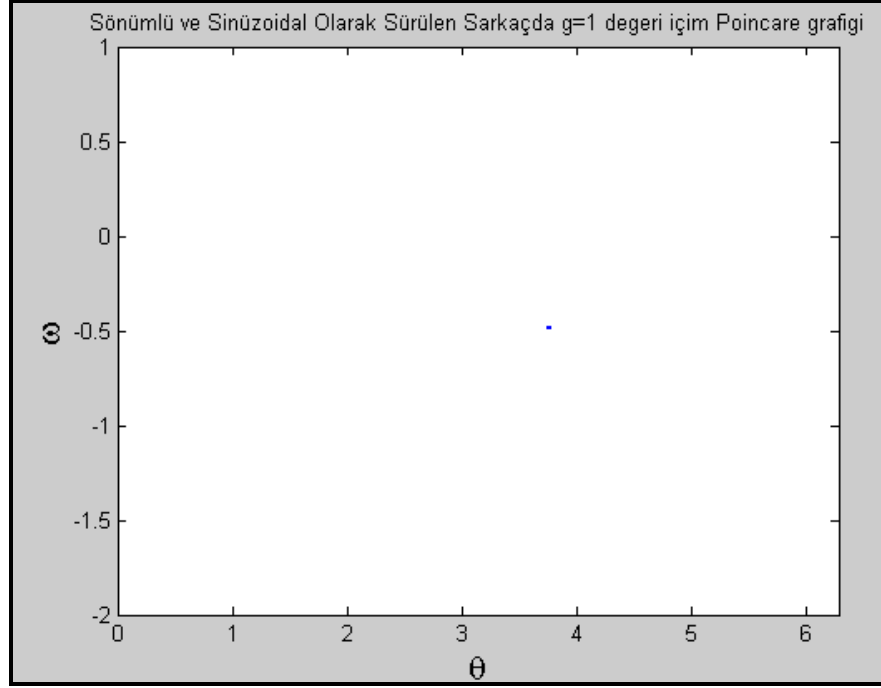
2.1.4 Poincare haritalama yöntemi

Sistemin faz uzayı izlenirken belirli aralıklarla sistemden örnekler alınarak elde edilir. Bu yöntemde, (n) .dereceden sürekli zamanlı dinamik bir sistem, (n) .dereceden ayrık zamanlı dinamik bir sisteme dönüştürülmektedir. Karmaşık sistemleri daha basit hale getirmek ve kararlılık analizi yapmak için elverişlidir [28].

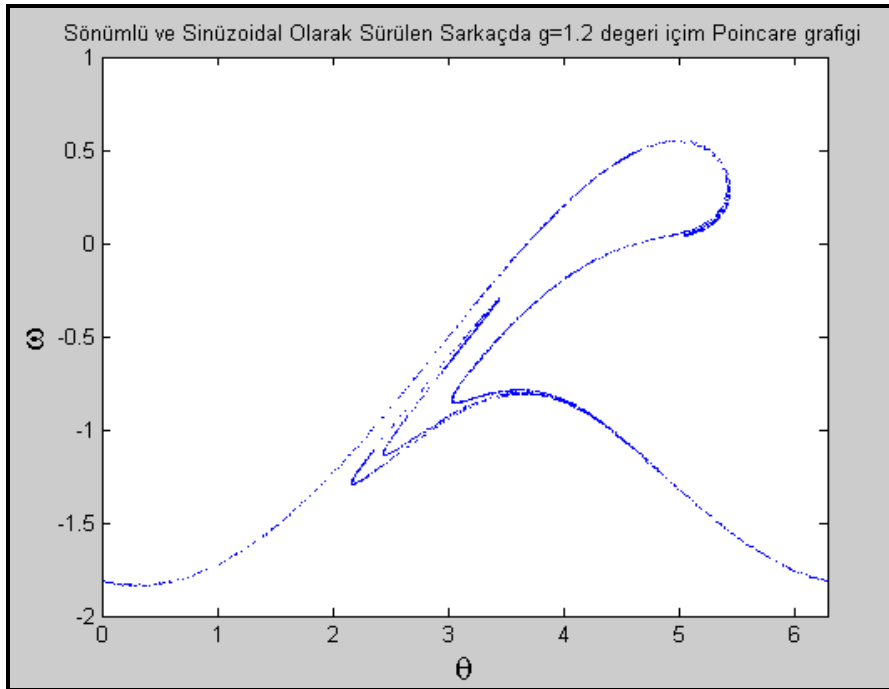
Eğer periyodik davranış gösteren bir sistem Poincare haritalama yöntemi ile incelenirse, sistem sabit bir noktada kalacaktır. Sistemin periyodu ile aynı zamanlara alınan örnekler hep aynı noktada olacağından sistem tek bir noktada görülecektir. Sistem periyodumsu ise Poincare haritalama yöntemi ile incelendiğinde kapalı bir çevrim şekli görülür. Kaotik sistemler ise Poincare haritalama yöntemi ile incelendiğinde geliş güzel, tahmin edilemeyen şekil görülecektir.

Şekil 2.8’de sönümlü ve sinüzoidal olarak sürülen sarkaçta $g=1$ için poincare grafiği verilmiştir. Şekilde görüldüğü gibi sistem tek bir noktada kalmıştır. Sistem Periyodik bir davranış göstermiştir. Şekil 2.9’da sönümlü ve sinüzoidal olarak sürülen sarkaçta $g=1.2$ için poincare grafiği verilmiştir. Şekilde görüldüğü gibi sistem geliş güzel

tahmin edilemeyen bir şekil elde edilmiştir. Sistem kaotik bir davranış göstermiştir. Grafiğin çizilmesinde *Matlab Lynch_2007* kullanılmıştır.



Şekil 2.8 Sönümlü ve sinüzoidal olarak sürülen sarkaçta $g = 1$ için poincare grafiği

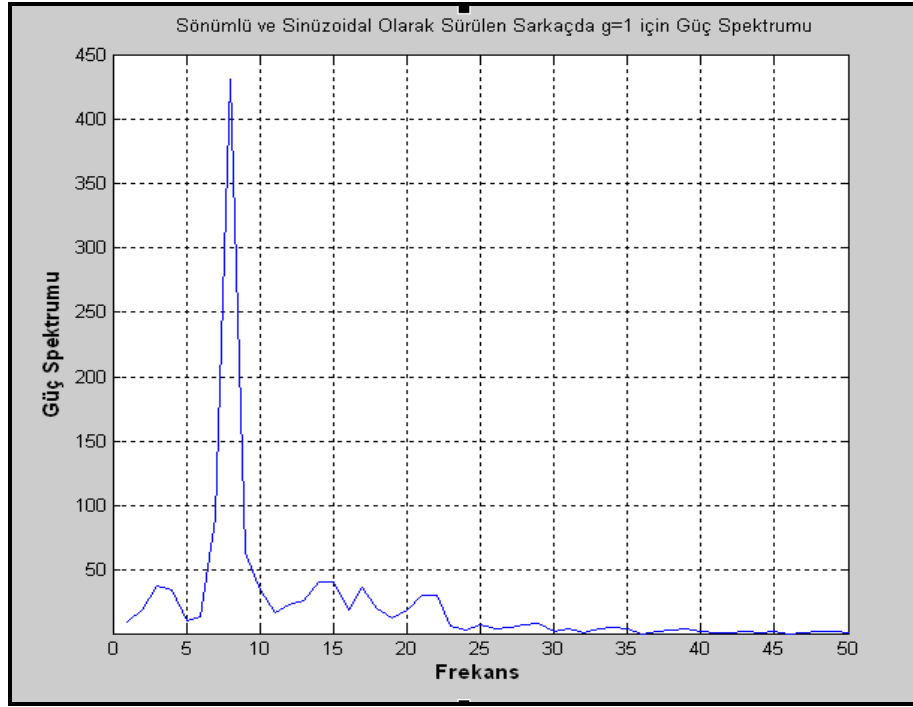


Şekil 2.9 Sönümlü ve sinüzoidal olarak sürülen sarkaçta $g = 1.2$ için poincare grafiği

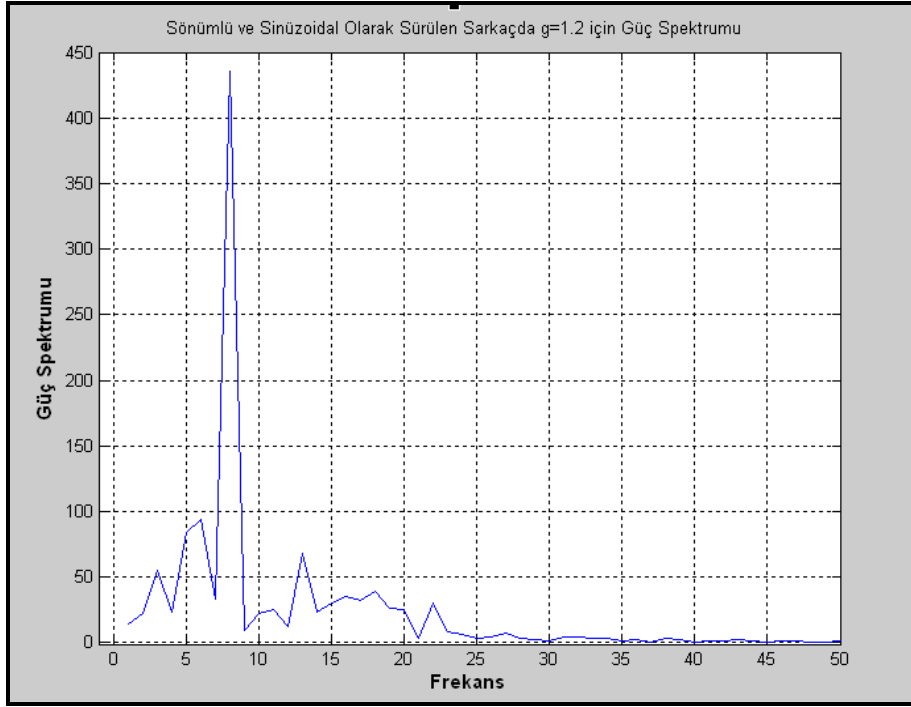
2.1.5 Güç spektrumu ile gösterim yöntemi

Kaotik sinyallerle kaotik olmayan sinyallerin güç spektrumları aynı değildir. Sistemlerin kaotik olup olmadığını belirlemek için sistemlere ait sinyallerin güç spektrumlarına bakılır. Kaotik sinyallerin güç spektrumlarında frekanslar arası sıçramalar daha fazladır. Kaotik olmayan sinyallerde ise daha azdır.

Şekil 2.10'da sönümlü ve sinüzoidal olarak sürülen sarkaçta $g=1$ için güç spektrumu grafiği verilmiştir. Şekilde durum değişkenlerinden ω 'nın güç spektrumu çizilmiştir. Şekil 2.11'de sönümlü ve sinüzoidal olarak sürülen sarkaçta $g=1.2$ için güç spektrumu grafiği verilmiştir. Şekilde durum değişkenlerinden ω 'nın güç spektrumu çizilmiştir. İki şekli kıyasladığımız zaman, kaotik sinyalin güç spektrumunda daha fazla sıçrama görülmektedir.



Şekil 2.10 Sönümlü ve sinüzoidal olarak sürülen sarkaçta $g=1$ için güç spektrumu grafiği

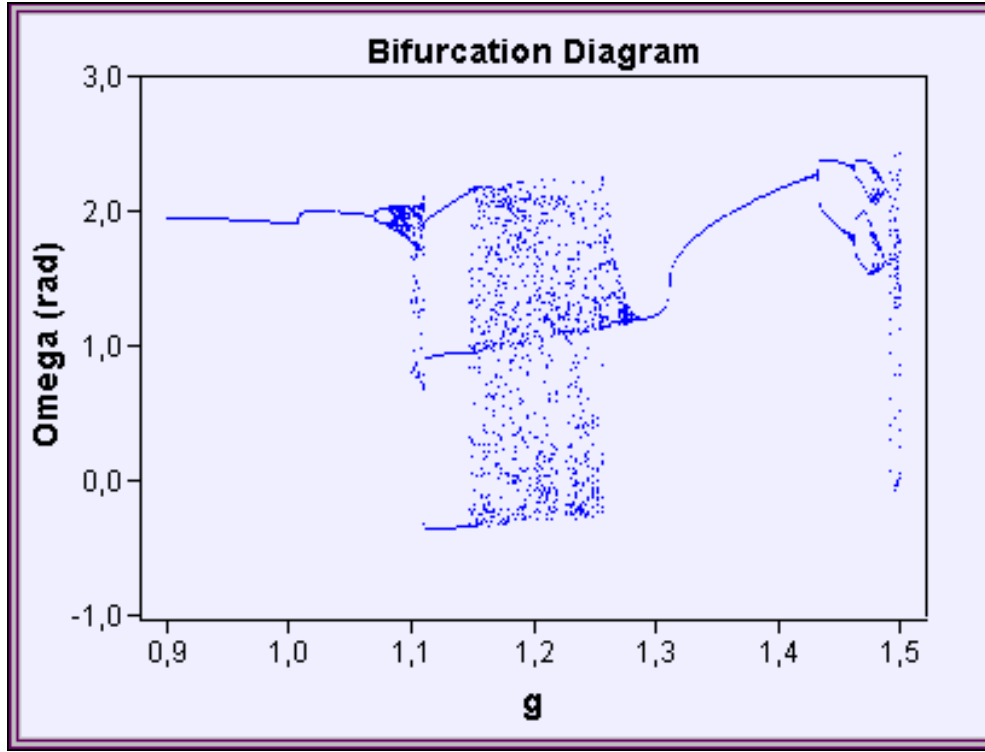


Şekil 2.11 Sönümlü ve sinüzoidal olarak sürülen sarkaçta $g = 1.2$ için güç spektrumu grafiği

2.1.6 Çatallaşma diyagram yöntemi

Şimdiye kadar anlatılan yöntemlerde ya durum değişkenlerinin zamana göre grafikleri ya da durum değişkenlerinin birbirine göre davranışları çizdirilmiştir. Yani parametrelerle ilgili herhangi bir grafik çizilmemişti. Bu yöntemde ise durum değişkenlerinden birinin parametrelerden birine göre davranışı çizilerek, sistemin davranışı incelenecektir.

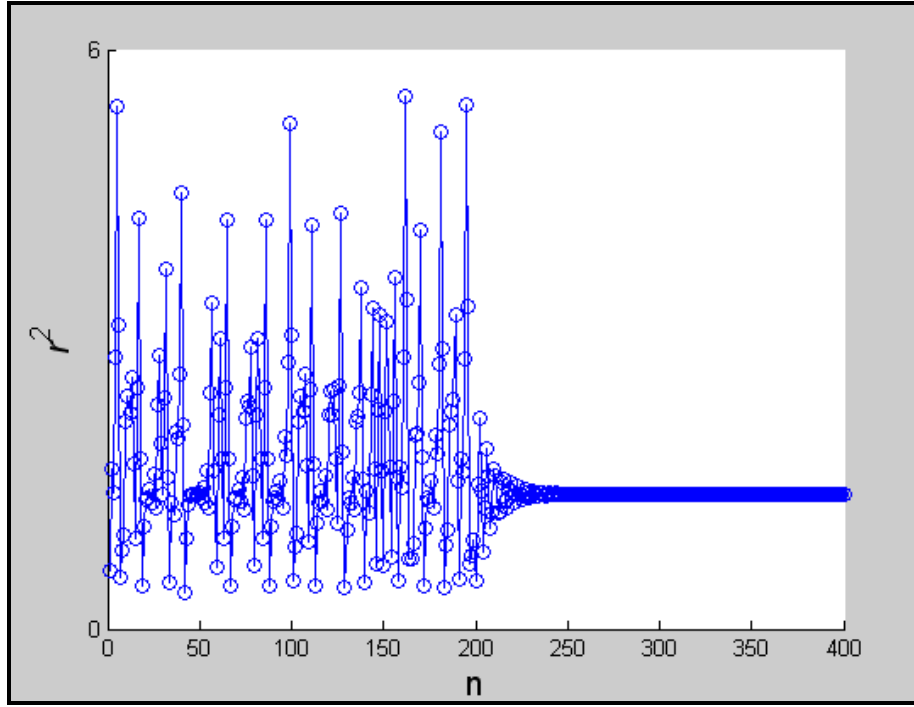
Şekil 2.12’de sönümlü ve sinüzoidal olarak sürülen sarkacın çatallaşma diyagramı grafiği verilmiştir. Şekilde görüldüğü gibi; g parametresinin $[0.9 - 1.1)$ arasındaki değerlerinde sistemimiz kaotik değildir ve tek bir denge noktası vardır. Sistem bu denge noktasına yaklaşarak orada sabit kalır. Sistem periyodik bir davranış göstermektedir. g parametresinin $[1.1-1.2)$ arasındaki değerleri için iki denge noktası vardır ve sistem herhangi bir denge noktasına yaklaşarak orada sabit kalır. Sistem periyodumsu bir davranış göstermektedir. g parametresinin $[1.2-1.3)$ değerinden sonra sistem kaosa girmiştir ve davranışı kestirilemez.



Şekil 2.12. Sönümlü ve sinüzoidal olarak sürülen sarkacın çatallaşma diyagramı grafiği [29]

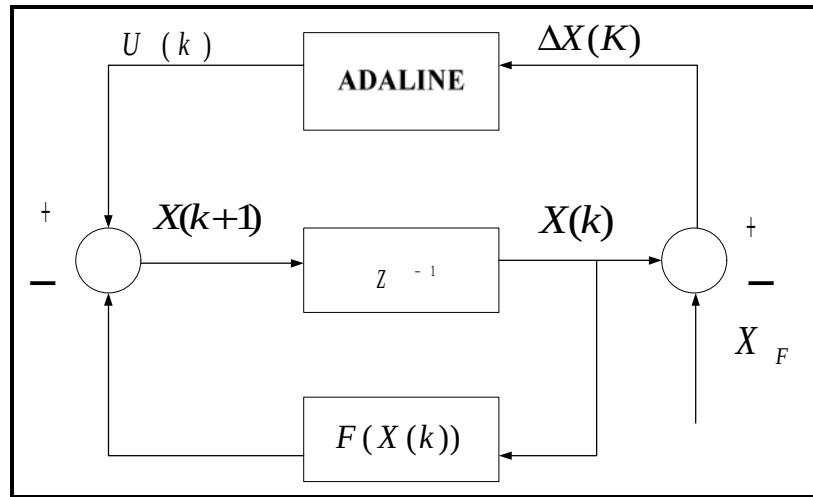
2.2 Kaos Kontrolü

Kaos kontrol algoritmalarını iki ana bölüme ayırabiliriz; geri beslemeli yöntemler, geri beslemesiz yöntemler [30]. Geri beslemeli yöntemlerde, yörüngeler faz uzayında incelenir ve istenilen modda çalışmayan yörüngeler geri besleme işlemleri ile istenilen yörüngeye getirilirler. Geri beslemesiz yöntemde ise sistemin yapısı değiştirilir. Şekil 2.13’de Henon haritasının kaos durumunun kontrolü verilmiştir. Sistem $n=200$ değerine kadar kaotik davranış göstermiştir, $n>200$ ’den sonra kaotik durum gösteren yörüngeler istenilen alana çekilmiştir. Kaotik sistemleri kontrol etmek için yapay sinir ağları denetleyici olarak kullanılarak kontrol sistemleri tasarlanmaktadır. Denetleyici olarak kullanılan yapay sinir ağları eğitilerek kaotik sistemler kontrol edilebilir.



Şekil 2.13 Henon haritanın kaos kontrolü

Deneyssel yada analitik yöntemler kullanılarak kaotik bir sistemde kararsız bir sabit periyot noktası bulunabilir. Sabit nokta deneyle elde ediliyorsa elde edilen nokta yaklaşık bir sabit noktadır ve analitik yöntemlerle sabitlenen sabit noktalar tamamen sabit değildir [31]. Şekil 2.14’de Kaos kontrol sisteminin diyagramı verilmiştir. Adalin ağı kontrolör olarak kullanılmış ve kaos kontrolü için eğitilmiştir.



Şekil 2.14 Kaos kontrol sisteminin diyagramı

3. KAOTİK YAPAY SİNİR AĞLARI

3.1 Yapay Sinir Ağları

Yapay sinir ağları insan beyninin çalışma prensibinden etkilenecek geliştirilmiş, ağırlıklı bağlantılar aracılığı ile birbirine bağlanan ve her biri kendi belleğine sahip olan çok sayıda doğrusal olmayan yapay hücreden (işlem elemanından) meydana gelen bir sistem ya da matematiksel model olarak tanımlanır. Yapay sinir ağının bir olayı öğrenmesi, o olay için en doğru yapay sinir ağının seçilmesi ile mümkündür. Yapay sinir ağının modelini, şu bilgileri karakterize etmektedir [32] ;

- Ağın topolojisi
- Kullanılan toplama fonksiyonu
- Kullanılan aktivasyon fonksiyonu
- Öğrenme stratejisi
- Öğrenme kuralları

Yapay sinir ağlarının kullanıldığı alanlar şu şekilde sıralanabilir;

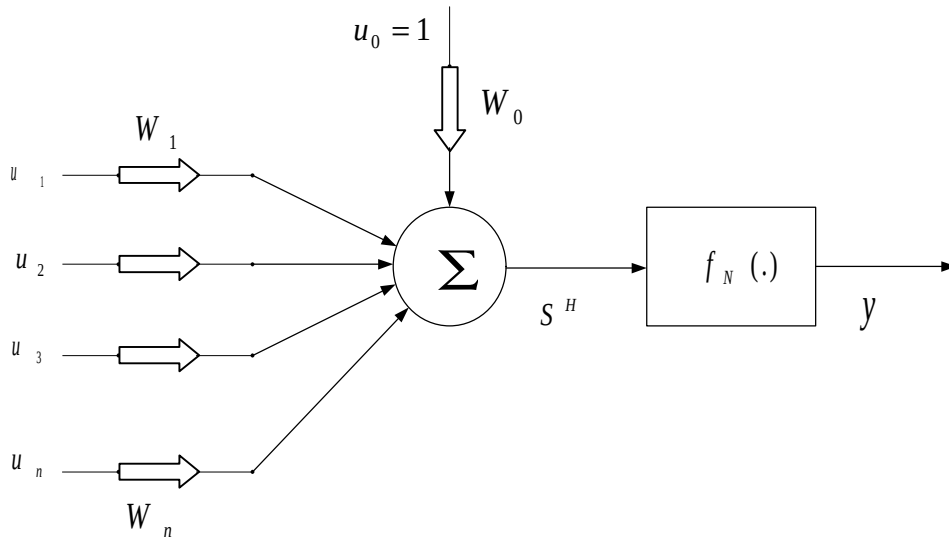
- Denetim ve üretim
- Sistem modelleme
- Ses ve parmak izi tanıma
- Haberleşme
- Meteorolojik yorum
- Arıza analizi ve tespiti
- Tıp Alanı
- Savunma sanayi
- Otomasyon ve kontrol

Yukarıda verilen başlıklara ilave olarak, yapay sinir ağları her türlü bilgiyi işlemek ve analiz etmek için kullanılır.

3.1.1 Yapay Hücre Modeli

Biyolojik sinir ağlarının en küçük birimi sinir hücresi olduğu gibi, yapay sinir ağlarının da çalışmasını esas teşkil eden en küçük birim yapay sinir hücresidir. Şekil 3.1’de görüldüğü gibi her hücrenin 5 temel elemanı vardır. Girdiler, bir yapay sinir hücresine dış ortamdan, diğer hücreden ya da kendisinden gelen bilgilerdir. Ağırlıklar, bir yapay hücreye gelen bilginin önemini ve hücre üzerindeki etkisini gösterir. Toplama fonksiyonu, bir hücreye gelen net girdiyi hesaplayan bir fonksiyondur. Genellikle hücrenin net girdisi, girişlerin ilgili ağırlıkla çarpımlarının toplamıdır. Birleştirme fonksiyonu, ağ yapısına göre farklı toplama fonksiyonları kullanılabilir. Maksimum alan, minimum alan, çarpım, çoğunluk, kumlatif toplam fonksiyonu birer örnektir. Hücre modellerinde net girdiyi +1 arttıran ya da -1 azaltan polarma girişi olabilir. Aktivasyon fonksiyonu, hücrenin net girdisini bir işlemde geçirerek hücrenin çıktısını belirleyen bir fonksiyondur. Yapay hücrenin modeline ve gerçekleştireceği görevine göre farklı aktivasyon fonksiyonları bulunmaktadır. Net girdinin aktivasyon fonksiyonundan geçmesiyle belirlenen çıktı değeridir. Üretilen çıktı dış dünyaya, başka bir hücreye ya da hücrenin kendisine tekrar girdi olarak gönderilebilir.

Ağırlıkları sabit olan, hücrede geri besleme ya da geciktirilmiş sinyallerin olmadığı, şekil 3.1’de verilen yapay sinir hücresinin matematiksel ifadesi denklem 3.1’de verilmiştir.



Şekil 3.1 Yapay Hücre Modeli

Şekil 3.1’deki yapay hücre modelinin matematiksel ifadesi;

$$S^H = \sum_{i=0}^n W_i u_i \quad yada \quad S^H = \sum_{i=1}^n W_i u_i + b \quad , \quad y = f_N(S^H) \quad (3.1)$$

Burada;

W = Hücrenin ağırlıklar matrisini,

u = Hücrenin giriş vektörünü,

S^H = Hücrenin net girişini,

b = Polarma ağırlığı,

y = Hücre çıkışını,

$f_N(.)$ = Hücrenin aktivasyon fonksiyonunu göstermektedir.

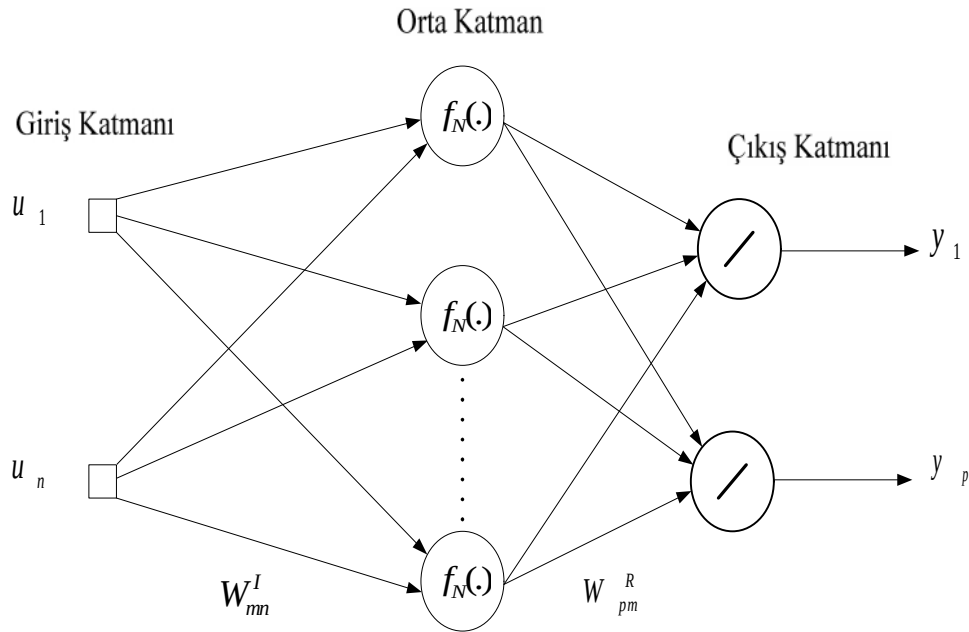
3.1.2 Yapay Sinir Ağ Yapıları

Yapay sinir ağları, yapay sinir hücrelerinin birbirleri ile çeşitli şekillerde bağlanmaları ile oluşur. Hücre çıkışları ağırlıklar üzerinden diğer hücrelere ya da kendisine giriş olarak bağlanabilir ve bağlantılarda gecikme birimi de kullanılabilir. Hücreler 3 katman içinde bir araya gelerek yapay sinir ağını oluştururlar. Bu katmanlar: Girdi katmanı, dış ortamdan ya da diğer hücrelerden gelen bilgileri ara katmana iletmekle sorumlu olan katmandır. Ara katman girdi katmanından gelen bilgileri işleyerek çıktı katmanına iletmekle sorumludur. Bir yapay sinir ağında birden fazla ara katman olabilir. Çıkış katmanı ara katmandan gelen bilgileri işleyerek, ağı girdi katmanından gelen bilgilere istenilen çıktıyı üretilip, dış dünyaya göndermekle sorumlu olan katmandır.

Problemlerin çözümünde kullanılan farklı yapay sinir ağları vardır. Bugün 50’ye yakın farklı yapılanma, diğer bir deyişle farklı model görülmekte ve bu sayı her geçen gün artmaktadır [33]. Bu bölümde sistem modellemede kullanılacak iki yapay sinir ağ modeli açıklanacaktır.

3.1.2.1 İleri Beslemeli Yapay Sinir Ağları (İBYSA)

İleri beslemeli yapay sinir ağlarında, hücreler katmanlar halinde düzenlenir ve ileriye doğru bağlantılıdır. Bir hücrenin çıkışı diğer hücrenin girişi olmaktadır. Giriş katmanı dış ortamdan aldıkları bilgileri ara katmana gönderir ve bu katmanda bilgi işleme olmaz. Gelen her bilgi, hiçbir işleme maruz kalmadan ara katmana gider. Ara katmana gelen bilgiler, işlenerek çıktı katmanına gönderilir ve çok sayıda ara katman bulunabilir. Ara katmandan gelen bilgiler işlenerek çıktı katmanından dış ortama iletilir. Bir çıktı katmanında birden fazla hücre olabilir. Her hücre bir önceki katmandaki hücrelerle bağlantılıdır. Bu tip YSA'ların eğitiminde en çok bilinen geriye yayılım öğrenme algoritması etkin olarak kullanılmaktadır. Şekil 3.2'de ileri beslemeli 3 katmanlı YSA modeli verilmiştir.



Şekil 3.2 İleri beslemeli 3 katmanlı YSA

İleri beslemeli 3 katmanlı ve çıkış katmanı doğrusal olan YSA'nın matematiksel modeli denklem 3.2 ve 3.3'deki gibi yazılabilir.

$$S^H_j = \sum_{i=0}^n W_{ji}^I u_i, \quad x_j = f_N(S^H_j) \quad j = 1, 2, \dots, m \quad (3.2)$$

$$y_{\ell} = \sum_{j=0}^m W_{\ell j}^R \cdot x_j \quad \ell = 1, 2, \dots, p \quad (3.3)$$

Burada;

W^I = Girdi katmanındaki hücrelerin ağırlıklar matrisini,

u = Hücresinin giriş vektörünü,

S^H = Hücresinin net girişini,

W^R = Ara katmandaki hücrelerin ağırlık matrisini,

x = Ara katmandaki hücrelerin çıkışı,

y = Hücre çıkışı,

$f_N(.)$ = Hücresinin aktivasyon fonksiyonunu göstermektedir.

XOR problemini çözmek için yapılan çalışmalar sonucu ileri beslemeli yapay sinir ağları modeli geliştirilmiştir. Rumelhart ve arkadaşları tarafından geliştirilen bu modele hata yayma modeli ya da geriye yayılım modeli de denilmektedir [32].

3.1.2.2 Geri Beslemeli Yapay Sinir Ağları (GBYSA)

Hücrelerden en az bir tanesinin çıktısı, kendisine veya diğer hücrelere giriş olarak verilen yapay sinir ağlarıdır. Geri besleme genellikle bir geciktirme elemanı üzerinden yapılmaktadır. Ağdaki geri besleme aynı katmandaki hücreler arasında olduğu gibi farklı katmandaki hücreler arasında da olabilmektedir. Bu yapısı ile geri beslemeli yapay sinir ağları, doğrusal olmayan davranış gösterirler ve doğrusal olmayan çözümler getirebilen bir ağıdır. Geri beslemeli yapay sinir ağlarında ara katman sayısı için belirli bir değer yoktur. Çözümde yardımcı olacak kurallar vardır.

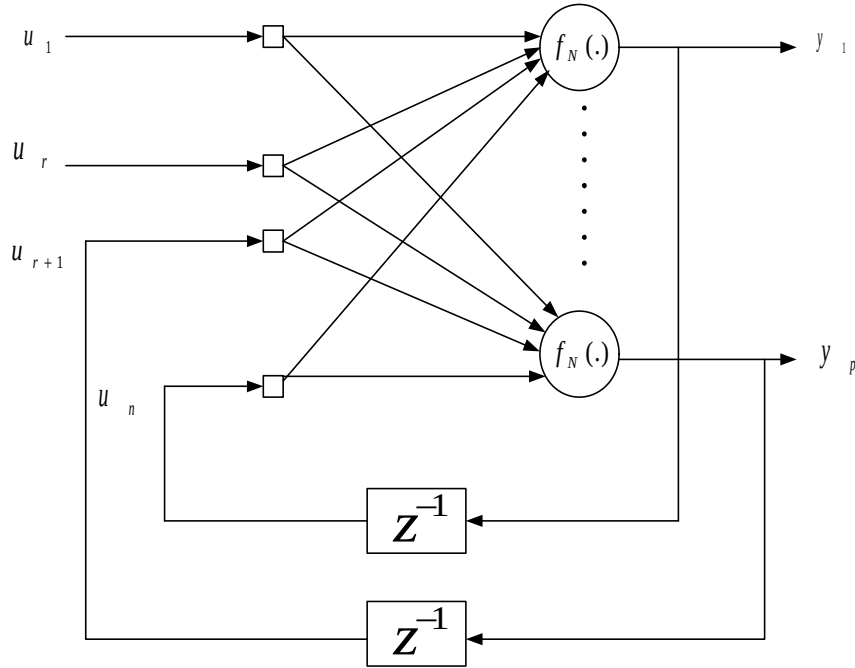
1. Girdi verisi ve çıktı arasındaki karmaşıklık artarsa, gizli katmandaki hücre sayısı da arttırılmalıdır.

2. Ele alınan süreç birçok aşamalara ayrılıyorsa ara katman sayısı arttırılmalı, ayrılmıyorsa çok fazla ara katman kullanılmamalı, yoksa ağda sadece ezberleme olur ve yanlış sonuç çıkar.

3. Ağda kullanılan eğitim verisinin miktarı, ara katmandaki elemanların sayısı için bir üst sınır oluşturmaktadır. Girdi ve çıktı çiftlerinin sayısı bulunur. Bulunan sayı ağdaki

toplam giriş ve çıkış düğümlerine bölünür. Çıkan sonuç 5-10 arasındaki bir dereceleme faktörüne bölünür [34].

Geri beslemenin yapılış şekline göre farklı geri beslemeli yapay sinir ağları vardır. Şekil 3.3’de iki katmanlı ve ağırlıklı çıkışlarından giriş katmanına geri beslemeli bir YSA yapısı görülmektedir.



Şekil 3.3 Geri beslemeli iki katmanlı YSA

Şekil 3.3’de verilen geri beslemeli YSA da,

r = Giriş vektör sayısı,

p = Ağırlıklı çıkışından gelen gecikmişlerin sayısını ifade eder.

YSA’nın giriş vektörü denklem 3.4’de verilmiştir.

$$u = (u_1, u_2, \dots, u_r, y_1(-1), y_2(-1), \dots, y_p(-1)) \quad (3.4)$$

Toplam $n = r + p$ adet ağ girişine göre YSA’nın matematiksel çıkışı denklemi 3.5’deki gibi yazılabilir.

$$y_j = f_N\left(\sum_{i=0}^n W_{ji} \cdot u_i\right) \quad j=0,1,2,\dots,p \quad (3.5)$$

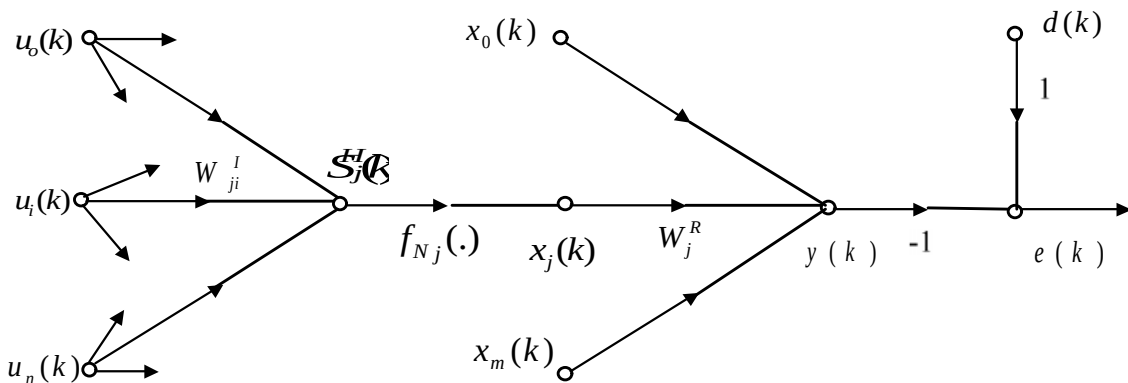
Geriye yayılım ağı 1970'lerin başında geliştirilmiş en popüler, en etkili ve en karmaşık, tanımlanamamış problemlere doğrusal olmayan çözümler getirebilen bir ağ çeşididir [34].

Yapay sinir ağının öğrenme işlemini gerçekleştirebilmesi için, sahip olduğu bütün ağırlıkların ilgili problemde öğrenilmesi istenen özellikleri genelleştirecek şekilde doğru değerlere sahip olması gerekir. Bu doğruluk ne kadar artarsa ağın öğrenme işlemi o kadar iyi olur. Doğru ağırlık değerleri, bir öğrenme kuralına göre tespit edilirler. Çoğunlukla bağlantılara başlangıç değeri olarak rastgele ağırlıklar atanır ve bu ağırlıklar eldeki örneklerle incelendikçe bir kurala göre değiştirilerek doğru ağırlık değerleri bulunmaya çalışılır. Bu işleme ise öğrenme denir [35]. Bu çalışmada YSA'nın öğrenmesinde geriye yayılım öğrenme algoritması kullanılacaktır.

3.1.3 Geriye Yayılım Öğrenme Algoritması

Geriye yayılım algoritması ile öğrenen ağlar hiyerarşik yapıdadırlar. Bu ağlar giriş, çıkış ve en az bir tane gizli katmandan oluşmaktadır. Gizli katman birçok düğümden oluşabilir, düğüm sayısı arttıkça ağın hatırlama yeteneğini artırır ve öğrenme işlem süresinin uzatmaktadır.

Çıkış katmanı doğrusal olan ileri beslemeli, 3 katmanlı YSA'nın sinyal akış şeması şekil 3.4'deki gibi çizilir.



Şekil 3.4 İleri beslemeli 3 katmanlı YSA sinyal akış şeması

Şekil 3.4’deki sinyal akış şemasından geçen sinyaller ayrık zamanda gösterilmiştir.

$u(k)$ =Ağ giriş vektörünü,

$x(k)$ =Orta katman çıkış vektörünü,

$d(k)$ =Arzu edilen çıkış vektörünü,

$y(k)$ =Gerçek ağ çıkış vektörünü,

W^I ve W^O =Ağırlıklar matrisini göstermektedir.

Şekil 3.4’deki sinyal akış şemasının matematiksel modeli denklem 3.6’da verilmiştir.

$$S^H_j = \sum_{i=0}^n W_{ji}^I \cdot u_i, \quad x_j = f_N(S^H_j) \quad j = 1, 2, \dots, m$$

$$y_\ell = \sum_{j=0}^m W_j^O \cdot x_j$$
(3.6)

Tek çıkışlı bir YSA için ağ çıkış hatası, hataların kareleri (örneksel amaç ölçütü) ve toplam amaç ölçütü (hataların karelerinin ortalaması) denklem 3.7’deki gibi ifade edilir.

$$e(k) = d(k) - y(k), \quad E(k) = \frac{1}{2} e^2(k), \quad J = \frac{1}{N} \sum_{k=1}^N E(k)$$
(3.7)

Toplam amaç ölçütü, N adet eğitim örneği için hataların karelerinin ortalaması olduğundan örneksel ya da toplu amaç ölçütünün ağırlıklara göre eğimi bulunarak geriye yayılım algoritması gerçekleştirilebilir. Hatalarının karelerinin eğimine göre, her bir eğitim örneğinin uygulanışında ağırlıklar yenilenirse örneksel öğrenme kuralı elde edilir. Toplam amaç ölçütünün eğimine göre, N adet eğitim örneğinin uygulanışından sonra ağırlıklar yenilenirse toplu öğrenme kuralı elde edilir [8].

$$\frac{\partial E(k)}{\partial W_j^O(k)} = \frac{\partial E(k)}{\partial \hat{e}(k)} \cdot \frac{\partial \hat{e}(k)}{\partial y(k)} \cdot \frac{\partial y(k)}{\partial W_j^O(k)}$$
(3.8)

Denklem 3.8’deki türev alınırsa;

$$\delta^2(k) = \frac{\partial E(k)}{\partial e(k)} \cdot \frac{\partial e(k)}{\partial y(k)} = -e(k) \quad (3.9)$$

$$\frac{\partial y(k)}{\partial W_j^O(k)} = x_j(k) \quad (3.10)$$

Buna göre çıkış katmanındaki ağırlıklara uygulanacak düzeltme ve ağırlıkların yeni değerleri denklem 3.11'deki gibi ifade edilir.

$$\Delta W_j^O(k) = \alpha \cdot \delta^2(k) \cdot x_j(k) \quad , \quad W_j^O(k+1) = W_j^O(k) - \Delta W_j^O(k) \quad (3.11)$$

Burada, δ_q^p = p. katmandaki q. hücrenin yöresel hatası olarak ifade edilir. Hataların kareleri toplamının orta katmandaki ağırlığa göre türevi denklem 3.12'deki gibi ifade edilir.

$$\frac{\partial E(k)}{\partial W_{ji}^I(k)} = \frac{\partial E(k)}{\partial e(k)} \cdot \frac{\partial e(k)}{\partial y(k)} \cdot \frac{\partial y(k)}{\partial x_j(k)} \cdot \frac{\partial x_j(k)}{\partial S_j^H(k)} \cdot \frac{\partial S_j^H(k)}{\partial W_{ji}^I(k)} \quad (3.12)$$

Denklem 3.13'deki türev alınır,

$$\delta_j^I(k) = \frac{\partial E(k)}{\partial e(k)} \cdot \frac{\partial e(k)}{\partial y(k)} \cdot \frac{\partial y(k)}{\partial x_j(k)} \cdot \frac{\partial x_j(k)}{\partial S_j^H(k)} = -e(k) W_j^O(k) f_N'(S_j^H(k)) \quad (3.13)$$

ve

$$\frac{\partial S_j^H(k)}{\partial W_{ji}^I(k)} = u_i(k) \quad (3.14)$$

Buna göre orta katmanındaki ağırlıklara uygulanacak düzeltme ve ağırlıkların yeni değerleri denklem 3.15'deki gibi ifade edilir.

$$\Delta W_{ji}^I(k+1) = \alpha \cdot \delta_j^I(k) \cdot u_i(k) \quad , \quad W_{ji}^I(k+1) = W_{ji}^I(k) - \Delta W_{ji}^I(k) \quad (3.15)$$

Yukarıdaki ağırlıkların yenilenmesinde, hataların karelerinin ağırlıklara göre eğimini bularak örneksel öğrenme ile geriye yayılım algoritması çıkartılmıştır. Toplu öğrenme ile geriye yayılım algoritması denklem 3.16'daki gibi ifade edilir.

$$\frac{\partial}{\partial \mathbf{w}_{ji}^I} = \frac{1}{N} \sum_{k=1}^N \frac{\partial \mathcal{E}(k)}{\partial \mathbf{w}_{ji}^I} \quad (3.16)$$

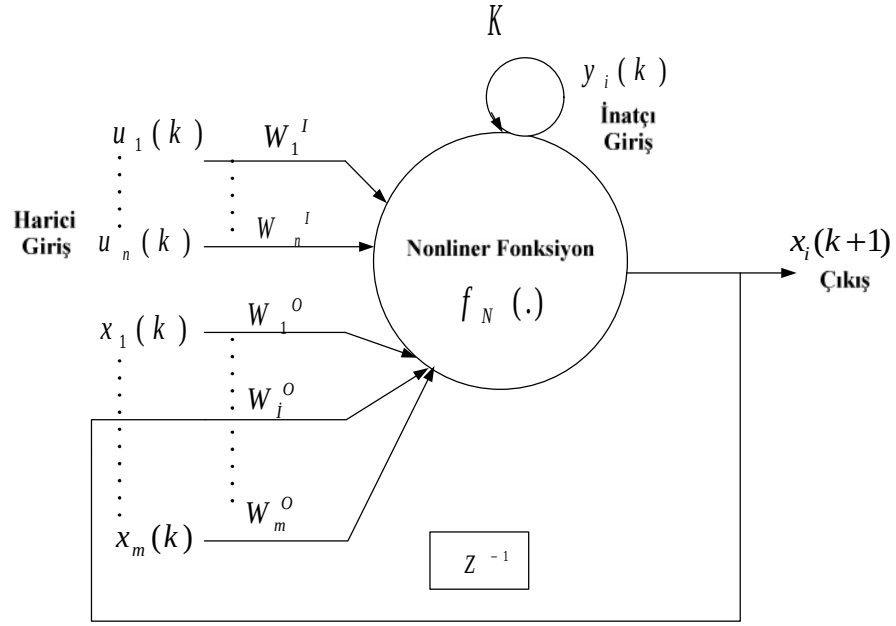
YSA ile ilgili önemli bir durum da düzgün bir öğrenme oranının belirlenmesidir. Öğrenme oranı yüksek ayarlandığında, sistemin davranışının bozulmasına sebep olur. Öte yandan çok küçük bir öğrenme oranı ayarlandığında da öğrenme işleminin yavaşlamasına sebep olur.

3.2. Kaotik Hücre

Biyolojik nöronlar genellikle sürekli ya da geçici periyotlar için kaotik karakteristiklere sahiptir [36]. Birçok araştırmacı tarafından, biyolojik nöronların kaotik davranışları modellenmiş ve incelenmiştir. İlk kez Caianiello üniversitesinden Hadking-Huxley, kaotik nöronları model edinmiştir [37,38]. Naogumo Sato [39] kaotik yapay sinir ağlarını yapmak için ilk kez bu modeli kullanmıştır. Aihara ve arkadaşları çıkış koşulları ile bir ayrık zaman modeli yapmış ve kaotik sinir ağları yapmada bu modeli kullanmıştır [40].

Kaotik yapay sinir ağları optimizasyon problemlerinin çözümünde kullanılmıştır. Nöronların kaotik karakteristiği genellikle optimizasyon problemlerinin üzerinde ters etki verebilir fakat nöron modelinin geçici kaosu lokal minimum problemler üzerinde faydalı olabilmektedir [11]. Aihara ve arkadaşları global optimizasyon için faydalı olabilmüş nöronların geçici kaotik karakteristiğini tanımlamıştır [41,42].

Bazı değişiklikler yapılsa bile ilk hazırlanmış kaotik nöronlar, yapay sinir ağındaki uygulamayı hala zorlaştırır ve nöronun kendi kendisine öğrenmesinde çok fazla dinamik karakteristiklere ihtiyacı vardır [11]. Şekil 3.5’de kaotik hücrenin şekli verilmiştir.



Şekil 3.5 Kaotik nöron birimi

Şekil 3.5’deki kaotik yapay hücrenin matematiksel ifadesi aşağıda verilmiştir.

$$y_i(k+1) = K.y_i(k) + \sum_{j=1}^n W_{ij}^I .u_j(k) + \sum_{j=1}^m W_{ij}^O .x_i(k) \quad (3.17)$$

$$x_i(k+1) = f_N[y_i(k+1)] \quad (3.18)$$

$$f_N[y_i(k+1)] = \frac{1}{1 + e^{\frac{-y_i(k+1)}{\varepsilon}}} \quad (3.19)$$

$u(k)$ = Kaotik hücrenin harici girişleri,

$y(k)$ = Geri beslemenin girişleri,

W^I, W^O = Ağırlıklar,

$x(k+1)$ = (k+1). zamandaki hücrenin çıktısı,

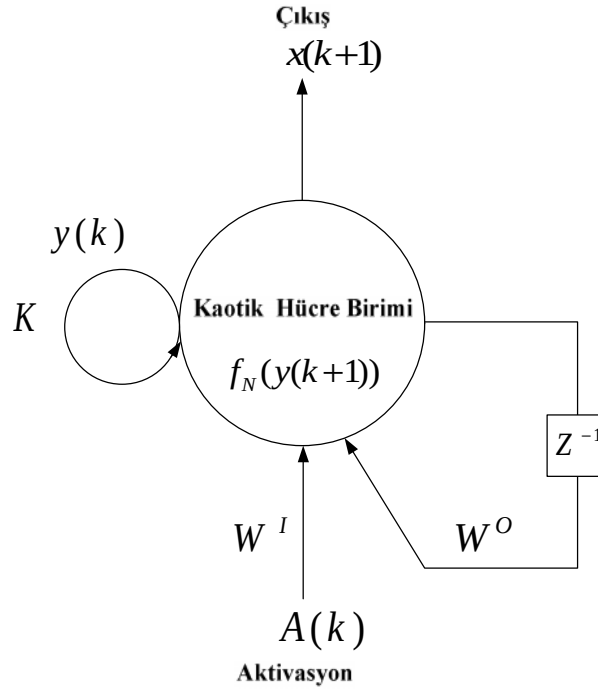
$y(k)$ = İnatçı giriş.

Şekil 3.5’e göre kaotik hücrenin üç tane girişi vardır, bunlar; harici girişler, inatçı giriş ve hücrenin geri beslemesidir. Bu çalışmada iki tane örnek kaotik hücre modeli verilmiş ve hücrelerin davranışları incelenmiştir.

3.2.1 Birinci Kaotik Hücrenin Kaotik Davranışının Analizi

Kaotik nöron modelleri, diğer kaotik nöronların geri beslemeleri ve harici girişlerin toplanarak düşünüldüğü *kaotik yapay sinir ağı* olarak adlandırdığımız yapay sinir ağlarının bileşenleridir [11].

Kaotik nöron modeli doğrusal olmayan karmaşık hale sahip fonksiyondur. Nagumo-Sato, Aihara ve arkadaşları tarafından tasarlanmış tek girişli hücre modeli şekil 3.6'da gösterilmiştir. Denklem 3.20, 3.21 ve 3.22'de kaotik hücrenin matematiksel ifadeleri verilmiştir.



Şekil 3.6 Kaotik dinamik hücre birimi

$$y(k+1) = K \cdot y(k) - x(k) + A(k) \quad (3.20)$$

$$x(k+1) = f_N[y_i(k+1)] = \frac{1}{1 + e^{\frac{-y_i(k+1)}{\varepsilon}}} \quad (3.21)$$

Yukarıdaki denklemlere göre;

$x(k+1)$ = (k+1). zamandaki hücrenin çıkışı,

$y(k+1)$ = (k+1). zamandaki hücrenin girişi,

K = İnatçı parametre,

$A(k)$ = k. zamandaki aktivasyon fonksiyonu,

ε = Sigmoid fonksiyonunun eğimi.

Denklem 3.20’de gösterildiği gibi kaotik hücrenin üç tane girişi bulunmaktadır. İnatçı, başlangıç(eşik), aktivasyon girişi ve hücrenin geri beslemesidir. Hücrenin çıkış denklemi 3.21’de doğrusal olmayan sigmoid fonksiyon ile tanımlanmıştır. Hücrenin çıkışı farklı aktivasyon değerlerine göre değişmektedir. $\varepsilon=0.06$ ve $K=0.7$ olsun. Dört farklı aktivasyon değeri için, $A=0.2$, $A=0.5$, $A=0.7$, $A=0.9$, hücrenin çıkışı incelenecektir. Hücre $A=0.2$, $A=0.7$ değerleri için kaotik çıkış göstermektedir. $A=0.5$, $A=0.9$ değerleri için çıkış sabitlenmiştir [19].

Şekil 3.6’daki kaotik hücre bölüm 2’de anlatılan doğrusal olmayan sistemlerde kaos analizi yapmak için en çok kullanılan yöntemlerle incelenecektir.

—Yörüngenin izlenmesi (zaman serileri)

—Lyapunov üstleri

—Poincare haritalama

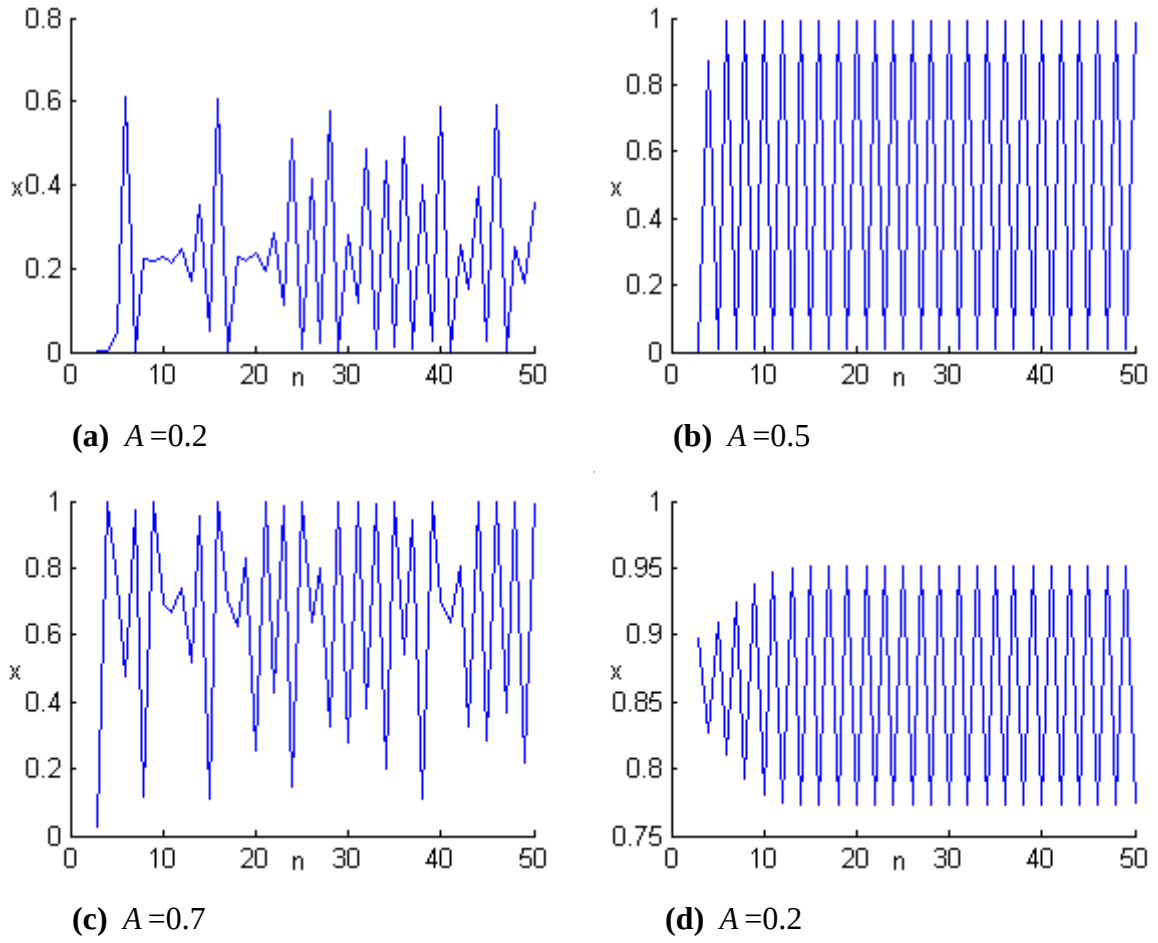
—Güç spektrumu

—Çatallaşma (Bifurcation) diyagramı.

3.2.1.1 Yörüngelerin İzlenmesi Yöntemi

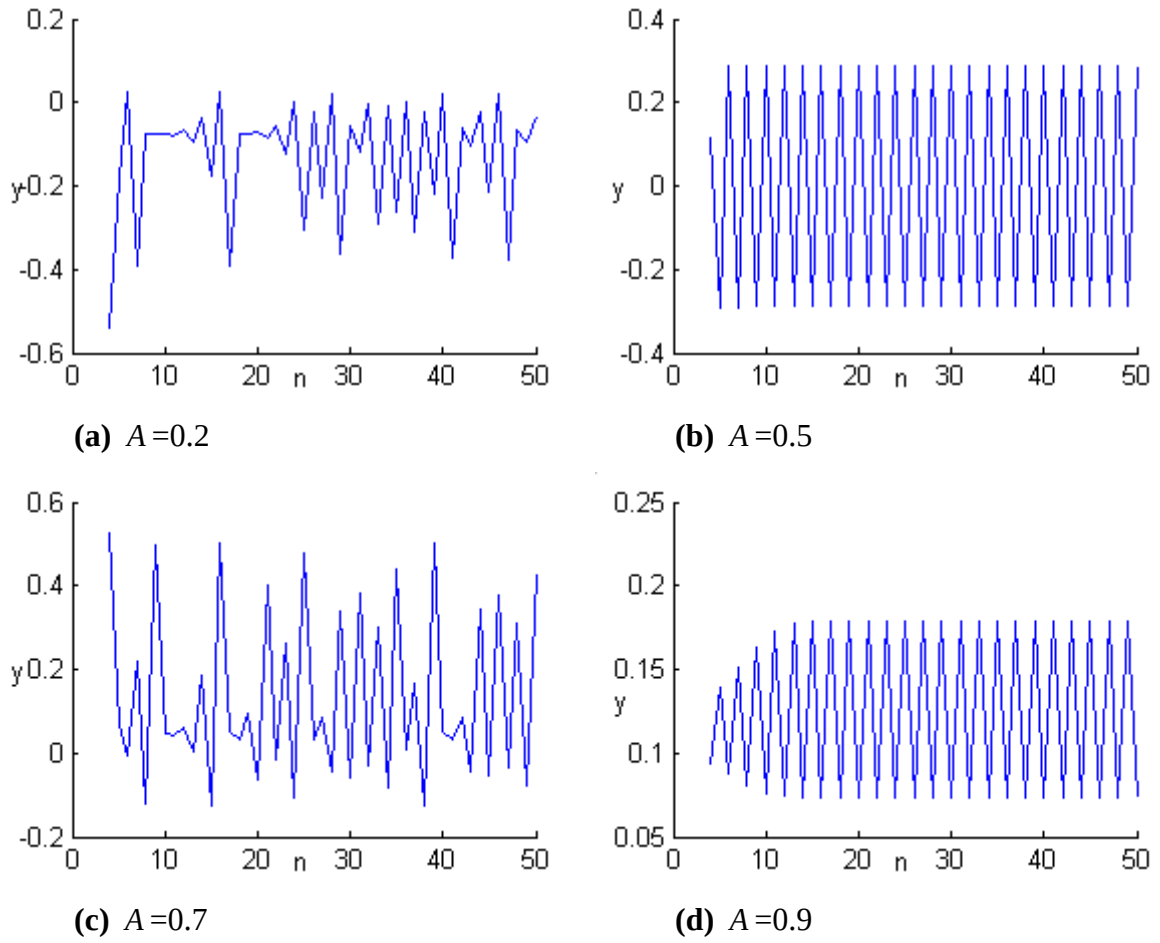
x eksenine zaman, y eksenine hücrenin çıkışı girilerek çizilen grafikte, hücrenin kaosa girip girmediğine bakılabilir.

Yörüngelerin izlenmesi yöntemine göre; A ’nın dört farklı değeri için $x-n$ grafiğini çizilmiştir. Şekil 3.7(a)’da $A=0.2$ değeri için hücre girişinin (x)-zaman ekseninde grafiği çizilmiş, hücrenin kaotik davranışı gösterilmiştir. Şekil 3.7(b)’de $A=0.5$ değeri için hücre girişinin (x)-zaman ekseninde grafiği çizilmiş, hücrenin zaman ekseninde periyodik davranış gösterilmiştir. Şekil 3.7(c)’de $A=0.7$ değeri için hücre girişinin (x)-zaman ekseninde grafiği çizilmiş, hücrenin kaotik davranış gösterilmiştir. Şekil 3.7(d)’de $A=0.9$ değeri için hücre girişinin (x)-zaman ekseninde grafiği çizilmiş, hücrenin zaman ekseninde periyodik davranış gösterilmiştir.



Şekil 3.7 (a) $A=0.2$ (b) $A=0.5$ (c) $A=0.7$ (d) $A=0.9$ $x-n$ grafiği

Yörüngelerin izlenmesi yöntemine göre; A 'nın dört farklı değeri için $y-n$ grafiğini çizilmiştir. Şekil 3.8(a)'da $A=0.2$ değeri için hücre çıkışının (y)-zaman ekseninde grafiği çizilmiş, hücrenin kaotik davranış gösterilmiştir. Şekil 3.8(b)'de $A=0.5$ değeri için hücre çıkışının (y)-zaman ekseninde grafiği çizilmiş, hücrenin zaman ekseninde periyodik davranış gösterilmiştir. Şekil 3.8(c)'de $A=0.7$ değeri için hücre çıkışının (y)-zaman ekseninde grafiği çizilmiş, hücrenin kaotik davranış gösterilmiştir. Şekil 3.8(d)'de $A=0.9$ değeri için hücre çıkışının (y)-zaman ekseninde grafiği çizilmiş, hücrenin periyodik davranış gösterilmiştir.



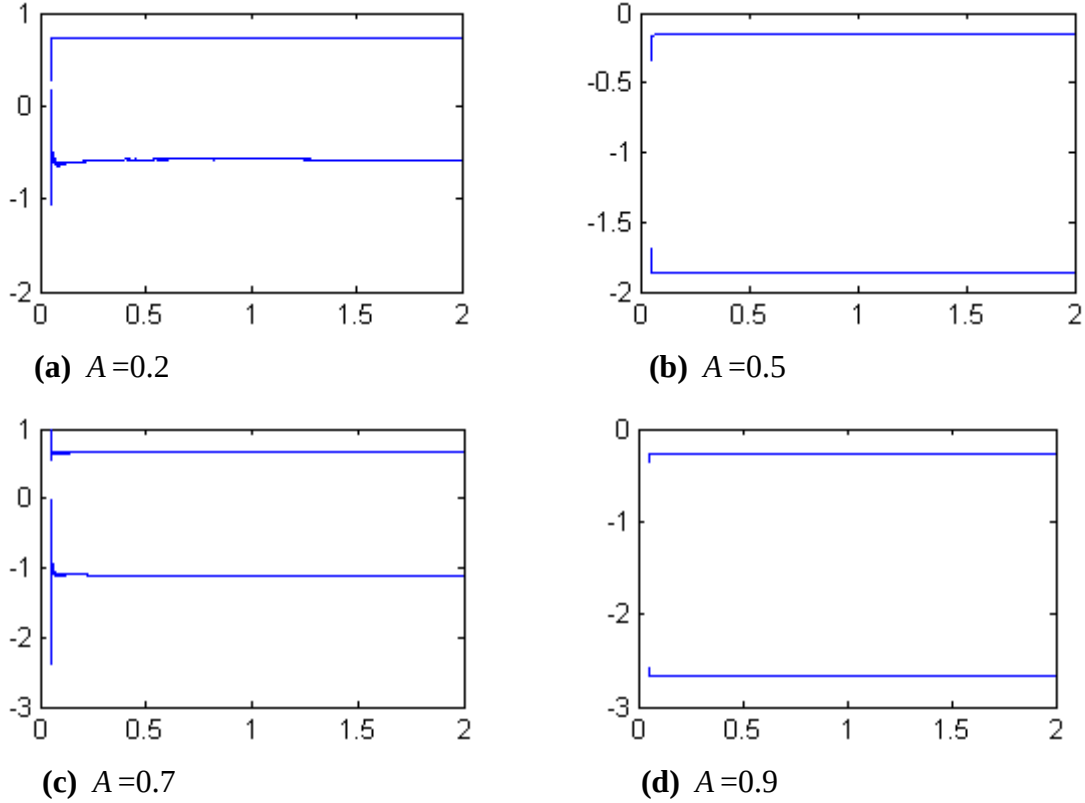
Şekil 3.8 (a) $A=0.2$ (b) $A=0.5$ (c) $A=0.7$ (d) $A=0.9$ $y - n$ grafiği

3.2.1.2 Lyapunov üstleri

Kaotik hücrenin davranışı iki tane Lyapunov değerine göre incelenmiştir. Eğer Lyapunov üstlerinin hepsi negatif yörüngede ise hücre kaotik olmayan bir davranış gösterecektir. Lyapunov üstlerinden biri pozitif yörüngede ise hücre kaotik bir davranış gösterecektir. Grafiğin çizilmesinde Matlab *Lynch_2007* kullanılmıştır.

Şekil 3.9(a)'da $A=0.2$ için 2 tane Lyapunov üstlerinin grafiği çizilmiştir. Şekilde Lyapunov üstlerinden bir tanesi pozitif yörüngededir ve hücre kaotik davranış göstermektedir. Şekil 3.9(b)'de $A=0.5$ için 2 tane Lyapunov üstlerinin grafiği çizilmiştir. Şekilde Lyapunov üstlerinin hepsi negatiftir ve hücre periyodik davranış göstermektedir. Şekil 3.9(c)'de $A=0.7$ için 2 tane Lyapunov üstlerinin grafiği çizilmiştir. Şekilde Lyapunov üstlerinden bir tanesi pozitif yörüngededir ve hücre kaotik davranış

göstermektedir. Şekil 3.9(d)'de $A=0.9$ için 2 tane Lyapunov üstlerinin grafiği çizilmiştir. Şekilde Lyapunov üstlerinin hepsi negatiftir ve hücre periyodik davranış göstermektedir.



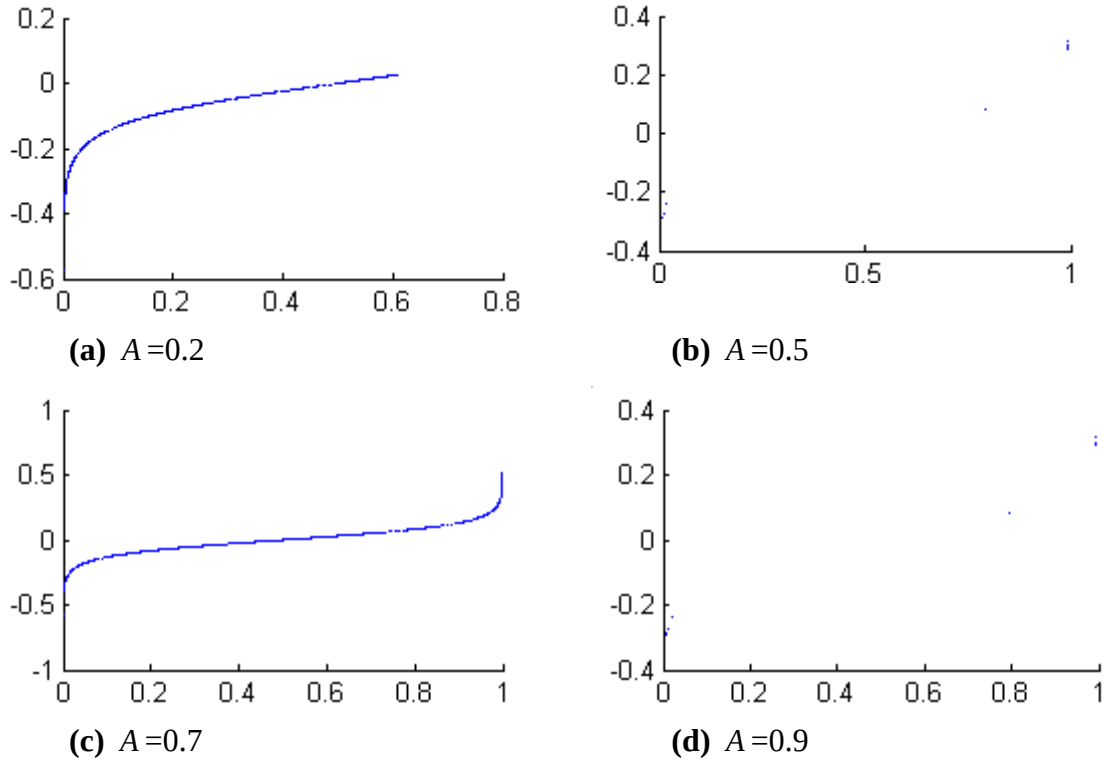
Şekil 3.9 (a) $A=0.2$ (b) $A=0.5$ (c) $A=0.7$ (d) $A=0.9$ Lyapunov üstlerinin grafiği

3.2.1.3 Poincare haritalama yöntemi

Hücrenin faz uzayı izlenirken belirli aralıklarla hücreden örnekler alınarak elde edilir. Kaotik hücrenin davranışı dört farklı aktivasyon değerleri için, Poincare Haritalama yöntemi ile incelenmiştir. Poincare Haritalama yöntemi ile incelen hücre sabit bir noktada kalıyorsa periyodik, kapalı bir çevrim görülüyorsa periyodumsu ve gelişi güzel tahmin edilemeyen bir şekil görünüyorsa kaotik davranış göstermektedir.

Şekil 3.10(a)'da $A=0.2$ değeri için hücre çıkışının poincare grafiği verilmiştir. Grafikte görüldüğü gibi tahmin edilemeyen şekil elde edilmiştir. Hücre kaotik davranış göstermiştir. Şekil 3.10(b)'de $A=0.5$ değeri için hücre çıkışının poincare grafiği verilmiştir. Grafikte görüldüğü gibi periyodik davranış elde edilmiştir. Şekil 3.10(c)'de $A=0.7$ değeri için hücre çıkışının poincare grafiği verilmiştir. Grafikte görüldüğü gibi tahmin edilemeyen şekil elde edilmiştir. Hücre kaotik davranış göstermiştir. Şekil

3.10(d)'de $A=0.9$ değeri için hücre çıkışının poincare grafiği verilmiştir. Grafikte görüldüğü gibi periyodik davranış elde edilmiştir.

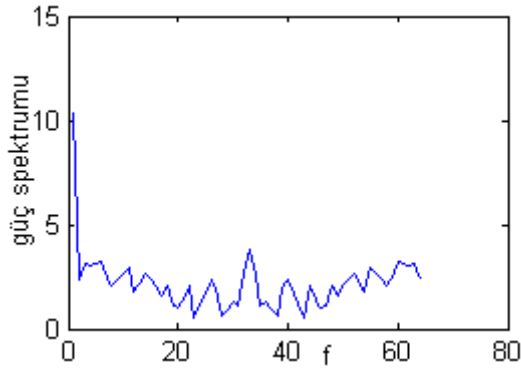


Şekil 3.10 (a) $A=0.2$ (b) $A=0.5$ (c) $A=0.7$ (d) $A=0.9$ Poincare grafiği

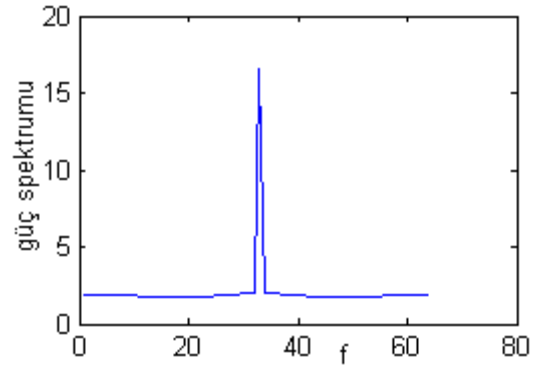
3.2.1.4 Güç spektrumu ile gösterim yöntemi

Hücrenin kaotik olup olmadığını belirlemede hücreye ait sinyalin güç spektrumuna bakılmıştır. Hücre kaotik davranış gösteriyorsa, sinyalin güç spektrumunda frekanslar arası sıçrama daha fazladır, periyodik davranış gösteriyorsa daha azdır.

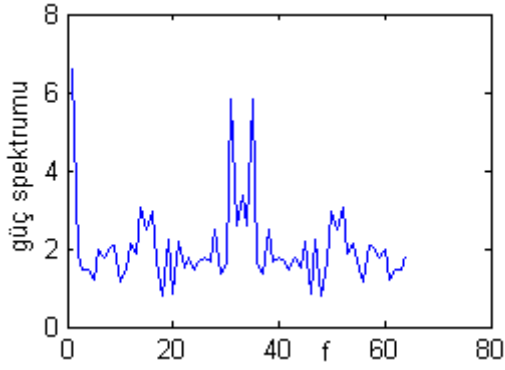
Şekil 3.11(a)'da $A=0.2$ için güç spektrumu grafiği verilmiştir. Sinyalin güç spektrumunda frekanslar arası sıçrama daha fazladır. Hücre kaotik davranış göstermektedir. Şekil 3.11(b)'de $A=0.5$ için güç spektrumu grafiği verilmiştir. Sinyalin güç spektrumunda frekanslar arası sıçrama daha azdır. Hücre periyodik davranış göstermektedir. Şekil 3.11(c)'de $A=0.7$ için güç spektrumu grafiği verilmiştir. Sinyalin güç spektrumunda frekanslar arası sıçrama daha fazladır. Hücre kaotik davranış göstermektedir. Şekil 3.11(d)'de $A=0.9$ için güç spektrumu grafiği verilmiştir. Sinyalin güç spektrumunda frekanslar arası sıçrama daha azdır. Hücre periyodik davranış göstermektedir.



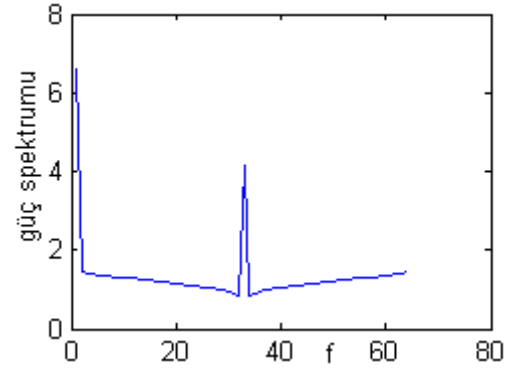
(a) $A=0.2$



(b) $A=0.5$



(c) $A=0.7$



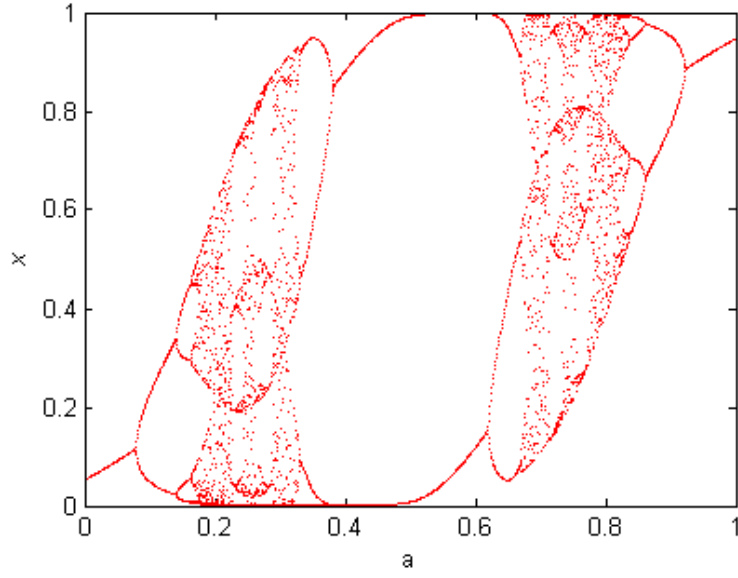
(d) $A=0.9$

Şekil 3.11 (a) $A=0.2$ (b) $A=0.5$ (c) $A=0.7$ (d) $A=0.9$ Güç spektrumu grafiği

3.2.1.5 Çatallaşma diyagramı yöntemi

A 'nın hücrenin çıkışına göre grafiği çizilmiştir. A parametresinin $[0-1]$ arasındaki değerlerinde hücrenin nasıl bir davranış gösterdiği tek bir grafikte verilmiştir.

Şekil 3.12'de kaotik hücrenin çatallaşma diyagramı grafiği verilmiştir. Şekilde görüldüğü gibi; A parametresinin $[0 - 0.2)$ arasındaki değerlerinde hücremiz kaotik değildir. Hücre periyodik ve artan değerlerde periyodumsu bir davranış göstermektedir. A parametresinin $[0.2-0.4)$ arasındaki değerleri için hücremiz kaosa girmiştir ve davranışı kestirilemez. A parametresinin $[0.5-0.7)$ arasındaki değerlerinde hücremiz kaotik değildir. Hücre periyodik ve periyodumsu bir davranış göstermektedir. A parametresinin $[0.7-0.9)$ arasındaki değerleri için hücremiz kaosa girmiştir ve davranışı kestirilemez. A parametresinin $[0.9-1]$ arasındaki değerlerinde hücremiz kaotik değildir. Hücre periyodumsu ve artan değerlerde periyodik bir davranış göstermektedir.

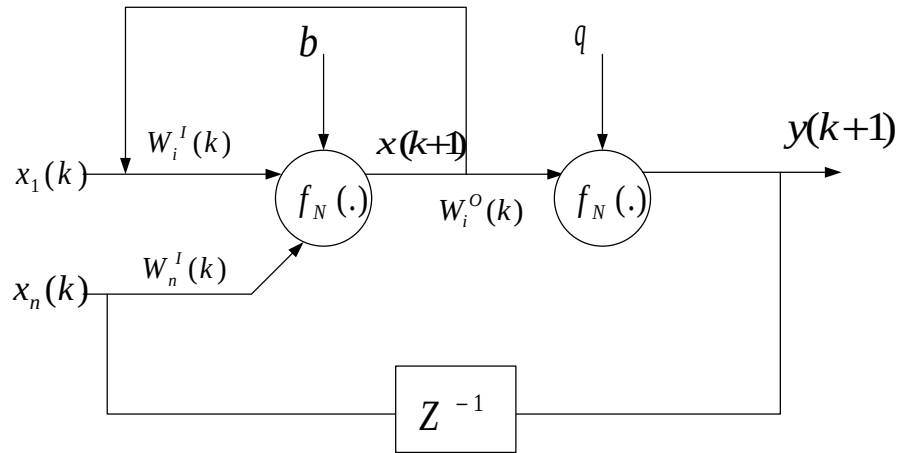


Şekil 3.12 Kaotik hücrenin çatallaşma diyagramı grafiği

3.2.2. İkinci Kaotik Hücrenin Kaotik Davranışının Analizi

Şekil 3.6’da verilen hücre birimi A aktivasyon fonksiyonuna göre kaotik davranış göstermektedir. Hücrenin kaotik durumunu oluşturan başka faktörler de bulunmaktadır.

Şekil 3.13’de örnek başka bir kaotik hücre verilmiştir. Hücrenin kaotik karakteristiğini belirleyen değişken ise toplam fonksiyonun aktivasyondan geçmesinden önce çarpılan a değişkenidir. Şekil 3.13’deki kaotik hücrenin durumu incelenecek ve kaosu ortaya çıkaran yöntemlerle kaotik ya da periyodik davranışları gösterilecektir.



Şekil 3.13 Kaotik hücre birimi

Şekil 3.13'deki kaotik hücre birimine ait matematiksel ifadeler denklem 3.22, 3.23, 3.24 ve 3.25'de verilmiştir.

$$S_j^H = \sum_{j=1}^n W_{ij}^I . x_j(k) + b_j(k) \quad (3.22)$$

$$x_j(k+1) = f_N(a.S_j^H) \quad (3.23)$$

$$S_j^O = \sum_{j=1}^m W_{ij}^O . x_j(k+1) + q_j(k) \quad (3.24)$$

$$y_j(k+1) = f_N(a.S_j^O) \quad (3.25)$$

Yukarıda verilen denklemde değişkenler;

$x(k)$ = Hücrenin girişi,

$x(k+1)$ = Ara katmanın çıkışı,

$S^H(k)$ = Orta katmandaki girişlerin toplam ağırlığı,

$W^I(k)$ = Giriş katmanının ağırlık vektörü,

$f_N(.)$ = Doğrusal olmayan sigmoid fonksiyonu,

$W^O(k)$ = Çıkış katmanının ağırlık vektörü,

$S_j^O(k)$ = Çıkış katmandaki girişlerin toplam ağırlığı,

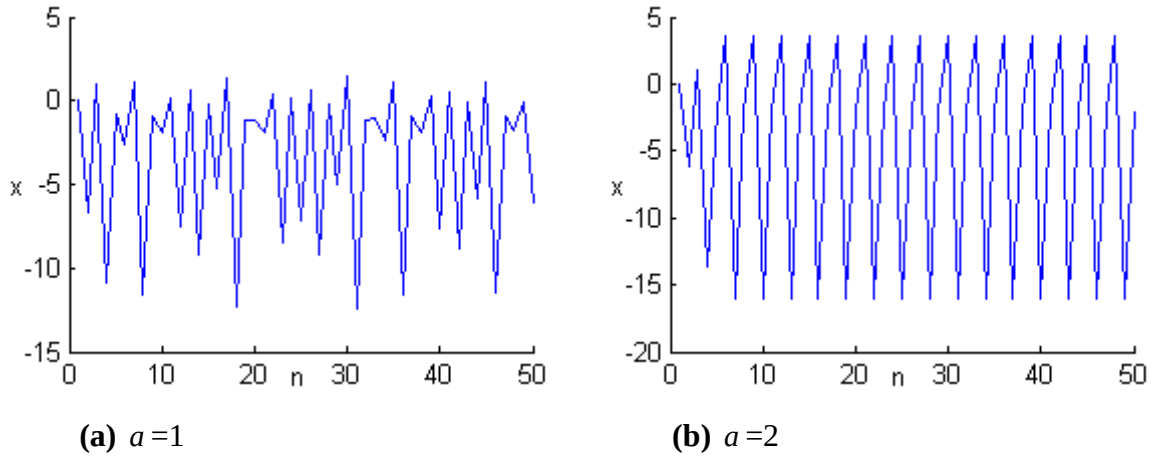
$y(k)$ = Hücrenin çıkışıdır.

Hücrenin çıkışı a parametresine göre değişmektedir. $a=1$ için hücre kaotik davranış gösterirken, $a=2$ için sistem periyodik davranış göstermektedir. Kaosu ortaya çıkaran yöntemlerle hücrenin kaotik ve periyodik davranışları incelenmiş olup hücrenin davranışı grafiklerle gösterilmiştir.

3.2.2.1 Yörüngelerin İzlenmesi Yöntemi

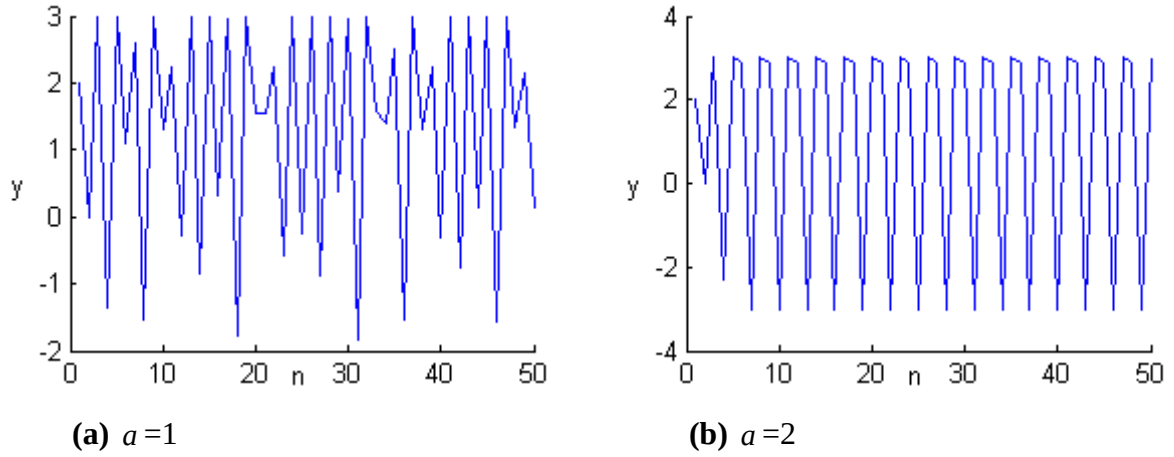
x eksenine zaman, y eksenine hücrenin çıkışı girilerek çizilen grafikte, hücrenin kaosa girip girmediğine bakılabilir.

Yörüngelerin izlenmesi yöntemine göre a 'nın iki farklı değeri için $x-n$ grafiğini çizilmiştir. Şekil 3.14(a)'da $a=1$ değeri için hücrenin çıkışının x -zaman ekseninde grafiği çizilmiş, hücrenin kaotik davranışı gösterilmiştir. Şekil 3.14(b)'de $a=2$ değeri için hücrenin çıkışının x -zaman ekseninde grafiği çizilmiş, hücrenin zaman ekseninde periyodik davranış gösterilmiştir.



Şekil 3.14 (a) $a=1$ (b) $a=2$ $x-n$ grafiği

Şekil 3.15(a)'da $a=1$ değeri için hücrenin çıkışının y -zaman ekseninde grafiği çizilmiş, hücrenin kaotik davranışı gösterilmiştir. Şekil 3.15(b)'de $a=2$ değeri için hücrenin çıkışının y -zaman ekseninde grafiği çizilmiş, hücrenin zaman ekseninde periyodik davranış gösterilmiştir.

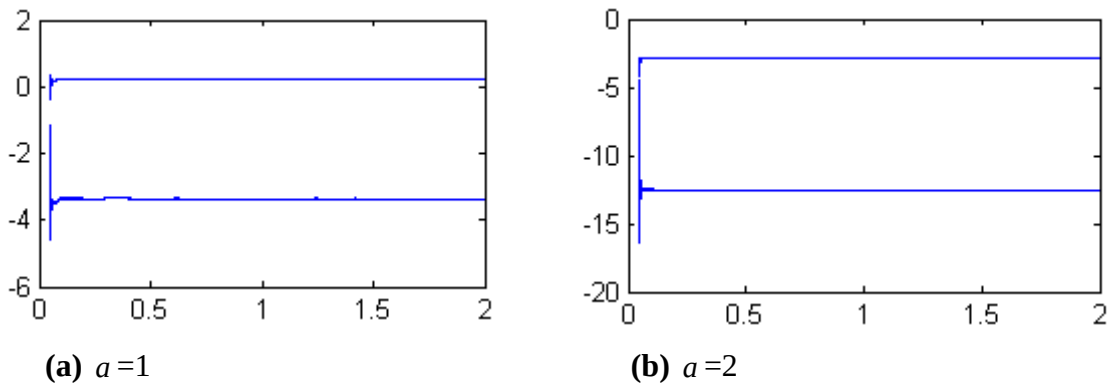


Şekil 3.15 (a) $a=1$ (b) $a=2$ $y-n$ grafiği

3.2.2.2 Lyapunov üstleri

Kaotik hücrenin davranışı iki tane Lyapunov değerine göre incelenmiştir. Eğer Lyapunov üstlerinin hepsi negatif yörüngede ise hücre kaotik olmayan bir davranış gösterecektir. Lyapunov üstlerinden biri pozitif yörüngede ise hücre kaotik bir davranış gösterecektir. Grafiğin çizilmesinde *Matlab Lynch_2007* kullanılmıştır.

Şekil 3.16(a)'da $a=2$ için 2 tane Lyapunov üstlerinin grafiği çizilmiştir. Şekilde Lyapunov üstlerinden bir tanesi pozitif yörüngededir ve sistem kaotik davranış göstermektedir. Şekil 3.16(b)'de $a=1$ için 2 tane Lyapunov üstlerinin grafiği çizilmiştir. Şekilde Lyapunov üstlerinin hepsi negatiftir ve sistem periyodik davranış göstermektedir.

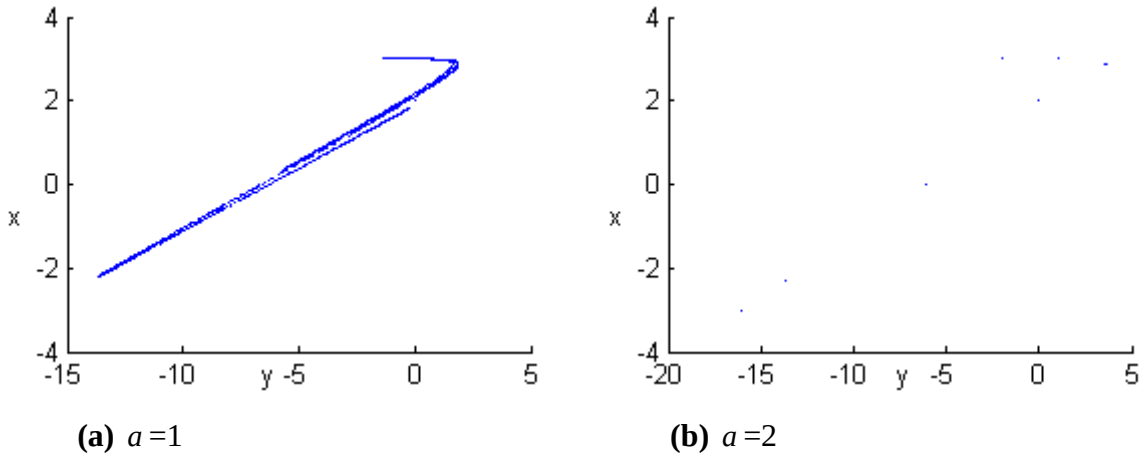


Şekil 3.16 (a) $a=1$ (b) $a=2$ Lyapunov üstlerinin grafiği

3.2.2.3 Poincare haritalama yöntemi

Hücresinin faz uzayı izlenirken belirli aralıklarla hücreden örnekler alınarak elde edilir. Kaotik hücrenin davranışı a 'nın iki farklı değeri için, Poincare Haritalama yöntemi ile incelenmiştir. Poincare Haritalama yöntemi ile incelen hücre sabit bir noktada kalıyorsa periyodik, kapalı bir çevrim görülüyorsa periyodumsu ve gelişi güzel tahmin edilemeyen bir şekil görünüyorsa kaotik davranış göstermektedir.

Şekil 3.17(a)'da $a=1$ değeri için hücrenin çıkışının poincare grafiği verilmiştir. Grafikte görüldüğü gibi tahmin edilemeyen şekil elde edilmiştir. Hücre kaotik davranış göstermiştir. Şekil 3.17(b)'de $a=2$ değeri için hücrenin çıkışının poincare grafiği verilmiştir. Grafikte görüldüğü gibi periyodik davranış elde edilmiştir.



(a) $a=1$

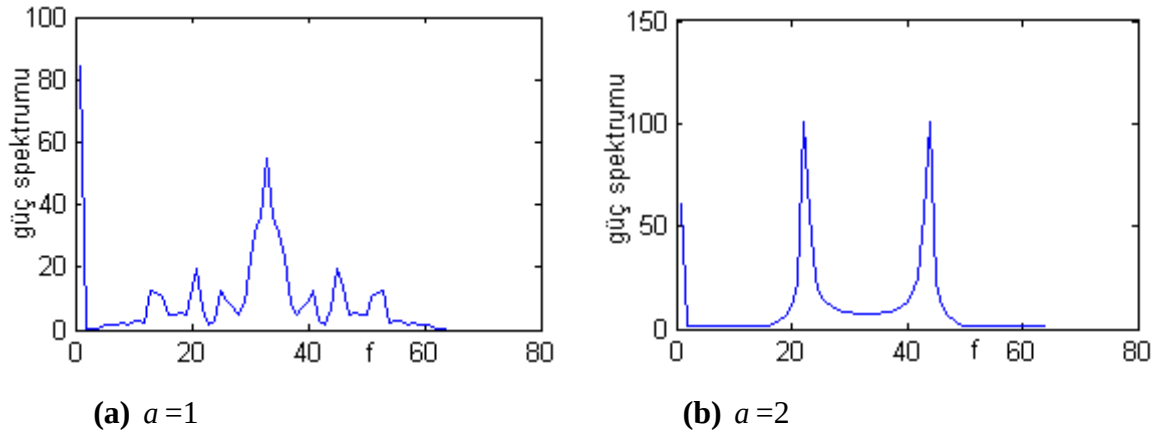
(b) $a=2$

Şekil 3.17 (a) $a=1$ (b) $a=2$ Poincare grafiği

3.2.2.4 Güç spektrumu ile gösterim yöntemi

Hücresinin kaotik olup olmadığını belirlemede hücreye ait sinyalin güç spektrumuna bakılmıştır. Hücre kaotik davranış gösteriyorsa, sinyalin güç spektrumunda frekanslar arası sıçrama daha fazladır, periyodik davranış gösteriyorsa daha azdır.

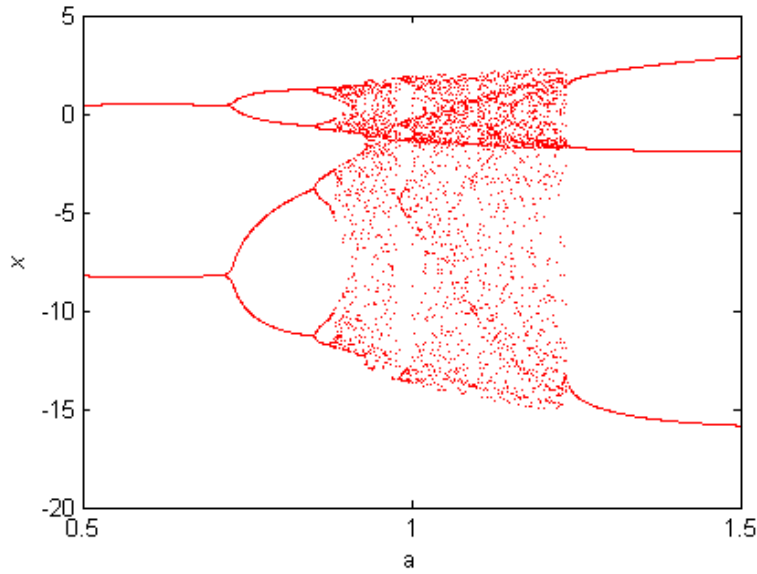
Şekil 3.18(a)'da $a=1$ için Güç spektrumu grafiği verilmiştir. Sinyalin güç spektrumunda frekanslar arası sıçrama daha fazladır. Hücre kaotik davranış göstermektedir. Şekil 3.18(b)'de $a=2$ için Güç spektrumu grafiği verilmiştir. Sinyalin güç spektrumunda frekanslar arası sıçrama daha azdır. Hücre periyodik davranış göstermektedir.



Şekil 3.18 (a) $a=1$ (b) $a=2$ $a=2$ Güç spektrumu grafiği

3.2.2.5 Çatallaşma diyagram yöntemi

a 'nın hücrenin çıkışına göre grafiği çizilmiştir. a parametresinin $[0.5-1.5]$ arasındaki değerlerinde hücrenin nasıl bir davranış gösterdiği tek bir grafikte verilmiştir.



Şekil 3.19 Kaotik hücrenin çatallaşma diyagramı grafiği

Şekil 3.19'da kaotik hücrenin çatallaşma diyagramı grafiği verilmiştir. Şekilde görüldüğü gibi; a parametresinin $[0 - 0.9)$ arasındaki değerlerinde hücremiz kaotik değildir. Hücre periyodik ve artan değerlerde periyodumsu bir davranış göstermektedir. a parametresinin $[0.9-1.2)$ arasındaki değerleri için sistem kaosa girmiştir ve davranışı

kestirilemez. a parametresinin [1.2-1.5) arasındaki deęerlerinde hücremiz kaotik deęildir. Hücre periyodik ve periyodumsu bir davranış göstermektedir.

3.3 Kaotik Yapay Sinir Ağ Yapıları

Kaotik yapay sinir ağları, kaotik nöronların bir araya getirilmesi ile oluşan yapay sinir ağlarıdır. Kaotik yapay sinir ağlarının yapısı 4.bölümde ele alınacaktır.

4. YAPAY SİNİR AĞLARI İLE SİSTEM MODELLEME

Sistem modelleme, uygulamalarla veya matematik ifadelerle elde edilmiş verilerden faydalanarak sistemlerin modellerinin elde edilmesidir [43]. Sistem modellemede amaç bilinmeyen bir sistemin transfer fonksiyonun yani sistemin yapısının belirlenmesidir.

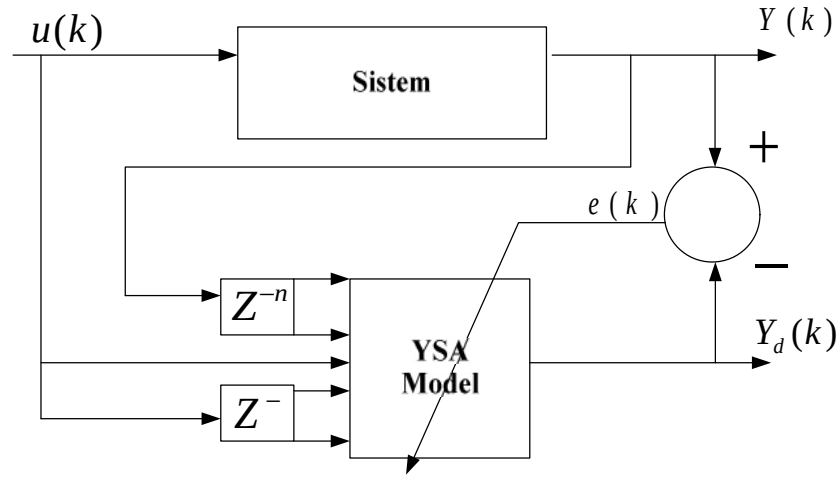
Sistem tanıma 4 aşamadan oluşur [44,45]

- 1.Sistemin giriş - çıkış bilgilerinin elde edilmesi
- 2.Model yapısının seçimi.
- 3.Model parametrelerinin kestirimi.
- 4.Modelin yapısının ve parametrelerinin uygunluğunun testi.

Sistem tanımlamada tanınacak sisteme uygulanacak işaretler çok önemlidir. Sistemin girişleri tüm modları uyarabilmelidir. Bu amaçla sistemin girişleri frekans yönünden çok zengin olmalıdır. Tüm frekansları içerdiği için rastgele işaretler kullanılmalıdır. Lineer sistemlerde girişin genliği iki seviyeli olması yeterli olurken, lineer olmayan sistemlerin giriş genliği rastgele seçilmelidir. Literatürde sistem modelleme için kullanılan birçok algoritma ve uygulamaları mevcuttur.

4.1 Düz modelleme

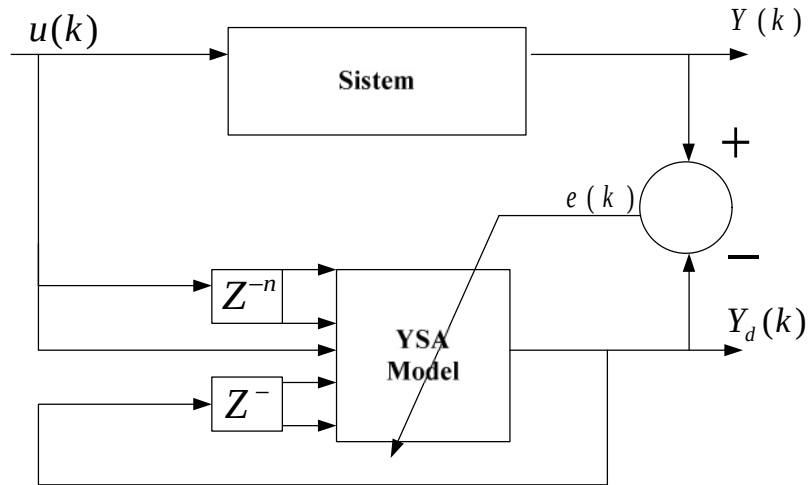
Düz modelleme, bir sistemin düz yönlü dinamiklerini gösteren yapay sinir ağı modelinin seçilmesi ve bu ağı eğitilmesidir. Şekil 4.1’de NARX (seri-paralel model yapısına göre bir sistemin yapay sinir ağı modeli) verilmiştir.



Şekil 4.1 Yapay sinir ağı ile düz modelleme(NARX model)

İleri beslemeli, katmanlı yapay sinir ağı ile elde edilen bir sistemin NARX modeli, yapay sinir ağındaki hataya göre geriye yayılım algoritmasıyla eğitilir.

Denetlenen sistemin dinamik davranışlarının model ve denetleyici yapay sinir ağında tam olarak temsil edilmesi dinamiklerin yapay sinir ağı yapısında tanımlanmaları ile sağlanır. Bu sistemler için geri beslemeli yapay sinir ağları kullanılır. Şekil 4.2’de NOE modeli(yapay sinir ağı ile düz modelleme) verilmiştir.



Şekil 4.2 Yapay sinir ağı ile düz modelleme (NOE model)

4.2 YSA ile sistem modellemede kullanılan örnek sistemler

YSA ile sistem modellemede iki örnek sistem kullanılacaktır. Bu sistemlerden biri tek giriş - tek çıkışlı sistem olup, ikinci örnek sistem ise MIMO (çok giriş - çok çıkışlı) sistemidir.

4.2.1 Tek Giriş - Tek Çıkışlı Sistem Örneği

Yapay sinir ağları ile sistem modellemede kullanılacak birinci sistemin matematiksel ifadesi denklem 4.1 ve 4.2’de verilmiştir [11].

Sistemin Denklemi:

$$y^p(k+1) = \frac{y^p(k)}{1 + y^p(k)} + u^3(k) \quad (4.1)$$

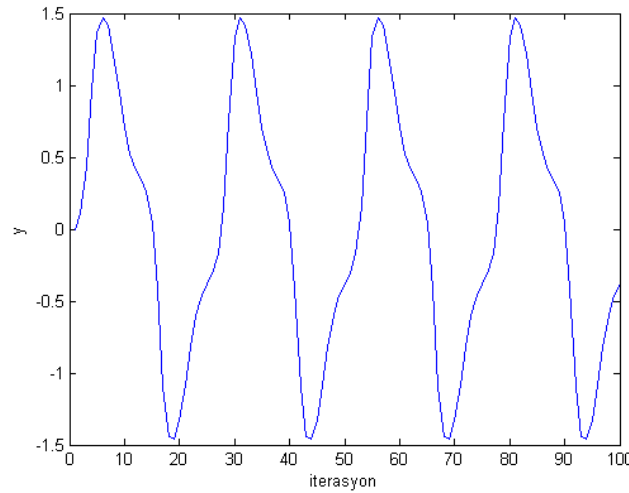
$$u = \sin(2.k.\pi / 25) \quad (4.2)$$

Yukarıdaki ifadelerle göre;

y^p = Sistemin çıkışı

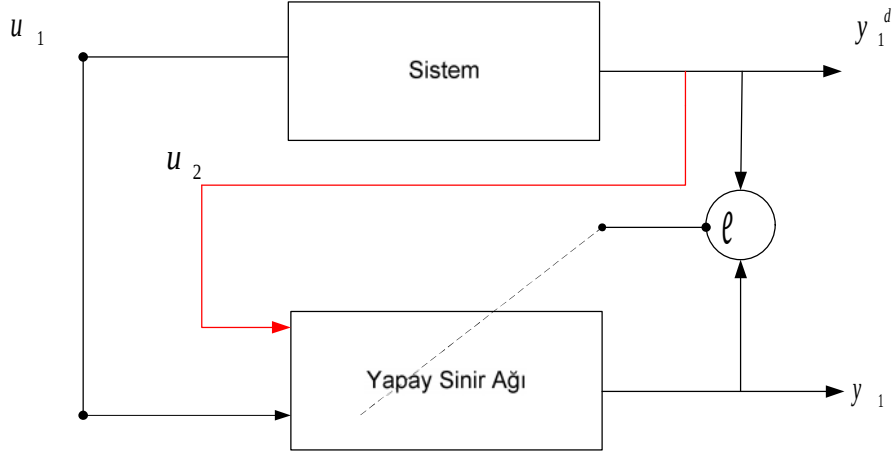
u = Sistemin girişini ifade eder.

Matematiksel ifadesi yukarıda verilen örnek sistemin grafiği şekil 4.3’de verilmiştir



Şekil 4.3 Birinci örnek sistemin grafiği

Denklem 4.1 ve 4.2’de matematiksel olarak ifade edilen sistemin YSA ile modellenmesi Şekil 4.4’de verilmiştir.



Şekil 4.4 YSA ile örnek bir sistemin modellenmesi

Yukarıdaki şekle göre;

u_1, u_2 = YSA’nın girişleri,

y_1^d = Sistemin çıkışı,

y_1 = YSA’nın çıkışı,

e = Hata.

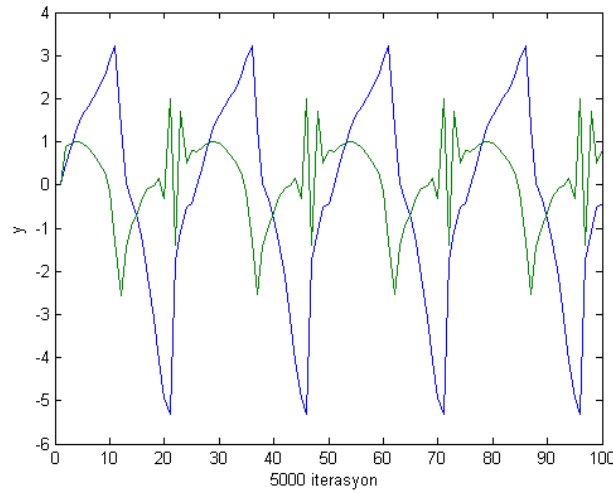
4.2.2 Çok Giriş - Çok Çıkışlı Sistem Örneği

İkinci sistem modellemede ise MIMO (çoklu girişli - çoklu çıkışlı sistem) dinamiği kullanılacaktır [11]. Sistemin matematiksel ifadesi denklem 4.3ve 4.4’de verilmiştir.

$$\begin{bmatrix} y_{p1}(k+1) \\ y_{p2}(k+2) \end{bmatrix} = \begin{bmatrix} \frac{y_{p1}(k)}{1 + y_{p2}^2(k)} \\ \frac{y_{p1}(k) + y_{p2}(k)}{1 + y_{p2}^2(k)} \end{bmatrix} + \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix} \quad (4.3)$$

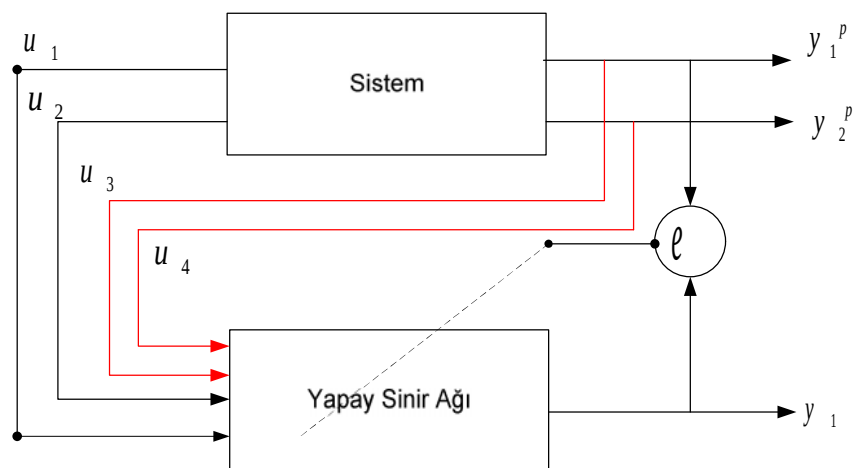
$$\begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix} = \begin{bmatrix} (\sin(2.\pi.k / 25)) \\ (\cos(2.\pi.k / 25)) \end{bmatrix} \quad (4.4)$$

Matematiksel ifadesi yukarıda verilen örnek sistemin grafiği şekil 4.5’de verilmiştir.



Şekil 4.5 İkinci örnek sistemin grafiği

Denklem 4.3 ve 4.4’de matematiksel olarak ifade edilen sistemin YSA ile modellenmesi şekil 4.6’da verilmiştir.



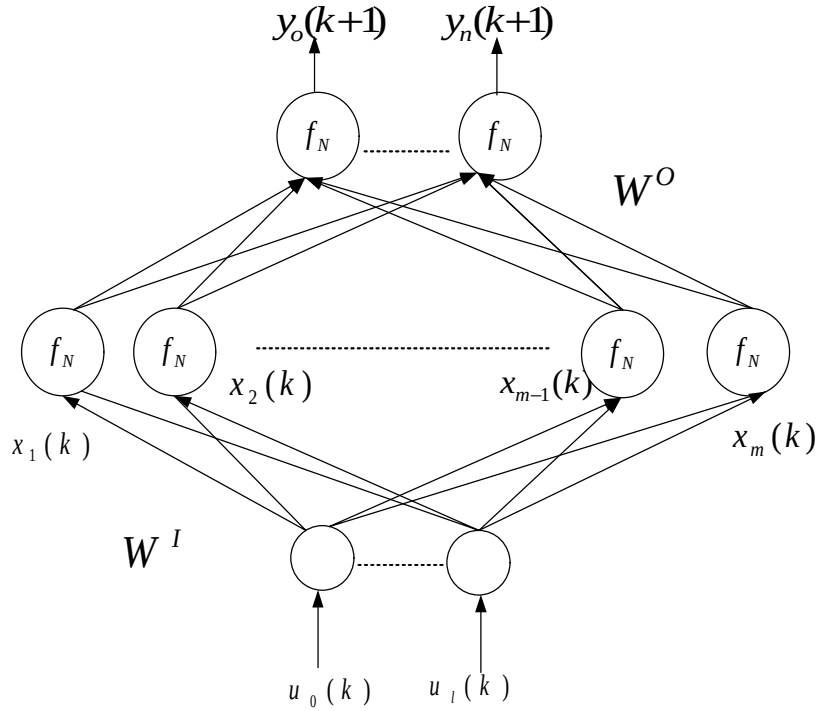
Şekil 4.6 YSA ile örnek bir sistemin modellenmesi

4.3.1 Sistem modellemede kullanılan YSA yapıları

YSA ile sistem modellemede dört çeşit yapay sinir ağ yapıları kullanılacaktır. Düz ileri beslemeli, öz geri beslemeli, tam geri beslemeli ve kaotik yapay sinir ağlarıdır. Bu YSA'ların yapısı, matematiksel denklemleri verilmiş ve geriye yayılım algoritması ile eğitimleri yapılmıştır.

4.3.1.1 İleri beslemeli YSA

Ağırlıkların sabit olduğu ve hücrede geri besleme ya da geciktirilmiş sinyallerin kullanılmadığı, yapay sinir ağlarıdır. Düz ileri beslemeli yapay sinir ağının modeli şekil 4.7'de verilmiştir.



Şekil 4.7 Düz ileri beslemeli yapay sinir ağının yapısı

Şekil 4.7'de verilen ileri beslemeli yapay sinir ağının matematiksel ifadeleri denklem 4.5, 4.6, 4.7 ve 4.8'de verilmiştir.

$S_j^H(k)$ 'ın $x_j(k)$ ve hesaplanması;

Orta katmandaki j. nöronun ağırlıkların toplamı denklem 4.5'de verilmiştir.

$$S_j^H(k) = \sum_{i=1}^l W_j^I u_i(k) \quad (4.5)$$

Orta katmandaki j. nöronun çıkışı denklem 4.6'da verilmiştir.

$$x_j(k) = f_N[S_j^H(k)] \quad (4.6)$$

Denklem 4.5 ve 4.6'ya göre;

$u_i(k)$ = Her bir farklı k zamanında i. giriş,

$W^I(k)$ = Giriş katmanının ağırlık vektörü,

$S_j^H(k)$ = Orta katmandaki girişlerin toplam ağırlığı,

$x_j(k)$ = Orta katmandaki j. nöronun çıkışı,

$f_N(\cdot)$ = Doğrusal olmayan sigmoid fonksiyonu.

Çıkış katmandaki j. nöronun ağırlıkların toplamı denklem 4.7'de verilmiştir.

$$S_p^O(k) = \sum_{j=1}^m W_{ij}^O x_j(k) \quad (4.7)$$

Çıkış katmandaki j. nöronun çıkışı denklem 4.8'de verilmiştir.

$$y(k) = f_N[S_j^O(k)] \quad (4.8)$$

Denklem 4.7 ve 4.8'e göre;

$y_j(k)$ = Her bir farklı k. zamanında j. çıkış,

$W^O(k)$ = Çıkış katmanının ağırlık vektörü,

$S_j^o(k)$ = Çıkış katmandaki girişlerin toplam ağırlığı,

$x_j(k)$ = Orta katmandaki j.nöronun çıkışıdır.

Dinamik öğrenme yönteminin matematiksel ifadesi denklem 4.9’da verilmiştir.

$$W(k+1) = W(k) - a \cdot \nabla_w E(k) \quad (4.9)$$

Denklem 4.9’a göre;

$W(k)$ = Ağırlık vektörü,

a = Öğrenme oranı,

∇_w = Ağırlıklara uygulanacak düzeltme miktarı,

$E(k)$ = Hata indeksidir.

YSA’nın eğitiminde hataların kareleri ile tanımlanan ortalama – karesel hata ifadesi denklem 4.10, 4.11 ve 4.12’de verilmiştir.

$$E(k) = \frac{1}{2} \sum_{i=1}^n [y_i^d(k) - y_i(k)]^2 \quad (4.10)$$

$$E(k) = \frac{1}{2} \sum_{i=1}^n e_i^2(k) \quad (4.11)$$

Çıkışlar arasındaki hata:

$$e_i(k) = y_i^d(k) - y_i(k) \quad (4.12)$$

W^o ağırlığının yenilenmesi için ilgili ağırlığa göre türevinin alınmasını gerektirir. Buna göre;

$$\frac{\partial E(k)}{\partial W^O(k)} = \frac{\partial E(k)}{\partial e(k)} \cdot \frac{\partial e(k)}{\partial y(k)} \cdot \frac{\partial y(k)}{\partial S^O(k)} \cdot \frac{\partial S^O(k)}{\partial W^O(k)} \quad (4.13)$$

hücresinin matematiksel denklemlerinden yararlanarak türevler alınır ;

$$\frac{\partial E(k)}{\partial W^O(k)} = -e(k) \cdot f_N[S^O(k)] \cdot x(k) \quad (4.14)$$

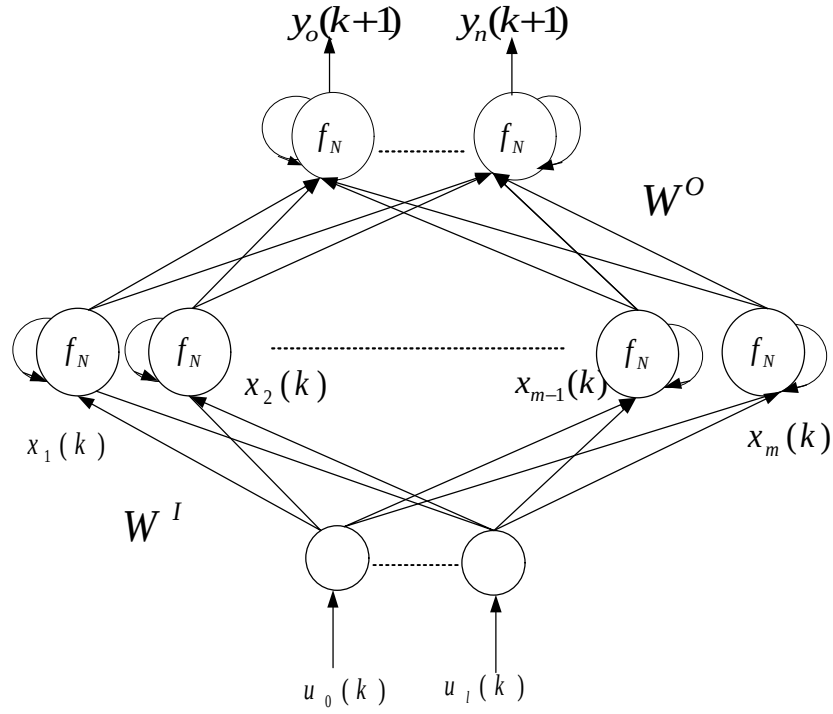
W^I ağırlığının yenilenmesi için ilgili ağırlığa göre türevinin alınması gerekir. Buna göre;

$$\frac{\partial E(k)}{\partial W^I(k)} = \frac{\partial E(k)}{\partial e(k)} \cdot \frac{\partial e(k)}{\partial y(k)} \cdot \frac{\partial y(k)}{\partial S^O(k)} \cdot \frac{\partial S^O(k)}{\partial x(k)} \cdot \frac{\partial x(k)}{\partial S^H(k)} \cdot \frac{\partial S^H(k)}{\partial W^I(k)} \quad (4.15)$$

$$\frac{\partial E(k)}{\partial W^I(k)} = -e(k) \cdot f_N[S^O(k)] \cdot W^O(k) \cdot f_N[S^H(k)] \cdot u(k) \quad (4.16)$$

4.3.2 Öz Geri Beslemeli Yapay Sinir Ağları

Ağırlıkların sabit olduğu hücrede, hücreye gelen girişlerin toplamının geri beslemesinin K katsayısı ile çarpılarak elde edilen değerin hücreye giriş olarak verildiği yapay sinir ağıdır. Öz geri beslemeli yapay sinir ağının modeli şekil 4.8’de verilmiştir.



Şekil 4.8 Öz geri beslemeli yapay sinir ağının yapısı

Şekil 4.8’de verilen öz geri beslemeli yapay sinir ağının matematiksel ifadeleri denklem 4.17, 4.18, 4.19 ve 4.20’de verilmiştir.

$S_j^H(k)$ ’ın $x_j(k)$ ve hesaplanması;

Orta katmandaki j. nöronun ağırlıkları toplamı denklem 4.17’de verilmiştir.

$$S_j^H(k) = \sum_{i=1}^I W_j^I u_i(k) + K.S_j^H(k-1) \quad (4.17)$$

Orta katmandaki j. nöronun çıkışı denklem 4.18’de verilmiştir.

$$x_j(k) = f_N[S_j^H(k)] \quad (4.18)$$

Denklem 4.17 ve 4.18’e göre;

$u_i(k)$ = Her bir farklı k zamanında i. giriş,

$W^I(k)$ = Giriş katmanının ağırlık vektörü,

$S_j^H(k)$ = Orta katmandaki girişlerin toplam ağırlığıdır,

$x_j(k)$ = Orta katmandaki j. nöronun çıkışı,

$f_N(.)$ = Doğrusal olmayan sigmoid fonksiyonu,

K = İnatçı parametre.

Çıkış katmandaki j. nöronun ağırlıkların toplamı denklem 4.19'da verilmiştir.

$$S_p^O(k) = \sum_{j=1}^m W_{ij}^O x_i(k) + K.S_p^O(k-1) \quad (4.19)$$

Çıkış katmandaki j. nöronun çıkışı denklem 4.20'de verilmiştir.

$$y_j(k) = f_N[S_j^O(k)] \quad (4.20)$$

Denklem 4.19 ve 4.20'ye göre;

$y_j(k)$ = Her bir farklı k zamanında j. çıkış,

$W^O(k)$ = Çıkış katmanının ağırlık vektörü,

$S_j^O(k)$ = Çıkış katmandaki girişlerin toplam ağırlığıdır,

$x_j(k)$ = Orta katmandaki j. nöronun çıkışı.

Dinamik öğrenme yönteminin matematiksel ifadesi denklem 4.21'de verilmiştir.

$$W(k+1) = W(k) - a.\nabla_w E(k) \quad (4.21)$$

Denklem 4.21'e göre;

$W(k)$ = Ağırlık vektörü,

a = Öğrenme oranı,

∇_w = Ağırlıklara uygulanacak düzeltme miktarı,

$E(k)$ = Hata indeksi.

YSA'nın eğitiminde hataların kareleri ile tanımlanan ortalama – karesel hata ifadesi denklem 4.22, 4.23 ve 4.24'de verilmiştir.

$$E(k) = \frac{1}{2} \sum_{i=1}^n [y_i^d(k) - y_i(k)]^2 \quad (4.22)$$

$$E(k) = \frac{1}{2} \sum_{i=1}^n e_i^2(k) \quad (4.23)$$

Çıkışlar arasındaki hata,

$$e_i(k) = y_i^d(k) - y(k) \quad (4.24)$$

Ağırlıkların yenilenmesidir.

Hücrenin matematiksel denklemlerinden yararlanarak türevleri alınırsa;

W^O ağırlığının yenilenmesi için ilgili ağırlığa göre türevinin alınması gerekir. Buna göre;

$$\frac{\partial E(k)}{\partial W^O(k)} = \frac{\partial E(k)}{\partial e(k)} \cdot \frac{\partial e(k)}{\partial y(k)} \cdot \frac{\partial y(k)}{\partial S^O(k)} \cdot \frac{\partial S^O(k)}{\partial W^O(k)} \quad (4.25)$$

$$\frac{\partial E(k)}{\partial S^O(k)} = -e(k) \cdot f_N[S^O(k)] \cdot A_j^O(k-1) \quad (4.26)$$

W^I ağırlığının yenilenmesi için ilgili ağırlığa göre türevinin alınması gerekir. Buna göre;

$$\frac{\partial E(k)}{\partial W^I(k)} = \frac{\partial E(k)}{\partial e(k)} \cdot \frac{\partial e(k)}{\partial y(k)} \cdot \frac{\partial y(k)}{\partial S^O(k)} \cdot \frac{\partial S^O(k)}{\partial x(k)} \cdot \frac{\partial x(k)}{\partial S^H(k)} \cdot \frac{\partial S^H(k)}{\partial W^I(k)} \quad (4.27)$$

$$\frac{\partial E(k)}{\partial W^I(k)} = -e(k) \cdot f_N[S^O(k)] \cdot W^O(k) \cdot f_N[S^H(k)] \cdot A_j^I(k-1) \quad (4.28)$$

$$A_{ij}^O(k) = O_i(k-1) + [K].A_{ij}^O(k-1)$$

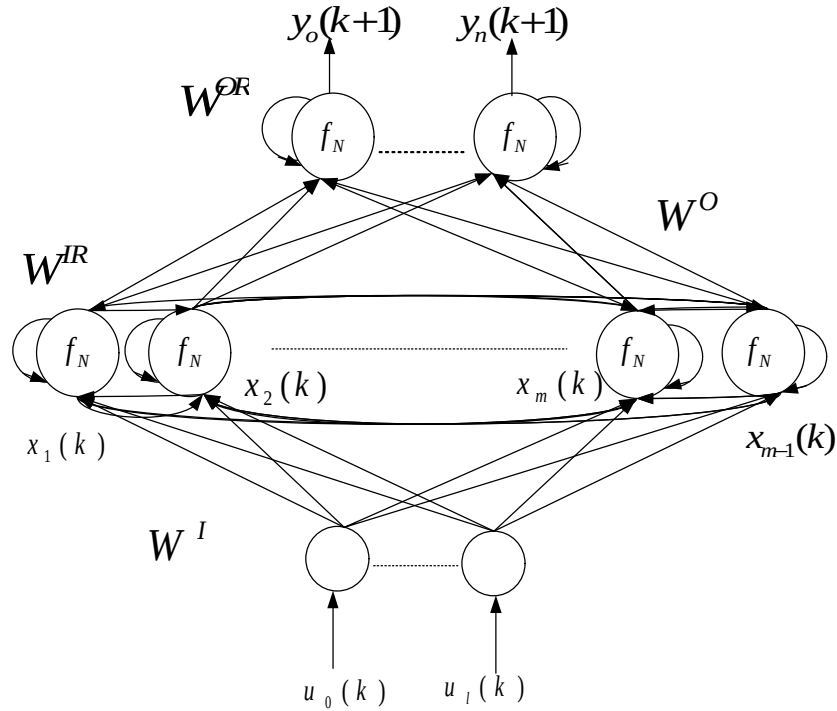
$$A_{ij}^O(0) = 0,$$
(4.29)

$$A_{ij}^I(k) = O_i(k-1) + [K].A_{ij}^{OR}(k-1)$$

$$A_{ij}^I(0) = 0,$$
(4.30)

4.3.3 Tam Geri Beslemeli Yapay Sinir Ağı

Tam geri beslemeli yapay sinir ağlarında iki geri besleme vardır. Birinci geri besleme, ağırlıkların sabit olduğu ve hücrede hücreye gelen girişlerin toplamının geri beslemesinin K katsayısı ile çarpılarak elde edilen değerin hücreye giriş olarak verildiği geri beslemedir. İkinci geri besleme ise orta katmandaki hücre çıkışının kendisine ve orta katmandaki diğer hücrelere giriş olarak verildiği geri beslemedir. Tam geri beslemeli yapay sinir ağının modeli şekil 4.9’da verilmiştir.



Şekil 4.9 Tam geri beslemeli yapay sinir ağının yapısı

Şekil 4.9’da verilen tam geri beslemeli yapay sinir ağının matematiksel ifadeleri denklem 4.31, 4.32, 4.33 ve 4.34’de verilmiştir.

$S_j^H(k)$ ’ın $x_j(k)$ ve hesaplanması;

Orta katmandaki j. nöronun ağırlıkları toplamı denklem 4.31’de verilmiştir.

$$S_j^H(k) = \sum_{i=1}^l W_{ji}^I u_i(k) + \sum_{q=1}^m W_{jq}^{IR} .x_q(k) + K.S_j^H(k-1) \quad (4.31)$$

Orta katmandaki j. nöronun çıkışı denklem 4.32’de verilmiştir.

$$x_j(k) = f_N[S_j^H(k)] \quad (4.32)$$

Denklem 4.31 ve 4.32’ye göre;

$u_i(k)$ = Her bir farklı k zamanında i.giriş,

$W^I(k)$ = Giriş katmanının ağırlık vektörü,

$W^{IR}(k)$ = Orta katmandaki hücreler arası ağırlık vektörü,

$S_j^H(k)$ = Orta katmandaki girişlerin toplam ağırlığı,

$x_j(k)$ = Orta katmandaki j. nöronun çıkışı,

$f_N(.)$ = Doğrusal olmayan sigmoid fonksiyonu,

K = İnatçı parametredir.

Çıkış katmandaki j. nöronun ağırlıkları toplamı denklem 4.33’de verilmiştir.

$$S_p^O(k) = \sum_{j=1}^m W_{jp}^O x_j(k) + K.S_p^O(k-1) \quad (4.33)$$

Çıkış katmandaki j. nöronun çıkışı denklem 4.34’de verilmiştir.

$$y(k) = f_N[S_p^O(k)] \quad (4.34)$$

Denklem 4.35 ve 4.36’ye göre;

$y_j(k)$ = Her bir farklı k zamanında j. çıkış,

$W^o(k)$ = Çıkış katmanının ağırlık vektörü,

$S_j^o(k)$ = Çıkış katmandaki girişlerin toplam ağırlığı,

$x_j(k)$ = Orta katmandaki j. nöronun çıkışıdır.

Dinamik öğrenme yönteminin matematiksel ifadesi denklem 4.35’de verilmiştir.

$$W(k+1) = W(k) - a \cdot \nabla_w E(k) \quad (4.35)$$

Denklem 4.35’e göre;

$W(k)$ = Ağırlık vektörü,

a = Öğrenme oranı,

∇_w = Ağırlıklara uygulanacak düzeltme miktarı,

$E(k)$ = Hata indeksidir.

YSA’nın eğitiminde hataların kareleri ile tanımlanan ortalama – karesel hata ifadesi denklem 4.36, 4.37 ve 4.38’de verilmiştir.

$$E(k) = \frac{1}{2} \sum_{i=1}^n [y_i^d(k) - y_i(k)]^2 \quad (4.36)$$

$$E(k) = \frac{1}{2} \sum_{i=1}^n e_i^2(k) \quad (4.37)$$

Çıkışlar arasındaki hata:

$$e_i(k) = y_i^d(k) - y_i(k) \quad (4.38)$$

Ağırlıkların yenilenmesi

Hücrenin matematiksel denklemlerinden yararlanarak türevler alınır;

W^o ağırlığının yenilenmesi için ilgili ağırlığa göre türevinin alınması gerekir. Buna göre;

$$\frac{\partial E(k)}{\partial W^O(k)} = \frac{\partial E(k)}{\partial e(k)} \cdot \frac{\partial e(k)}{\partial y(k)} \cdot \frac{\partial y(k)}{\partial S^O(k)} \cdot \frac{\partial S^O(k)}{\partial W^O(k)} \quad (4.39)$$

$$\frac{\partial E(k)}{\partial S^O(k)} = -e(k) \cdot f_N[S^O(k)] \cdot A_j^O(k-1) \quad (4.40)$$

W^I ağırlığının yenilenmesi için ilgili ağırlığa göre türevinin alınması gerekir. Buna göre;

$$\frac{\partial E(k)}{\partial W^I(k)} = \frac{\partial E(k)}{\partial e(k)} \cdot \frac{\partial e(k)}{\partial y(k)} \cdot \frac{\partial y(k)}{\partial S^O(k)} \cdot \frac{\partial S^O(k)}{\partial x(k)} \cdot \frac{\partial x(k)}{\partial S^H(k)} \cdot \frac{\partial S^H(k)}{\partial W^I(k)} \quad (4.41)$$

$$\frac{\partial E(k)}{\partial W^I(k)} = -e(k) \cdot f_N[S^O(k)] \cdot W^O(k) \cdot f_N[S^H(k)] \cdot A_j^I(k-1) \quad (4.42)$$

W^{IR} ağırlığının yenilenmesi için ilgili ağırlığa göre türevinin alınması gerekir. Buna göre;

$$\frac{\partial E(k)}{\partial W^{IR}(k)} = \frac{\partial E(k)}{\partial e(k)} \cdot \frac{\partial e(k)}{\partial y(k)} \cdot \frac{\partial y(k)}{\partial S^O(k)} \cdot \frac{\partial S^O(k)}{\partial x(k)} \cdot \frac{\partial x(k)}{\partial S^H(k)} \cdot \frac{\partial S^H(k)}{\partial W^{IR}(k)} \quad (4.43)$$

$$\frac{\partial E(k)}{\partial W^{IR}(k)} = -e(k) \cdot f_N[S^O(k)] \cdot W^O(k) \cdot f_N[S^H(k)] \cdot A_j^{IR}(k-1) \quad (4.44)$$

Sonuçlar:

$$A_{ij}^O(k) = O_i(k-1) + [K] \cdot A_{ij}^O(k-1) \quad (4.45)$$

$$A_{ij}^O(0) = 0,$$

$$A_{ij}^{IR}(k) = x_i(k-1) + [W_{ij}^{IR} f'_N(S_j^H(k-1)) + K] \cdot A_{ij}^{IR}(k-1) \quad (4.46)$$

$$A_{ij}^{IR}(0) = 0,$$

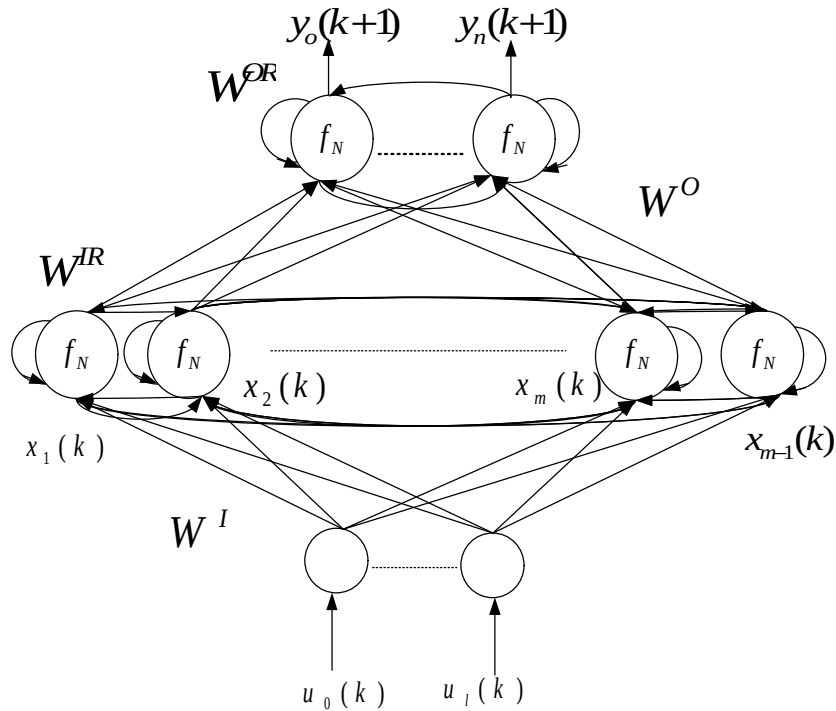
$$A_{ij}^I(k) = u_i(k) + [W_{ij}^{IR} f'_N(S_j^H(k-1)) + K] \cdot A_{ij}^I(k-1)$$

$$A_{ij}^I(0) = 0,$$

4.3.4 Kaotik Yapay Sinir Ağı

Geleneksel kaotik yapay sinir ağları Aihara ve arkadaşları tarafından tasarlanarak, bu sinir ağlarının yapısındaki dinamik karakteristikler ve öğrenme kuralları artırıldı [11]. Değiştirilmiş kaotik sinir ağı global olarak eşleşmiş sinir ağlarıdır. Her bir kaotik nöron ünitesi, kaotik nöron ünitelerinin şimdiki ve geçmiş çıkışlarının global eşleşmesidir [42]. Katmanlar arasındaki ileri yönde girişler için geriye yayılım algoritması, iç katmanlarda tekrarlayan girişler için zamanla ilerleyen öğrenme kuralları kullanmışlardır [11].

Kaotik yapay sinir ağlarında üç geri besleme vardır. Birinci geri besleme, ağırlıkların sabit olduğu ve hücreye gelen girişlerin toplamının geri beslemesinin K katsayısı ile çarpılarak elde edilen değerinin hücreye giriş olarak verildiği geri beslemedir. İkinci geri besleme, orta katmandaki hücre çıkışının kendisine ve orta katmandaki diğer hücrelere giriş olarak verildiği geri beslemedir. Üçüncü geri besleme ise çıkış katmanındaki hücre çıkışının kendisine ve çıkış katmanındaki diğer hücrelere giriş olarak verildiği geri beslemedir. Şekil 4.10'deki kaotik yapay sinir ağlarının yapısı verilmiştir. Kaotik yapay sinir ağına ait matematiksel ifadeler aşağıda verilmiştir.



Şekil 4.10 Kaotik yapay sinir ağının yapısı

Sistemin Parametreleri,

$u_i(k)$ = Her bir farklı k zamanında i. giriş,

$W^I(k)$ = Giriş katmanının ağırlık vektörü,

$W^{IR}(k)$ = Orta katmandaki hücreler arası ağırlık vektörü,

$S_j^H(k)$ = Orta katmandaki girişlerin toplam ağırlığıdır,

$x_j(k)$ = Orta katmandaki j. nöronun çıkışı,

K = Kaotik yapay sinir ağlarının inatçı parametresi,

$f_N(\cdot)$ = Doğrusal olmayan sigmoid fonksiyonu.

Orta katmandaki j. nöronun ağırlıkların toplam denklemi...

$$S_j^H(k) = \sum_{i=1}^l W_j^I u_i(k) + \sum_{q=1}^m W_{qj}^{IR} . x_q(k) + K . S_j^H(k-1) \quad (4.47)$$

Orta katmandaki j. nöronun çıkış denklemi...

$$x_j(k) = f_N[S_j^H(k)] \quad (4.48)$$

Şekil göre;

$y_j(k)$ = Her bir farklı k zamanında i. çıkış,

$W^O(k)$ = Çıkış katmanını ağırlık vektörü,

$W^{OR}(k)$ = Çıkış katmanındaki hücreler arası ağırlık vektörü,

$S_j^O(k)$ = Çıkış katmandaki girişlerin toplam ağırlığıdır,

$x_j(k)$ = Orta katmandaki j. nöronun çıkışı.

Çıkış katmandaki j. nöronun ağırlıkların toplam denklemi...

$$S_p^O(k) = \sum_{j=1}^m W_{ij}^O x_i(k) + \sum_{r=1}^n W_{rj}^{OR} . y_r(k-1) + K . S_p^O(k-1) \quad (4.49)$$

Çıkış katmandaki j. nöronun çıkış denklemi...

$$y(k) = f_N[S_j^O(k)] \quad (4.50)$$

$S_j^O(k)$ Hesaplanması:

$$\begin{aligned} S_p^O(k) &= \sum_{j=1}^m W_{jp}^O f_N \left[\sum_{i=1}^I W_{ij}^I u_i(k) + \sum_{q=1}^m W_{jq}^{IR} x_q(k-1) + K.S_j^H(k-1) \right] + \sum_{r=1}^n W_{rp}^{OR} y_r(k-1) + K.S_p^O(k-1) \\ &= \sum_{j=1}^m W_{jp}^O f_N \left[\sum_{i=1}^I W_{ij}^I u_i(k) + \sum_{q=1}^m W_{jq}^{IR} f_N \left[\sum_{i'=1}^I u_{i'}(k-1) + \sum_{j'=1}^m W_{j'q}^{IR} x_{j'}(k-2) + K.S_{q'}^H(k-2) \right] \right] \\ &\quad + \sum_{r=1}^n W_{rp}^{OR} f_N \left[\sum_{j'=1}^m W_{j'r}^O x_{j'}(k-1) + \sum_{r'=1}^n W_{r'r}^{OR} y_{r'}(k-2) + K.S_{r'}^O(k-2) \right] + K.S_p^O(k-1) \end{aligned} \quad (4.51)$$

$$O_p(k) = NF(u(I), x(I), y(I); I < k) \quad (4.52)$$

$O_p(k) = p$. kaotik nöronun çıkışıdır.

NF = Nonlinear fonksiyon

Denklem 4.52'de nöronun şimdiki ve geçmişteki giriş ve çıkışlarının global eşleşmesi ifade edilmiştir. Bu yüzden bu sistem, herhangi doğrusal olmayan dinamik bir sistemi simule edebilir. Dinamik öğrenme süreci denklem 4.53'de matematiksel olarak ifade edilmiştir.

Dinamik öğrenme yönteminin denklemi

$$W(k+1) = W(k) - a \cdot \nabla_w E(k) \quad (4.53)$$

$W(k)$ = Ağırlık vektörü,

a = Öğrenme oranı,

∇_w = Ağırlıklara uygulanacak düzeltme miktarı,

$E(k)$ = Hata indeksi,

$E(k)$ = Hata indeksidir.

YSA'nın eğitiminde hataların kareleri ile tanımlanan ortalama – karesel hata denklem 4.54 ve 4.55'deki gibi tanımlanır.

$$E(k) = \frac{1}{2} \sum_{i=1}^n [y_i^d(k) - y_i^m(k)]^2 \quad (4.54)$$

$$E(k) = \frac{1}{2} \sum_{i=1}^n e_i^2(k) \quad (4.55)$$

Çıkışlar arasındaki hata,

$$e_i(k) = y_i^d(k) - y_i(k) \quad (4.56)$$

$$\nabla_w E(k) = -e(k) \nabla_w y(k) = -e(k) \nabla_w O(k) \quad (4.57)$$

$O(k)$ yapay sinir ağının çıkış vektörü olduğu zaman, $O(k) = y(k)$ durumda basit tanımlama görevi.

W^I , W^{IR} , W^O , W^{OR} ağırlıklarının yenilenmesi için ilgili ağırlığa göre türevinin alınmasını gerektirir. Buna göre sonuçlar [11]:

$$\frac{\partial O(k)}{\partial W_{ij}^O(k)} = f'_N[S_j^O(k)] A_{ij}^O(k) \quad (4.58)$$

$$\frac{\partial O(k)}{\partial W_{ij}^{OR}(k)} = f'_N[S_j^O(k)] A_{ij}^{OR}(k) \quad (4.59)$$

$$\frac{\partial O(k)}{\partial W_{ij}^{IR}(k)} = \left[\sum_{r=1}^n f'_N(S_r^O(k)) \cdot W_{jr}^O \right] \cdot f'_N[S_j^H(k)] A_{ij}^{IR}(k) \quad (4.60)$$

$$\frac{\partial O(k)}{\partial W_{ij}^I(k)} = \left[\sum_{r=1}^n f'_N(S_r^O(k)) \cdot W_{jr}^O \right] \cdot f'_N[S_j^H(k)] A_{ij}^I(k) \quad (4.61)$$

Sonuçlar:

$$A_{ij}^O(k) = O_i(k-1) + [W_{jj}^{OR} f'_N(S_j^O(k-1)) + K].A_{ij}^O(k-1) \quad (4.62)$$

$$A_{ij}^O(0) = 0,$$

$$A_{ij}^{OR}(k) = O_i(k-1) + [W_{jj}^{OR} f'_N(S_j^O(k-1)) + K].A_{ij}^{OR}(k-1) \quad (4.63)$$

$$A_{ij}^{OR}(0) = 0,$$

$$A_{ij}^{IR}(k) = x_i(k-1) + [W_{ij}^{IR} f'_N(S_j^H(k-1)) + K].A_{ij}^{IR}(k-1) \quad (4.64)$$

$$A_{ij}^{IR}(0) = 0,$$

$$A_{ij}^I(k) = u_i(k) + [W_{ij}^{IR} f'_N(S_j^H(k-1)) + K].A_{ij}^I(k-1) \quad (4.65)$$

$$A_{ij}^I(0) = 0,$$

5. SİSTEM MODELLEME İLE İLGİLİ SONUÇLAR

Bu bölümde, bölüm 4’de anlatılan tek giriş - tek çıkışlı sistem ile çok giriş - çok çıkışlı sistemin YSA ile modellenmesi yapılmıştır. Düz ileri beslemeli, öz geri beslemeli, tam geri beslemeli ve kaotik yapay sinir ağları ile modellenme sonuçları verilmiştir. Kullanılan bu dört yapay sinir ağı ile sistemden alınan geri beslemeden, ağın kendi geri beslemesinin sistem modellenmesindeki etkisi incelenmiştir. Her bir sistem modellemedeki YSA’lar için geriye yayılım algoritması ile örneksel ve toplu öğrenme sonuçları verilmiş ve bu yöntemlerin YSA’nın eğitimi üzerindeki etkisi incelenmiştir. YSA ile sistem modelleme sonuçlarından sonra toplam hatanın değişim grafiği ve hata değeri (E) verilmiştir. Eğitilmiş YSA’nın girişine farklı sinyaller verilerek, test performansı incelenmiştir. Sistem modellenme ile ilgili tüm sonuçlar Matlab m dosyasında program yazılarak elde edilmiştir.

5.1 Tek Giriş - Tek Çıkışlı Sistem Örneğinin YSA ile Modellenme Sonuçları

Bölüm 4’de anlatılan tek bir giriş ve tek bir çıkıştan oluşan bu sistemin, dört farklı YSA ile sistem modellenme sonuçları aşağıda verilmiştir.

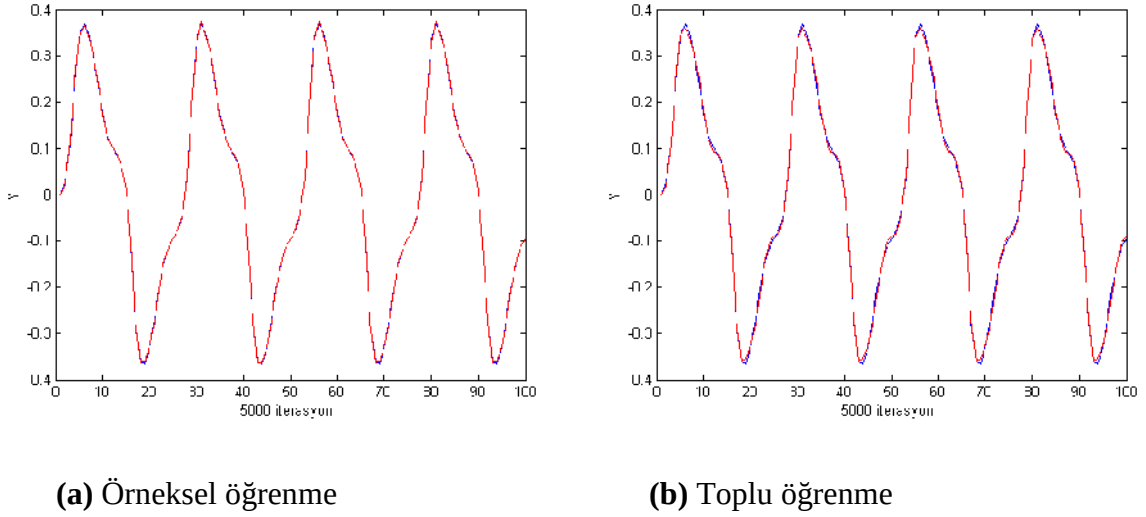
5.1.1 Düz İleri Beslemeli YSA İle Modelleme Sonuçları

Düz ileri beslemeli YSA kendi yapısında geri beslemenin olmadığı, sistemin çıkışından alınan geri beslemenin girişe verilerek, düz ileri beslemeli YSA’nın eğitilmesi ile sistem modellenmesi yapılmaktadır. Bu çalışmada sistemden geri beslemenin alındığı tek modellenme yapısıdır.

Şekil 5.1(a)’da geriye yayılım algoritması kullanılarak örneksel öğrenme yöntemi ile eğitilen düz ileri beslemeli YSA’nın sistem modelleme performansı verilmiştir. Düz ileri beslemeli YSA’yı eğitmek için öğrenme oranı (α)=0.01 seçilmiştir. Şekil 5.1(a)’da görüldüğü gibi düz ileri beslemeli YSA çıkışı ile sistem çıkışı çakışmaktadır bu da eğitimin başarılı olduğunu göstermektedir. Şekil 5.1(b)’de geriye yayılım algoritması kullanılarak toplu öğrenme yöntemi ile eğitilen düz ileri beslemeli YSA’nın sistem modelleme

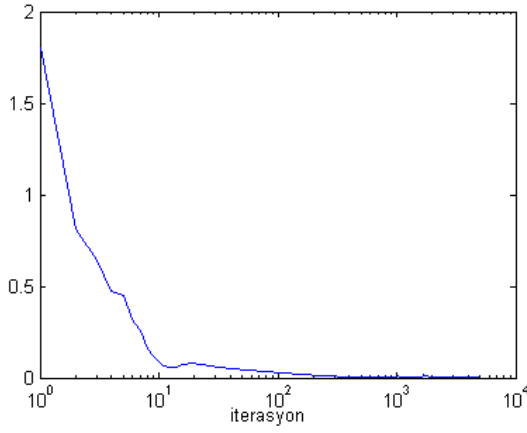
performansı verilmiştir. Düz ileri beslemeli YSA'yı eğitmek için öğrenme oranı (α)=0.1 seçilmiştir. Şekil 5.1(b)'de görüldüğü gibi düz ileri beslemeli YSA çıkışı ile sistem çıkışı çakışmaktadır bu da eğitimin başarılı olduğunu göstermektedir.

Şekil 5.1(a)'daki düz ileri beslemeli YSA 14×5000, şekil 5.1(b)'deki düz ileri beslemeli YSA 48×5000 iterasyon eğitilmiştir. Şekil 5.1'e bakıldığında düz ileri beslemeli YSA'nın eğitiminde kullanılan örneksel öğrenme yönteminin, toplu öğrenme yöntemine göre daha başarılı bir performans verdiği görülmektedir.

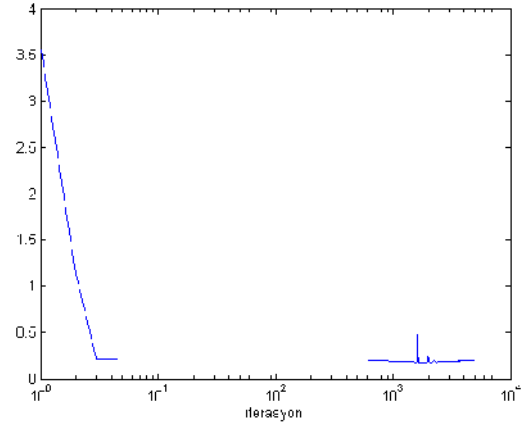


Şekil 5.1 (a) örneksel (b) toplu öğrenme yöntemi ile eğitilen düz ileri beslemeli YSA'nın sistem modelleme performansı

Şekil 5.1'de düz ileri beslemeli YSA ile sistem modellenme performansının, toplam karesel hata değişim grafiği şekil 5.2'de verilmiştir. Şekil 5.2(a)'da örneksel öğrenme yöntemi ile eğitilen düz ileri beslemeli YSA'nın sistem modellenmesinin toplam karesel hata değişim grafiği verilmiştir. 1. iterasyonda $E=1.5296$ değerinde iken 14×5000 iterasyon sonucunda $E=0.000879951$ değerine düşmektedir. Şekil 5.2(b)'de toplu öğrenme yöntemi ile eğitilen düz ileri beslemeli YSA'nın sistem modellenmesinin toplam karesel hata değişim grafiği verilmiştir. 1. iterasyonda $E=3.60276$ değerinde iken 48×5000 iterasyon sonucunda $E=0.00547908$ değerine düşmektedir. Bu hata değişim değerlerine göre toplu öğrenme yöntemi ile eğitilen düz ileri beslemeli YSA'nın sistem modelleme performansı daha yavaş gerçekleşmektedir.



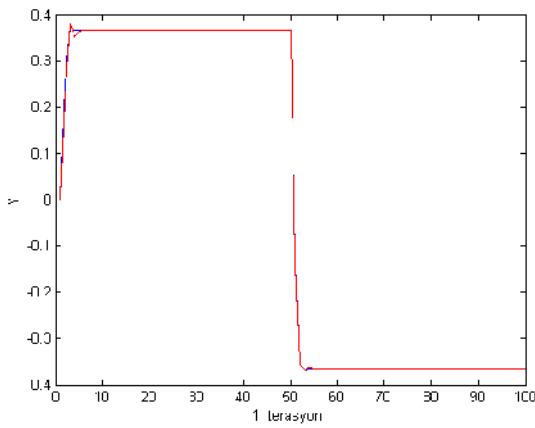
(a) Örnekse öğrenme



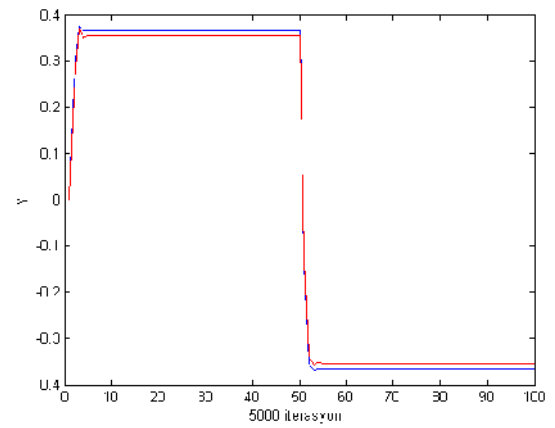
(b) Toplu öğrenme

Şekil 5.2 (a) örnekse (b) toplu öğrenme yöntemi ile eğitilen düz ileri beslemeli YSA'nın sistem modellenmesinin toplam karesel hatanın değişimi

Şekil 5.3'de (a) örnekse (b) toplu öğrenme yöntemi ile eğitilmiş düz ileri beslemeli YSA'nın kare dalga giriş sinyali için test performansı verilmiştir. Şekil 5.3(a)'da, ortalama toplam karesel hata $E=0.0012486$ olarak elde edilmiş ve eğitilmiş düz ileri beslemeli YSA'nın genellemesinin yeterli olduğu görülmüştür. Şekil 5.3(b)'de, ortalama toplam karesel hata $E=0.0161777$ olarak elde edilmiş ve eğitilmiş düz ileri beslemeli YSA'nın genellemesinin yeterli olduğu görülmüştür.



(a) Örnekse öğrenme



(b) Toplu öğrenme

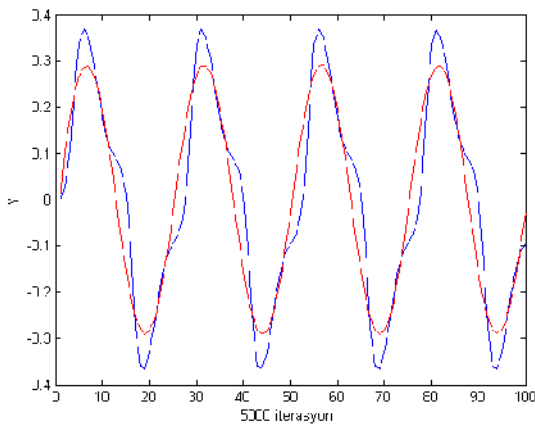
Şekil 5.3 (a) örnekse (b) toplu öğrenme yöntemi ile eğitilmiş düz ileri beslemeli YSA'nın kare dalga giriş sinyali için test performansı

5.1.2 Öz Geri Beslemeli YSA İle Sistem Modelleme Sonuçları

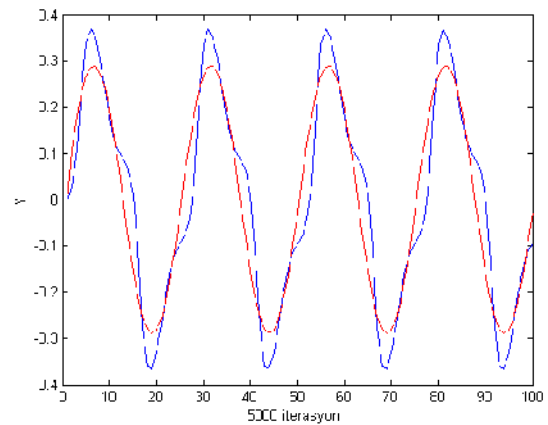
Bölüm 4’de anlatılan öz geri beslemeli YSA’nın yapısında bir geri beslemesi bulunmaktadır. Sistemin çıkışından geri besleme alınmadan, öz geri beslemeli YSA’nın eğitilmesi ile sistem modellenmesi yapılmaktadır.

Şekil 5.4(a)’da geriye yayılım algoritması kullanılarak örnek sel öğrenme yöntemi ile eğitilen öz geri beslemeli YSA’nın sistem modelleme performansı verilmiştir. Öz geri beslemeli YSA’yı eğitmek için öğrenme oranı (a)=0.0001 seçilmiştir. Şekil 5.4(a)’da görüldüğü gibi öz geri beslemeli YSA çıkışı ile sistem çıkışı çakışmamaktadır, bu da eğitimin tam olarak başarılı olmadığını göstermektedir. Şekil 5.4(b)’de geriye yayılım algoritması kullanılarak toplu öğrenme yöntemi ile eğitilen öz geri beslemeli YSA’nın sistem modelleme performansı verilmiştir. Öz geri beslemeli YSA’yı eğitmek için öğrenme oranı (a)=0.001 seçilmiştir. Şekil 5.4(b)’de görüldüğü gibi öz geri beslemeli YSA çıkışı ile sistem çıkışı çakışmamaktadır, bu da eğitimin tam olarak başarılı olmadığını göstermektedir.

Şekil 5.4(a)’daki öz geri beslemeli YSA 1×5000, şekil 5.4(b)’deki öz geri beslemeli YSA 1×5000 iterasyon eğitilmiştir. Şekil 5.4’e bakıldığında öz geri beslemeli YSA’nın eğitiminde kullanılan örnek sel öğrenme ile toplu öğrenme yönteminin arasında performans farkı pek görülmemektedir



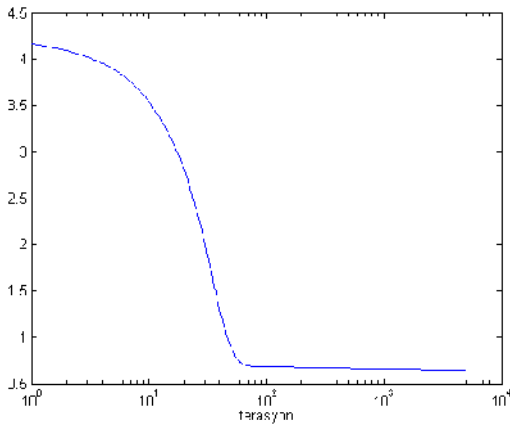
(a) Örnek sel öğrenme



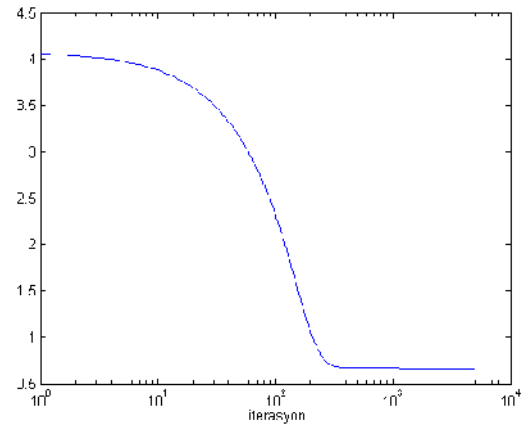
(b) Toplu öğrenme

Şekil 5.4 (a) örnek sel (b) toplu öğrenme yöntemi ile eğitilen öz geri beslemeli YSA’nın sistem modelleme performansı

Şekil 5.4’de verilen öz geri beslemeli YSA ile sistem modellenme performansının, toplam karesel hata değişim grafiği şekil 5.5’de verilmiştir. Şekil 5.5(a)’da örnekssel öğrenme yöntemi ile eğitilen öz geri beslemeli YSA’nın sistem modellenmesinin toplam karesel hata değişim grafiği verilmiştir. 1. iterasyonda $E=4.1722$ değerinde iken 1×5000 iterasyon sonucunda $E=0.649495$ değerine düşmektedir. Şekil 5.5(b)’de toplu öğrenme yöntemi ile eğitilen öz geri beslemeli YSA’nın sistem modellenmesinin toplam karesel hata değişim grafiği verilmiştir. 1. iterasyonda $E=4.0606$ değerinde iken 1×5000 iterasyon sonucunda $E=0.6546$ değerine düşmektedir. Bu hata değişim değerlerine göre örnekssel veya toplu öğrenme yöntemi ile eğitilen öz geri beslemeli YSA’nın sistem modelleme performans farkı fazla değildir.



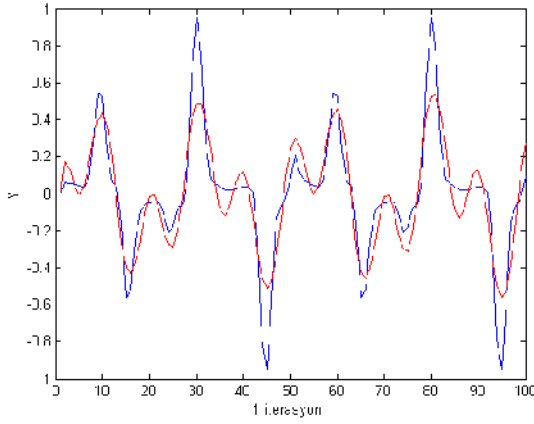
(a) Örnekssel öğrenme



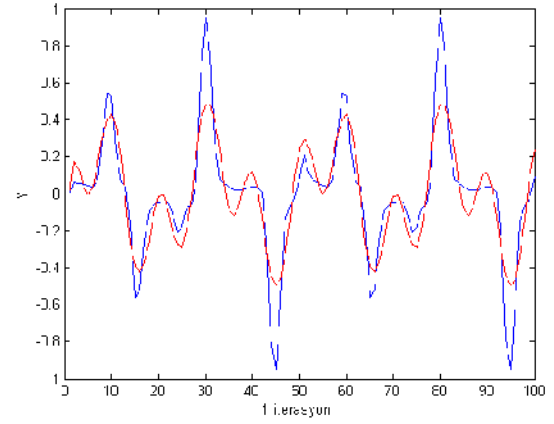
(b) Toplu öğrenme

Şekil 5.5 (a) örnekssel (b) toplu öğrenme yöntemi ile eğitilen öz geri beslemeli YSA’nın sistem modellenmesinin toplam karesel hatanın değişimi

Şekil 5.6’da (a) örnekssel (b) toplu öğrenme yöntemi ile eğitilmiş öz geri beslemeli YSA’nın $u = \sin(2.k.\pi/25) + \cos(3.k.\pi/15)$ giriş sinyali için test performansı verilmiştir. Şekil 5.6(a)’da, ortalama toplam karesel hata $E=2.16577$ olarak elde edilmiş ve eğitilmiş YSA’nın genellemesinin tam olarak yeterli olmadığı görülmüştür. Şekil 5.6(b)’de, ortalama toplam karesel hata $E=2.27228$ olarak elde edilmiş ve eğitilmiş YSA’nın genellemesinin tam olarak yeterli olmadığı görülmüştür.



(a) Örnekse öğrenme



(b) Toplu öğrenme

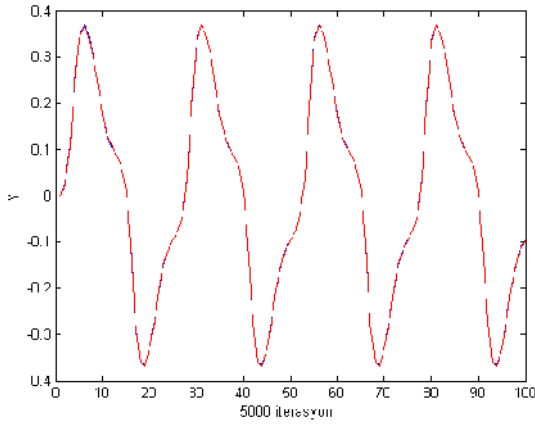
Şekil 5.6 (a) örnekse (b) toplu öğrenme yöntemi ile eğitilmiş öz geri beslemeli YSA'nın $u = \sin(2.k.\pi / 25) + \cos(3.k.\pi / 15)$ giriş sinyali için test performansı

5.1.3. Tam Geri Beslemeli Yapay Sinir Ağı İle Modellenmesi

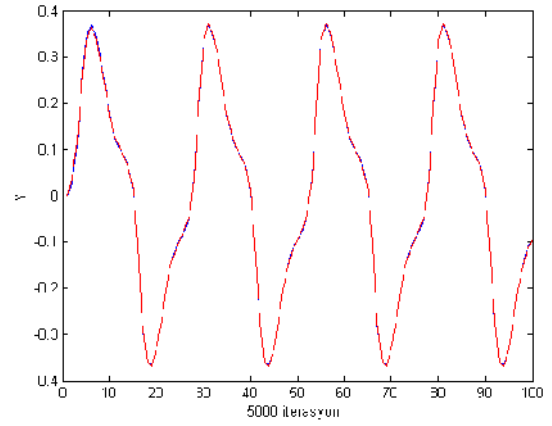
Bölüm 4'de anlatılan tam geri beslemeli YSA'nın yapısında iki geri beslemesi bulunmaktadır. Sistemin çıkışından geri besleme alınmadan ağı eğitilmesi ile sistem modellenmesi yapılmaktadır.

Şekil 5.7(a)'da geriye yayılım algoritması kullanılarak örnekse öğrenme yöntemi ile eğitilen tam geri beslemeli YSA'nın sistem modelleme performansı verilmiştir. YSA'yı eğitmek için öğrenme oranı (a)=0.01 seçilmiştir. Şekil 5.7(a)'da görüldüğü gibi YSA çıkışı ile sistem çıkışı çakışmaktadır buda eğitimin başarılı olduğunu göstermektedir. Şekil 5.7(b)'de geriye yayılım algoritması kullanılarak toplu öğrenme yöntemi ile eğitilen tam geri beslemeli YSA'nın sistem modelleme performansı verilmiştir. YSA'yı eğitmek için öğrenme oranı (a)=0.1 seçilmiştir. Şekil 5.7(b)'de görüldüğü gibi YSA çıkışı ile sistem çıkışı çakışmaktadır buda eğitimin başarılı olduğunu göstermektedir.

Şekil 5.7(a)'daki tam geri beslemeli YSA 3×5000, şekil 5.7(b)'deki tam geri beslemeli YSA 9×5000 iterasyon eğitilmiştir. Şekil 5.7'e bakıldığında tam geri beslemeli YSA'nın eğitiminde kullanılan örnekse öğrenme yönteminin, toplu öğrenme yöntemine göre performans farkı fazla görülmemektedir fakat örnekse öğrenme yöntemiyle tam geri beslemeli YSA daha hızlı eğitilmektedir.



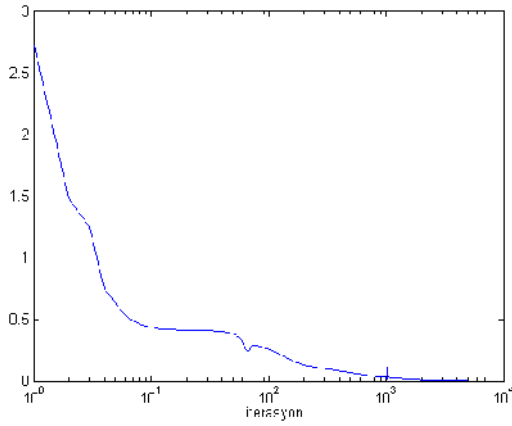
(a) Örnekse öğrenme



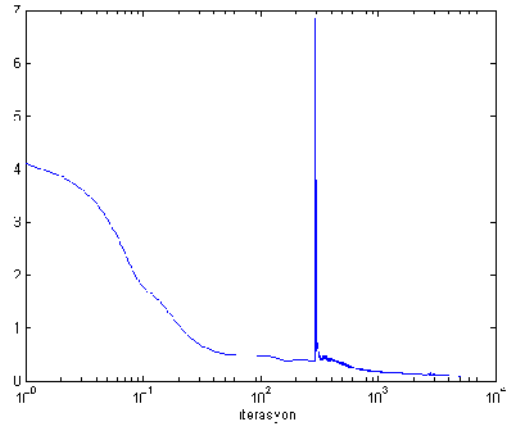
(b) Toplu öğrenme

Şekil 5.7 (a) örnekse (b) toplu öğrenme yöntemi ile eğitilen tam geri beslemeli YSA'nın sistem modelleme performansı

Şekil 5.7'de verilen tam geri beslemeli YSA ile sistem modellenme performansının, toplam karesel hata değişim grafiği şekil 5.8'de verilmiştir. Şekil 5.8(a)'da örnekse öğrenme yöntemi ile eğitilen tam geri beslemeli YSA'nın sistem modellenmesinin toplam karesel hata değişim grafiği verilmiştir. 1. iterasyonda $E=2.7183$ değerinde iken 3×5000 iterasyon sonucunda $E=0.000225636$ değerine düşmektedir. Şekil 5.8(b)'de birinci sistemin toplu öğrenme yöntemi ile eğitilen tam geri beslemeli YSA'nın sistem modellenmesinin toplam karesel hata değişim grafiği verilmiştir. 1. iterasyonda $E=4.0138$ değerinde iken 9×5000 iterasyon sonucunda $E=0.000839931$ değerine düşmektedir. Bu hata değişim değerlerine göre toplu öğrenme yöntemi ile eğitilen tam geri beslemeli YSA daha yavaş eğitilmektedir.



(a) Örnekse öğrenme



(b) Toplu öğrenme

Şekil 5.8 (a) örnekse (b) toplu öğrenme yöntemi ile eğitilen tam geri beslemeli YSA'nın sistem modellenmesinin toplam karesel hatanın değişimi

Şekil 5.5'de öz geri beslemeli ve şekil 5.7'de tam geri beslemeli YSA'nın sistem modelleme sonuçlarına bakılırsa YSA'nın yapısındaki geri besleme arttıkça sistem modellemedeki performansının da arttığını göstermektedir.

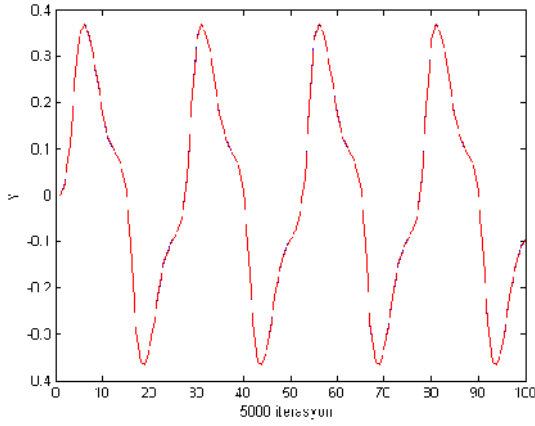
Tam geri beslemeli YSA'nın farklı sinyal girişleri için test performansı yapılmış fakat YSA'nın genellemesinin yeterli olmadığı görülmüştür.

5.1.4 Kaotik Yapay Sinir Ağı İle Modellenmesi

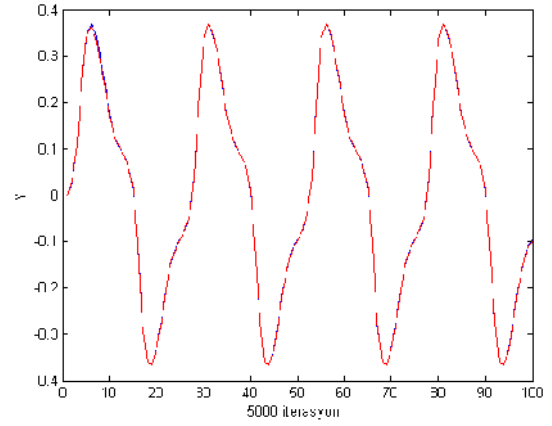
Bölüm 4'de anlatılan kaotik YSA'nın yapısında üç geri beslemesi bulunmaktadır. Sistemin çıkışından geri besleme alınmadan kaotik YSA'nın eğitilmesi ile sistem modellenmesi yapılmaktadır.

Şekil 5.9(a)'da geriye yayılım algoritması kullanılarak örnekse öğrenme yöntemi ile kaotik YSA'nın sistem modelleme performansı verilmiştir. YSA'yı eğitmek için öğrenme oranı (a)=0.003 seçilmiştir. Şekil 5.9(a)'da görüldüğü gibi YSA çıkışı ile sistem çıkışı çakışmaktadır buda eğitimin başarılı olduğunu göstermektedir. Şekil 5.9(b)'de geriye yayılım algoritması kullanılarak toplu öğrenme yöntemi ile eğitilen kaotik YSA'nın sistem modelleme performansı verilmiştir. Kaotik YSA'yı eğitmek için öğrenme oranı (a)=0.03 seçilmiştir. Şekil 5.9(b)'de görüldüğü gibi kaotik YSA çıkışı ile sistem çıkışı çakışmaktadır bu da eğitimin başarılı olduğunu göstermektedir.

Şekil 5.9(a)'daki YSA 9×5000, şekil 5.9(b)'deki YSA 46×5000 iterasyon eğitilmiştir. Şekil 5.9'a bakıldığında kaotik YSA'nın eğitiminde kullanılan örnekssel öğrenme yönteminin, toplu öğrenme yöntemine göre performans farkı fazla görülmemektedir fakat örnekssel öğrenme yöntemiyle kaotik YSA daha hızlı eğitilmektedir.



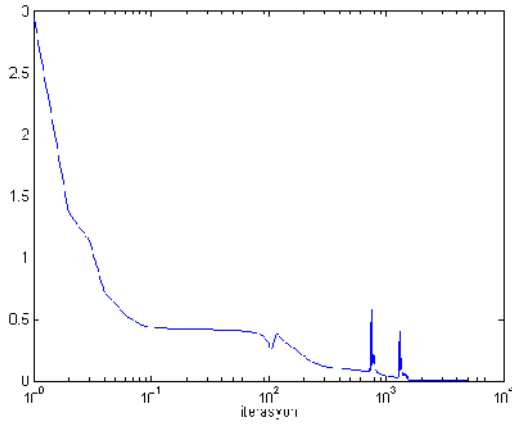
(a) Örnekssel öğrenme



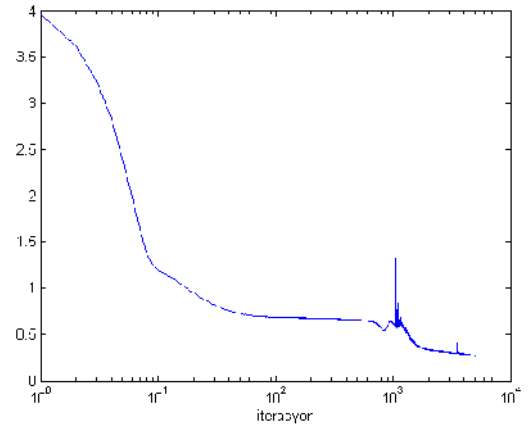
(b) Toplu öğrenme

Şekil 5.9 (a) örnekssel (b) toplu öğrenme yöntemi ile eğitilen kaotik YSA'nın sistem modelleme performansı

Şekil 5.9'da verilen kaotik YSA ile sistem modellenme performansının, toplam karesel hata değişim grafiği şekil 5.10'da verilmiştir. Şekil 5.10(a)'da örnekssel öğrenme yöntemi ile eğitilen kaotik YSA'nın sistem modellenmesinin toplam karesel hata değişim grafiği verilmiştir. 1. iterasyonda $E=2.9282$ değerinde iken 9×5000 iterasyon sonucunda $E=0.000158773$ değerine düşmektedir. Şekil 5.10(b)'de toplu öğrenme yöntemi ile eğitilen tam geri beslemeli YSA'nın sistem modellenmesinin toplam karesel hata değişim grafiği verilmiştir. 1. iterasyonda $E=3.6930$ değerinde iken 46×5000 iterasyon sonucunda $E=0.000618078$ değerine düşmektedir. Bu hata değişim değerlerine göre toplu öğrenme yöntemi ile eğitilen kaotik YSA daha yavaş eğitilmekte olduğunu göstermektedir.



(a) Örneksele öğrenme



(b) Toplu öğrenme

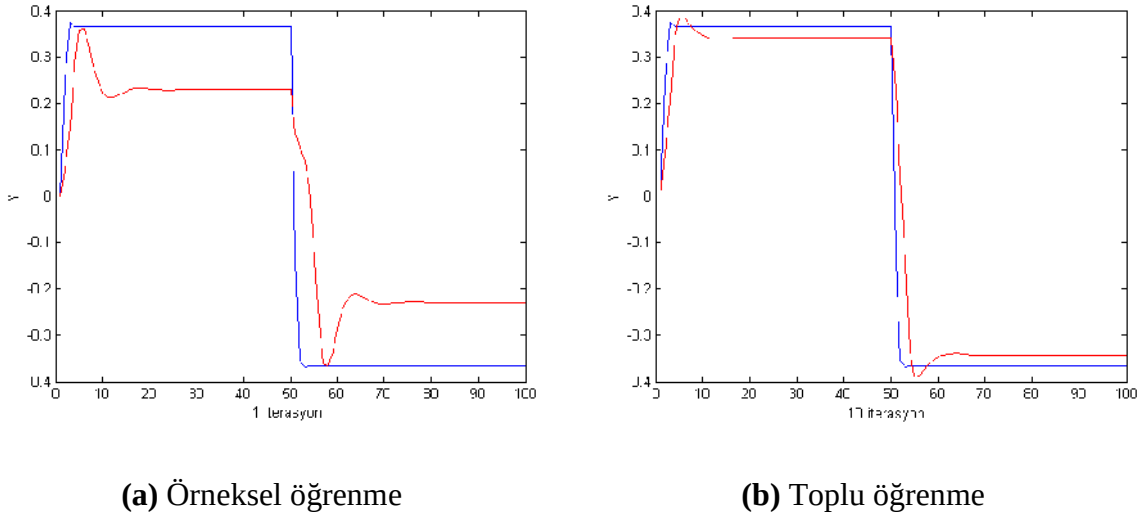
Şekil 5.10 (a) örneksele (b) toplu öğrenme yöntemi ile eğitilen kaotik beslemeli YSA'nın sistem modellenmesinin toplam karesel hatanın değişimi

YSA'nın yapısındaki geri beslemeler artırıldıkça, YSA'nın sistem modellemedeki performansı da artmaktadır. Şekil 5.1 düz ileri beslemeli ve şekil 5.9'da kaotik YSA'nın sistem modelleme sonuçlarına bakılırsa, fazla bir fark görülmemektedir fakat sistem modelleme yapıları tamamen farklıdır. Düz ileri beslemeli YSA'da sistemden geri besleme alınırken kaotik YSA'da sistemden geri besleme alınmaz, ağı yapısında geri beslemeler bulunmaktadır. Şekil 5.4'de öz geri beslemeli ve şekil 5.9'da kaotik YSA'nın sistem modelleme sonuçlarına bakılırsa fazla performans farkı görülmemektedir, bu da öz geri beslemeli ağların yapısında yeterli geri beslemenin olmamasının sonucudur. Şekil 5.7'de tam geri beslemeli ve şekil 5.9'da kaotik YSA'nın sistem modelleme sonuçlarına bakılırsa performans farkı görülmemektedir. Farkın az olmasının sebebi sistemin tek çıkışlı olmasıdır. Çok çıkışlı bir istemde bu fark biraz daha artmaktadır.

Kaotik YSA'nın, farklı sinyal girişleri için test performansı yapılmış fakat YSA'nın genellemesinin yeterli olmadığı görülmüştür. YSA'nın yapısında geri besleme arttıkça, test performansının başarılı olmadığı gözlenmiştir.

Şekil 5.11 (a)'da toplu öğrenme yöntemi ile eğitilmiş kaotik YSA'nın kare dalga giriş sinyali için test performansı verilmiştir. Şekil 5.11(a)'da, ortalama toplam karesel hata $E=2.37769$ olarak elde edilmiş ve eğitilmiş YSA'nın genellemesinin yeterli olmadığı görülmüştür. Kaotik YSA'nın 10 iterasyon sonucu şekil 5.11(b)'de verilmiştir. Şekil

5.11(b)'de ortalama toplam karesel hata $E=0.440593$ olarak elde edilmiştir. Kaotik YSA'nın farklı giriş sinyallerini çabuk öğrendiği gösterilmiştir.



Şekil 5.11 (a) toplu öğrenme 1 iterasyon (b) toplu öğrenme 10 iterasyon için eğitilmiş kaotik YSA'nın kare dalga giriş sinyali için test performansı

5.2 Çok Giriş - Çok Çıkışlı Sistem Örneğinin YSA ile Modellenme Sonuçları

Bölüm 4'de anlatılan çok giriş ve çok çıkıştan oluşan bu sistemin, dört farklı YSA ile sistem modellenme sonuçları aşağıda verilmiştir.

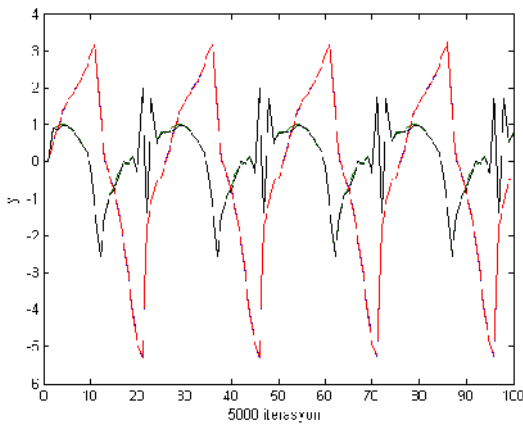
5.2.1 Düz İleri Beslemeli YSA İle Sistem Modelleme Sonuçları

Düz ileri beslemeli YSA ile çok giriş - çok çıkışlı sistemin modellenmesi yapıldığında, iki düz ileri beslemeli YSA'nın girişi, ikide sistemin çıkışından geri besleme alınarak, düz ileri beslemeli YSA'nın 4 girişi olmaktadır. Düz ileri beslemeli YSA'nın eğitilmesi ile sistem modellenmesi yapılmaktadır. Bu çalışmada sistemden geri beslemenin alındığı tek modellenme yapısıdır.

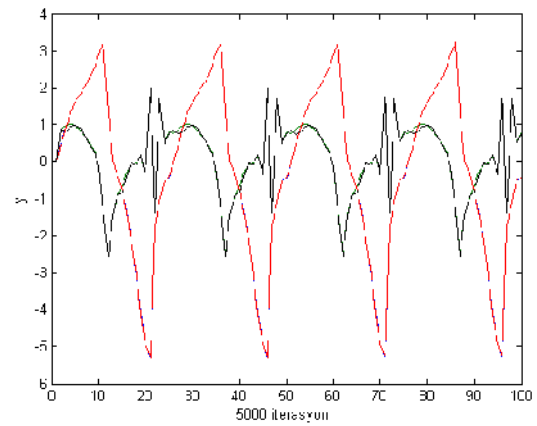
Şekil 5.12(a)'da geriye yayılım algoritması kullanılarak örneksel öğrenme yöntemi ile eğitilen düz ileri beslemeli YSA'nın sistem modelleme performansı verilmiştir. Düz ileri beslemeli YSA'yı eğitmek için öğrenme oranı (a)=0.005 seçilmiştir. Şekil 5.12(a)'da görüldüğü gibi düz ileri beslemeli YSA çıkışı ile sistem çıkışı çakışmaktadır bu da eğitimin başarılı olduğunu göstermektedir. Şekil 5.12(b)'de geriye yayılım algoritması kullanılarak

toplu öğrenme yöntemi ile eğitilen düz ileri beslemeli YSA'nın sistem modelleme performansı verilmiştir. Düz ileri beslemeli YSA'yı eğitmek için öğrenme oranı (a)=0.05 seçilmiştir. Şekil 5.12(b)'de görüldüğü gibi düz ileri beslemeli YSA çıkışı ile sistem çıkışı çakışmaktadır bu da eğitimin başarılı olduğunu göstermektedir.

Şekil 5.12(a)'daki düz ileri beslemeli YSA 6×5000, şekil 5.12(b)'deki düz ileri beslemeli YSA 21×5000 iterasyon eğitilmiştir. Şekil 5.12'ye bakıldığında düz ileri beslemeli YSA'nın eğitiminde kullanılan örneksel öğrenme ile toplu öğrenme yönteminin arasında performans farkı pek görülmemektedir.



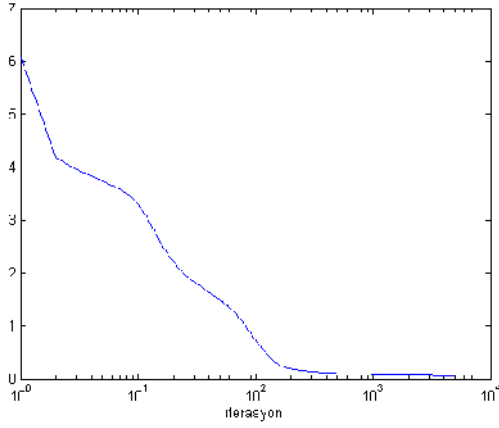
(a) Örneksel öğrenme



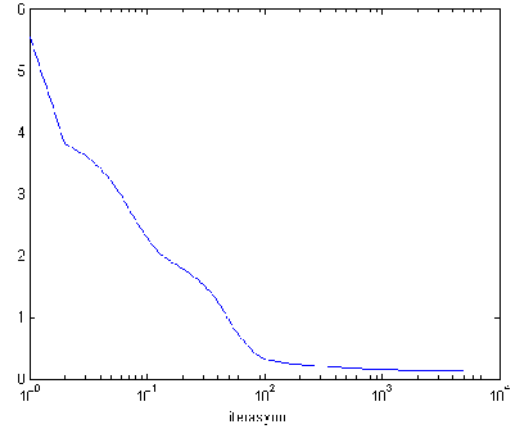
(b) Toplu öğrenme

Şekil 5.12 (a) örneksel (b) toplu öğrenme yöntemi ile eğitilen düz ileri beslemeli YSA 'nın sistem modelleme performansı

Şekil 5.12'de düz ileri beslemeli YSA ile sistem modellenme performansının, toplam karesel hata değişim grafiği şekil 5.13'de verilmiştir. Şekil 5.13(a)'da örneksel öğrenme yöntemi ile eğitilen düz ileri beslemeli YSA'nın sistem modellenmesinin toplam karesel hata değişim grafiği verilmiştir. 1. iterasyonda $E=6.0111$ değerinde iken 6×5000 iterasyon sonucunda $E=0.00701167$ değerine düşmektedir. Şekil 5.13(b)'de toplu öğrenme yöntemi ile eğitilen düz ileri beslemeli YSA'nın sistem modellenmesinin toplam karesel hata değişim grafiği verilmiştir. 1. İterasyonda $E=5.4194$ değerinde iken 21×5000 iterasyon sonucunda $E=0.011086$ değerine düşmektedir. Bu hata değişim değerlerine göre toplu öğrenme yöntemi ile eğitilen düz ileri beslemeli YSA'nın sistem modelleme performansı daha yavaş gerçekleşmektedir.



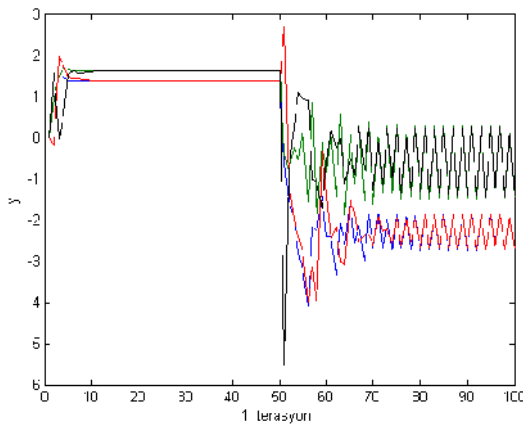
(a) Örneksele öğrenme



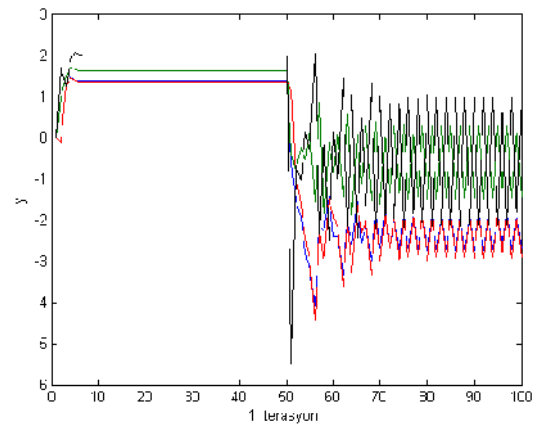
(b) Toplu öğrenme

Şekil 5.13 (a) örneksele (b) toplu öğrenme yöntemi ile eğitilen düz ileri beslemeli YSA'nın sistem modellenmesinin toplam karesel hatanın değişimi

Şekil 5.14'de (a) örneksele (b) toplu öğrenme yöntemi ile eğitilmiş düz ileri beslemeli YSA'nın kare dalga giriş sinyali için test performansı verilmiştir. Şekil 5.14(a)'da, ortalama toplam karesel hata $E=2.08994$ olarak elde edilmiş ve eğitilmiş düz ileri beslemeli YSA'nın genellemesinin yeterli olmadığı görülmüştür. Şekil 5.14(b)'de, ortalama toplam karesel hata $E=8.67842$ olarak elde edilmiş ve eğitilmiş düz ileri beslemeli YSA'nın genellemesinin yeterli olmadığı görülmüştür.



(a) Örneksele öğrenme



(b) Toplu öğrenme

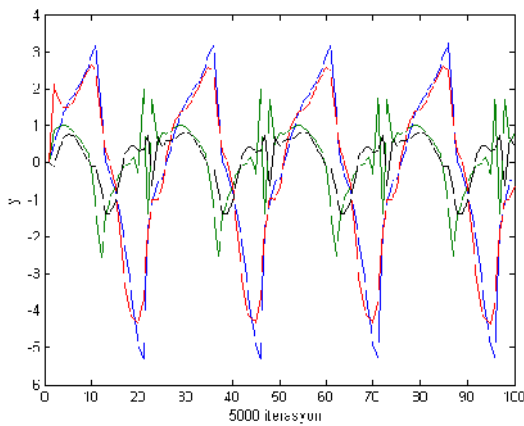
Şekil 5.14 (a) örneksele (b) toplu öğrenme yöntemi ile eğitilmiş düz ileri beslemeli YSA'nın kare dalga giriş sinyali için test performansı

5.2.2 Öz Geri Beslemeli YSA İle Sistem Modelleme Sonuçları

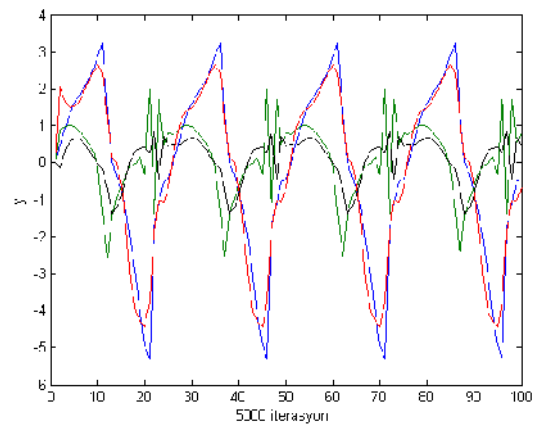
Bölüm 4’de anlatılan öz geri beslemeli YSA’nın yapısında bir geri beslemesi bulunmaktadır. Sistemin çıkışından geri besleme alınmadan, öz geri beslemeli YSA’nın eğitilmesi ile sistem modellenmesi yapılmaktadır.

Şekil 5.15(a)’da geriye yayılım algoritması kullanılarak örnekssel öğrenme yöntemi ile eğitilen öz geri beslemeli YSA’nın sistem modelleme performansı verilmiştir. Öz geri beslemeli YSA’yı eğitmek için öğrenme oranı (a)=0.0001 ve seçilmiştir. Şekil 5.15(a)’da görüldüğü gibi öz geri beslemeli YSA çıkışı ile sistem çıkışı çakışmamaktadır, bu da eğitimin tam olarak başarılı olmadığını göstermektedir. Şekil 5.15(b)’de geriye yayılım algoritması kullanılarak toplu öğrenme yöntemi ile eğitilen öz geri beslemeli YSA’nın sistem modelleme performansı verilmiştir. Öz geri beslemeli YSA’yı eğitmek için öğrenme oranı (a)=0.001 seçilmiştir. Şekil 5.15(b)’de görüldüğü gibi öz geri beslemeli YSA çıkışı ile sistem çıkışı çakışmamaktadır, bu da eğitimin tam olarak başarılı olmadığını göstermektedir.

Şekil 5.15(a)’daki öz geri beslemeli YSA 1×5000, şekil 5.15(b)’deki öz geri beslemeli YSA 1×5000 iterasyon eğitilmiştir. Şekil 5.15’e bakıldığında öz geri beslemeli YSA’nın eğitiminde kullanılan örnekssel öğrenme ile toplu öğrenme yönteminin arasında performans farkı pek görülmemektedir.



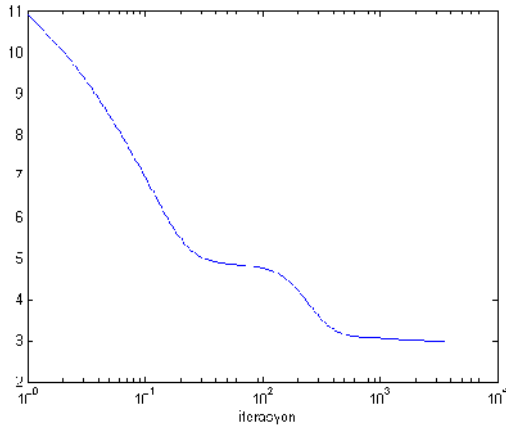
(a) Örnekssel öğrenme



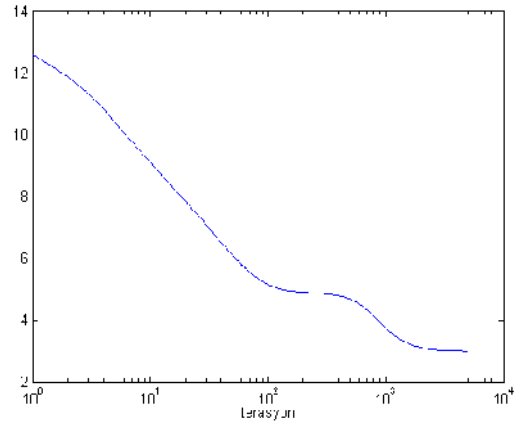
(b) Toplu öğrenme

Şekil 5.15 (a) örnekssel (b) toplu öğrenme yöntemi ile eğitilen öz geri beslemeli YSA’nın sistem modelleme performansı

Şekil 5.15’de verilen öz geri beslemeli YSA ile sistem modellenme performansının, toplam karesel hata değişim grafiği şekil 5.16’da verilmiştir. Şekil 5.16(a)’da örnekssel öğrenme yöntemi ile eğitilen öz geri beslemeli YSA’nın sistem modellenmesinin toplam karesel hata değişim grafiği verilmiştir. 1. iterasyonda $E=10.9071$ değerinde iken 1×5000 iterasyon sonucunda $E=2.95043$ değerine düşmektedir. Şekil 5.16(b)’de toplu öğrenme yöntemi ile eğitilen öz geri beslemeli YSA’nın sistem modellenmesinin toplam karesel hata değişim grafiği verilmiştir. 1. iterasyonda $E=12.5728$ değerinde iken 1×5000 iterasyon sonucunda $E=2.97843$ değerine düşmektedir. Bu hata değişim değerlerine göre örnekssel veya toplu öğrenme yöntemi ile eğitilen öz geri beslemeli YSA’nın sistem modelleme performans farkı fazla değildir.



(a) Örnekssel öğrenme



(b) Toplu öğrenme

Şekil 5.16 (a) örnekssel (b) toplu öğrenme yöntemi ile eğitilen öz geri beslemeli YSA’nın sistem modellenmesinin toplam karesel hatanın değişimi

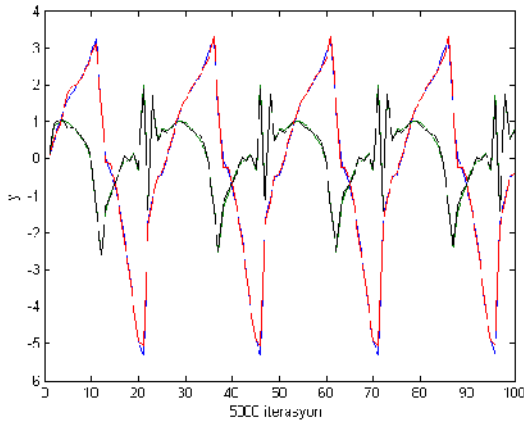
Öz geri beslemeli YSA’nın farklı sinyal girişleri için test performansı yapılmış fakat YSA’nın genellemesinin yeterli olmadığı görülmüştür.

5.2.3. Tam Geri Beslemeli Yapay Sinir Ağı İle Modellenmesi

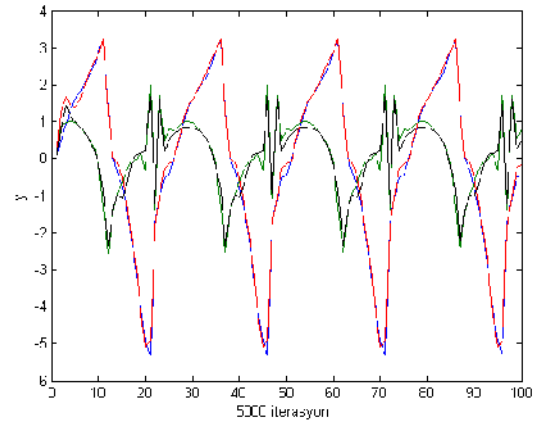
Bölüm 4’de anlatılan tam geri beslemeli YSA’nın yapısında iki geri beslemesi bulunmaktadır. Sistemin çıkışından geri besleme alınmadan tam geri beslemeli YSA’nın eğitilmesi ile sistem modellenmesi yapılmaktadır.

Şekil 5.17(a)'da geriye yayılım algoritması kullanılarak örnekse öğrenme yöntemi ile eğitilen tam geri beslemeli YSA'nın sistem modelleme performansı verilmiştir. YSA'yı eğitmek için öğrenme oranı (a)=0.00005 seçilmiştir. Şekil 5.17(a)'da görüldüğü gibi YSA çıkışı ile sistem çıkışı çakışmaktadır buda eğitimin başarılı olduğunu göstermektedir. Şekil 5.17(b)'de geriye yayılım algoritması kullanılarak toplu öğrenme yöntemi ile eğitilen tam geri beslemeli YSA'nın sistem modelleme performansı verilmiştir. YSA'yı eğitmek için öğrenme oranı (a)=0.001 seçilmiştir. Şekil 5.17(b)'de görüldüğü gibi YSA çıkışı ile sistem çıkışı çakışmaktadır buda eğitimin başarılı olduğunu göstermektedir.

Şekil 5.17(a)'daki tam geri beslemeli YSA 3×5000, şekil 5.17(b)'deki tam geri beslemeli YSA 8×5000 iterasyon eğitilmiştir. Şekil 5.17'e bakıldığında tam geri beslemeli YSA'nın eğitiminde kullanılan örnekse öğrenme yönteminin, toplu öğrenme yöntemine göre performansının daha iyi olduğu görülmektedir.



(a) Örnekse öğrenme

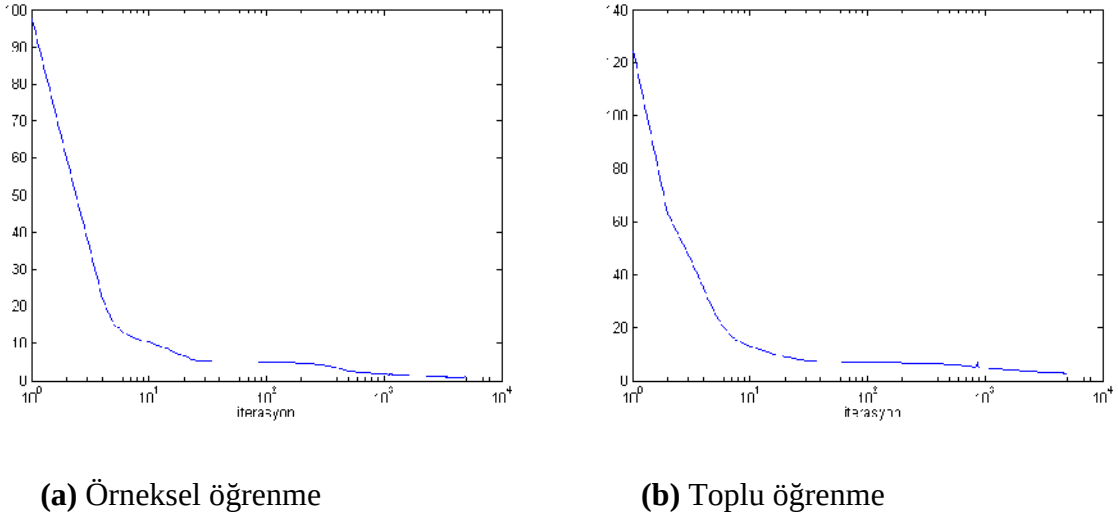


(b) Toplu öğrenme

Şekil 5.17 (a) örnekse (b) toplu öğrenme yöntemi ile eğitilen tam geri beslemeli YSA'nın sistem modelleme performansı

Şekil 5.17'de verilen tam geri beslemeli YSA ile sistem modellenme performansının, toplam karesel hata değişim grafiği şekil 5.18'de verilmiştir. Şekil 5.18(a)'da örnekse öğrenme yöntemi ile eğitilen tam geri beslemeli YSA'nın sistem modellenmesinin toplam karesel hata değişim grafiği verilmiştir. 1. İterasyonda $E=96.5727$ değerinde iken 6×5000 iterasyon sonucunda $E=0.0454523$ değerine düşmektedir. Şekil 5.18(b)'de birinci sistemin toplu öğrenme yöntemi ile eğitilen tam geri beslemeli YSA'nın

sistem modellenmesinin toplam karesel hata değişim grafiği verilmiştir. 1. iterasyonda $E=121.9431$ değerinde iken 8×5000 iterasyon sonucunda $E=0.245294$ değerine düşmektedir. Bu hata değişim değerlerine göre toplu öğrenme yöntemi ile eğitilen tam geri beslemeli YSA daha yavaş ve az eğitilmektedir.



Şekil 5.18 (a) örneksel (b) toplu öğrenme yöntemi ile eğitilen tam geri beslemeli YSA'nın sistem modellenmesinin toplam karesel hatanın değişimi

Tam geri beslemeli YSA'nın farklı sinyal girişleri için test performansı yapılmış fakat YSA'nın genellemesinin yeterli olmadığı görülmüştür.

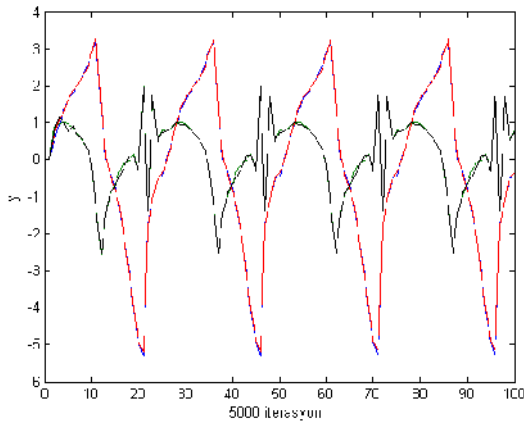
5.2.4 Kaotik Yapay Sinir Ağı İle Modellenmesi

Bölüm 4'de anlatılan kaotik YSA'nın yapısında üç geri beslemesi bulunmaktadır. Sistemin çıkışından geri besleme alınmadan kaotik YSA'nın eğitilmesi ile sistem modellenmesi yapılmaktadır.

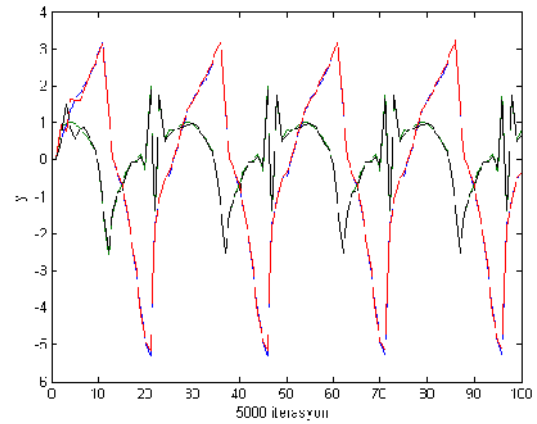
Şekil 5.19(a)'da geriye yayılım algoritması kullanılarak örneksel öğrenme yöntemi ile kaotik YSA'nın sistem modelleme performansı verilmiştir. YSA'yı eğitmek için öğrenme oranı (a)=0.003 seçilmiştir. Şekil 5.19(a)'da görüldüğü gibi YSA çıkışı ile sistem çıkışı çakışmaktadır buda eğitimin başarılı olduğunu göstermektedir. Şekil 5.19(b)'de geriye yayılım algoritması kullanılarak toplu öğrenme yöntemi ile eğitilen kaotik YSA'nın sistem modelleme performansı verilmiştir. Kaotik YSA'yı eğitmek için

öğrenme oranı (α)=0.03 seçilmiştir. Şekil 5.19(b)'de görüldüğü gibi kaotik YSA çıkışı ile sistem çıkışı çakışmaktadır bu da eğitimin başarılı olduğunu göstermektedir.

Şekil 5.19(a)'daki YSA 6×5000, şekil 5.19(b)'deki YSA 12×5000 iterasyon eğitilmiştir. Şekil 5.19'a bakıldığında kaotik YSA'nın eğitiminde kullanılan örneksel öğrenme yönteminin, toplu öğrenme yöntemine göre performans farkı fazla görülmemektedir fakat örneksel öğrenme yöntemiyle kaotik YSA daha hızlı eğitilmektedir.



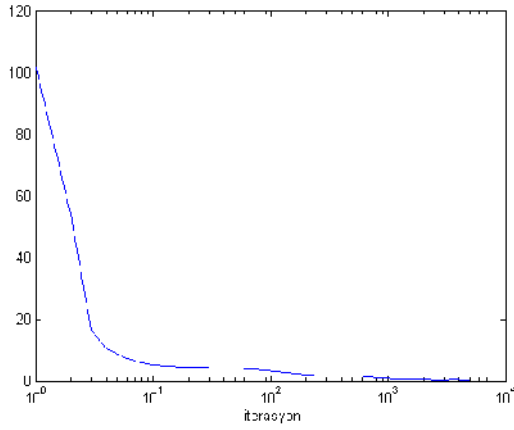
(a) Örnekse öğrenme



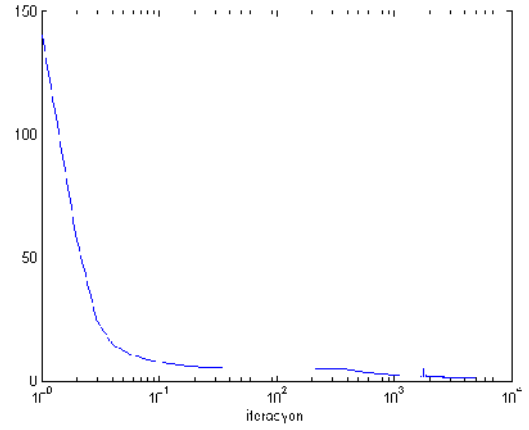
(b) Toplu öğrenme

Şekil 5.19 (a) örnekse (b) toplu öğrenme yöntemi ile eğitilen kaotik YSA'nın sistem modelleme performansı

Şekil 5.19'da verilen kaotik YSA ile sistem modellenme performansının, toplam karesel hata değişim grafiği şekil 5.20'de verilmiştir. Şekil 5.20(a)'da örnekse öğrenme yöntemi ile eğitilen kaotik YSA'nın sistem modellenmesinin toplam karesel hata değişim grafiği verilmiştir. 1. iterasyonda $E=101.4199$ değerinde iken 6×5000 iterasyon sonucunda $E=0.0203142$ değerine düşmektedir. Şekil 5.20(b)'de toplu öğrenme yöntemi ile eğitilen tam geri beslemeli YSA'nın sistem modellenmesinin toplam karesel hata değişim grafiği verilmiştir. 1. iterasyonda $E=138.9010$ değerinde iken 12×5000 iterasyon sonucunda $E=0.0492556$ değerine düşmektedir. Bu hata değişim değerlerine göre toplu öğrenme yöntemi ile eğitilen kaotik YSA daha yavaş eğitilmekte olduğunu göstermektedir.



(a) Örneksele öğrenme



(b) Toplu öğrenme

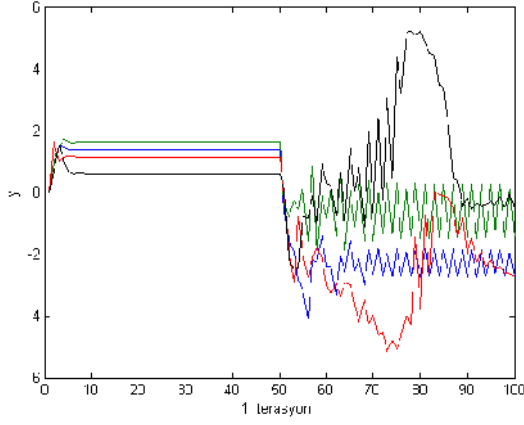
Şekil 5.20 (a) örneksel (b) toplu öğrenme yöntemi ile eğitilen kaotik YSA'nın sistem modellenmesinin toplam karesel hatanın değişimi

İkinci sistem ile YSA'ların modellenme sonuçlarına göre YSA'nın yapısındaki geri beslemeler artırıldıkça, YSA'nın sistem modellemedeki performansı da artmaktadır. Şekil 5.12'de düz ileri beslemeli ve şekil 5.19'da kaotik YSA'nın sistem modelleme sonuçlarına bakılırsa, fazla bir fark görülmemektedir fakat sistem modelleme yapıları tamamen farklıdır. Düz ileri beslemeli YSA'da sistemden geri besleme alınırken kaotik YSA'da sistemden geri besleme alınmaz, ağıın yapısında geri beslemeler bulunmaktadır. Şekil 5.15'de öz geri beslemeli ve şekil 5.19'da kaotik YSA'nın sistem modelleme sonuçlarına bakılırsa fazla performans farkı görülmektedir, bu da öz geri beslemeli ağların yapısında yeterli geri beslemenin olmamasının sonucudur. Şekil 5.17'de tam geri beslemeli ve şekil 5.19'da kaotik YSA'nın sistem modelleme sonuçlarına bakılırsa performans farkı tek giriş - tek çıkışlı sistem modelleme sonuçlarına göre biraz daha artmıştır.

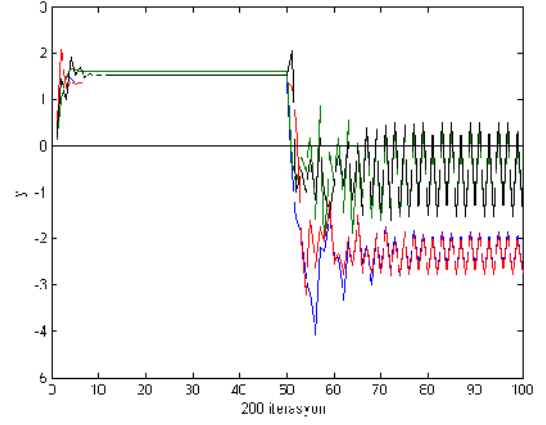
Kaotik YSA'nın, farklı sinyal girişleri için test performansı yapılmış fakat YSA'nın genellemesinin yeterli olmadığı görülmüştür. YSA'nın yapısında geri besleme arttıkça, test performansının başarılı olmadığı gözlenmiştir.

Şekil 5.21 (a)'da toplu öğrenme yöntemi ile eğitilmiş kaotik YSA'nın kare dalga giriş sinyali için test performansı verilmiştir. Şekil 5.21(a)'da, ortalama toplam karesel hata $E=14.056$ olarak elde edilmiş ve eğitilmiş YSA'nın genellemesinin yeterli olmadığı görülmüştür. Kaotik YSA'nın 200 iterasyon sonucu şekil 5.21(b)'de verilmiştir. Şekil

5.21(b)'de ortalama toplam karesel hata $E=0.784472$ olarak elde edilmiştir. Kaotik YSA'nın farklı giriş sinyallerini çabuk öğrendiği bu sistem içinde gösterilmiştir.



(a) Örneksele öğrenme



(b) Toplu öğrenme

Şekil 5.21 (a) toplu öğrenme 1 iterasyon (b) toplu öğrenme 200 iterasyon için eğitilmiş kaotik YSA'nın kare dalga giriş sinyali için test performansı

6. SONUÇLAR

Bu tez çalışmasında, kaotik hücrelerin davranışları (periyodik, kaotik vs) çeşitli yöntemlerle incelenmiştir. Tek girişli - tek çıkışlı ve çok girişli - çok çıkışlı olmak üzere iki farklı doğrusal olmayan dinamik sistemin düz ileri beslemeli, öz geri beslemeli, tam geri beslemeli ve kaotik yapay sinir ağları ile modelleme performansı belirlenmiştir. Kaotik YSA modellerin, ileri beslemeli YSA modellere göre daha az girişe ihtiyaç duymakla birlikte sistem modellemede etkili olduğu belirlenmiştir.

Bölüm 2’de *kaos* kuramı açıklanmış, sürekli zamanlı bir sistemde ve ayrık zamanlı bir sistemde kaotik durum için gerekli şartların neler olduğu incelenmiştir. Bir sürekli zaman dinamik sistem örneği (Doğrusal olmayan sarkaç) kullanılarak sistemin dinamik davranışını (periyodik, kaotik vs) gösteren yöntemler açıklanmıştır.

Bölüm 3’de yapay sinir ağlarının yapısı, çeşitleri ve eğitimi hakkında bilgi verilmiştir. Kaotik hücrenin yapısı incelenmiş ve matematiksel denklemleri çıkartılmıştır. Kaosu ortaya çıkaran yöntemlerle örnek kaotik hücrelerin davranışları incelenmiş ve kaotik hücrede kaos kontrolü yapılmıştır.

Bölüm 4’de yapay sinir ağları ile sistem modelleme hakkında bilgi verilmiştir. Sistem modellemede kullanılacak tek giriş - tek çıkışlı ve çok giriş - çok çıkışlı olmak üzere iki farklı doğrusal olmayan dinamik sistem açıklanmıştır. Sistem modellemede kullanılan düz ileri beslemeli, öz geri beslemeli, tam geri beslemeli ve kaotik YSA’ların yapısı ve eğitimi hakkında bilgi verilmiştir.

Bölüm 5’de tek giriş - tek çıkışlı ve çok giriş - çok çıkışlı sistemlerin yapay sinir ağlarıyla sistem modellenmesi yapılmıştır. Dört farklı YSA’nın sistem modellemedeki performansı gösterilmiştir. Sistemden alınan geri beslemeden, ağın kendi geri beslemesinin sistem modellenmedeki etkisi incelenmiştir. Eğitilmiş YSA’ların girişine farklı sinyaller verilerek test performansları incelenmiştir. Sistem modellemede kaotik yapay sinir ağları, düz ileri beslemeli yapay sinir ağları gibi sistemden geri besleme almamasına rağmen etkili bir performansı vermiştir.

KAYNAKLAR

- [1] **Uyaroğlu, Y., Gündüz, S., Yığın İ.H., ve Keskin H.,** 2009. Kaos Teorisindeki Lorenz Eşitliklerinin Matlab ve Simulink Ortamında Benzetimi ile Karakterize Edilmesi, 5. Uluslararası İleri Teknolojiler Sempozyumu (IATS'09), 13-15 Mayıs 2009, Karabük University, Karabük.
- [2] **Ott, E., Grebogi, C., Yorke, J.A.,** 1990. Controlling chaos, *Physical Review Letters*, **64**, 1196-1199.
- [3] **Aihara, K.,** 2002. Chaos Engineering and its application to distributed processing with chaotic neural network, *Proceedings of the IEEE*, **90(5)**, 919-930.
- [4] **Akın, E., Özer, A.B.,** 2006. Kaos Kuramı, Türk bilişim ansiklopedisi, 536-542.
- [5] **Sneyers, R,** 1997. Climate chaotic instability: Statistical determination and theoretical background, *Environmetrics*, **8(5)**, 517-532.
- [6] **Alligood, K.T., Sauer, T.D. and Yorke, J.A.,** 1997. Chaos : An Introduction to Dynamical Systems, , pp. 537-556, Springer-Verlag, New York.
- [7] **Giannakopoulos, K., Deliyannis, T. and Hadjidemetriou, J.,** 2002. Means for Detecting Chaos and Hyperchaos in Nonlinear Electronic Circuits,. *Proc. 14th Int. Conf. DSP*, Santorini, Greece, pp. 951-954.
- [8] **Gökbulut, M.,** 2001. Kontrol Sistemleri Ders Notları, Fırat Üniversitesi Teknik Eğitim Fakültesi Elektronik-Bilgisayar Eğitimi Bölümü, Elazığ.
- [9] **Degn, H., Holden, A. V. and Olsen, L. F. (Eds.),** 1987. Chaos in biological systems, Plenum Press, New York.
- [10] **Adachi, M. and Aihara, K.,** 1997. Associative dynamics in a chaotic neuralnetwork, *Neural Networks*, **10**, 83–98.
- [11] **Kim, S.H., Hong, S. D., Park, W. W.,** 2001. An adaptive neurocontroller with modified chaotic neural, In: *Neural Networks, Proceedings. IJCNN '01, IEEE*, vol.1, pp. 509-514, Piscataway, NJ, USA.
- [12] **Sompolinsky, H., Crisanti, A., Sommers, H.J.,** 1988. Chaos in random neural Networks, *Phys Rev Lett.*, **61(3)**, 259-262.
- [13] **Riedel, U., Kuhn R. and Van Hemmen J.L.,** 1988. Temporal sequences and chaos in neural nets, *Phys Rev A*, **38 (2)**, 1105–1108.

- [14] **Aihara, K., Takabe, T., Toyoda, M.,** 1990. Chaotic neural network, *Physics Letters A*, **144**, 333-340.
- [15] **Tsuda, I,** 1992. Dynamic link of memory - chaotic memory map in nonequilibrium neural networks, *Neural Networks*, **5**, 313-326.
- [16] **Nara, S., Davis, P., Totsuji, H.,** 1993. Memory search using complex dynamics in a recurrent neural network model, *Neural Networks*, **6**, 963-973.
- [17] **Zhou, C., Chen, T., Huang, W.,** 1997. Chaotic neural network with nonlinear self-feedback and its application in optimization, *Neurocomputing*, **14**, 209-222.
- [18] **Chen, S. S., Shih, C. W.,** 2004. Asymptotic behaviors in a transiently chaotic neural network, *Discrete and Continuous Dynamical Systems (DCDS-A)*, **10(3)**, 805-826.
- [19] **Kim, S. H., Park, W. W.,** 2003. Convergence analysis of chaotic dynamic neuron, *In: Neural Networks, Proceedings of the International Joint Conference*, July 2003, vol. 2, pp. 858-863.
- [20] **URL-1,** www.freepatentsonline.com/5926385.html. 17. Ekim.2007
- [21] **Pecora L.M., Carroll, T. L.,** 1990. Synchronization in chaotic systems, *Physical Review Letters*, **64**, 821-824.
- [22] **Pyragas, K.,** 1992. Continuous control of chaos by self controlling feedback, *Physics Letters A*, **170**, 421-428.
- [23] **Jones, A. J., Tsui, A.P.M., Oliveira, A. G.,** 2002. Neural models of arbitrary chaotic systems: construction and the role of time delayed feedback in control and synchronization, *Complexity International*, **9**, Paper ID: tsui01, URL: <http://www.complexity.org.au/vol09/tsui01/>
- [24] **He, G., Cho, Z., Zhu, P. and Ogura, H.,** 2003. Controlling chaos in a chaotic neural network, *Neural Networks*, **16**, 1195–1200.
- [25] **Baker, G.L. and Gollub, J.P.,** 1990, *Chaotic Dynamics*, Cambridge University Press, Cambridge.
- [26] **Strogatz, S.H.,** 1994. *Nonlinear Dynamics and Chaos*, Perseus Books Publishing, New York.
- [28] **Parker, T.S., Chua L.O.,** 1989. *Practical Numerical Algorithms for Chaotic Systems*, Springer-Verlag, New York.

- [29] **URL-2**, www.phy.davidson.edu/StuHome/chgreene/Chaos/Pendulum/bifurcation_diagram.htm. 20.Kasım.2007
- [30] **Castillo, O., Melin, P.**, 2001. Soft Computing for Control of Nonlinear Dynamical Systems, Physica-verlag.
- [31] **Shen, L., Wang, M.**, 2007. Adaptive control of chaotic systems based on a single layer neural network, *Physics Letters A*, **368**, 379-382.
- [32] **Öztemel, E.**, 2006. Yapay Sinir Ağları, Papatya Yayıncılık.
- [33] **Öztemel, E.**, 1989. Yapay Nöral Ağlar, *Yüksek Lisans Tezi*, İTÜ Fen Bilimleri Enstitüsü, İstanbul.
- [34] **Elmas Ç.** 2007. Yapay Zeka Uygulamaları, Seçkin Yayıncılık.
- [35] **Zurada, M.J.**, 1992. Introduction to Artifical Neural Systems, West Publishing Company Inc., New York.
- [36] **Skarda, C. A, Freeman, W. J.**, 1987. How brain make chaos in order to make sense of the world, *Behavioral and Brain Science*, **10**, 161-195.
- [37] **Caianiello, E.R.**, 1961. Outline of a theory, of thought process and thinking machines, *Journal of Theoretical Biology*, **2**, 204-235.
- [38] **Caianiello, E. R., De Luca, A.**, 1966. Decision equation for binary system: Application to neuronal behavior, *Kybernetik*, **3**, 33-40.
- [39] **Nagumo, J., Sato, S.**, 1972. On a response characteristics of a mathematical neuron model, *Kybemetik*, **10**, 155-164.
- [40] **Aihara, K., Takabe, T. and Toyoda, M.**, 1990. Chaotic neural networks, *Physics Letters A*, **144**, 333-340.
- [41] **Chen, L., Aihara, K.**, 1995. Chaotic simulated annealing by a neural network model with transient chaos, *Neural Networks*, **8(6)**, 915-930.
- [42] **Chen, L., Aihara, K.**, 1999. Global search ability of chaotic neural networks, *IEEE Trans. on Circuit and System I*, 46(8), 914-993.
- [43] **Babuska, R.**, 1998. Fuzzy Modeling for Control, Kluwer Academic Publishers, Boston.
- [44] **Söderström, T., Stoica, P.**, 1989. System Identification, Prentice-Hall International, London.
- [45] **Landau, I.D.**, 1990. System Identification and Control Design. Englewood Cliffs, NJ, Prentice-Hall, USA.

ÖZGEÇMİŞ

Nida KAVAK, 1983 yılında Elazığ'da doğdu ilk ve orta öğretimini Elazığ'da tamamladıktan sonra 2001–2005 yılları arasında Fırat Üniversitesi Teknik Eğitim Fakültesi Elektronik ve Bilgisayar Eğitimi bölümünde lisans öğrenimini tamamlayıp mezun oldu. 2006 yılında Fırat Üniversitesi Fen Bilimleri Enstitüsü Elektronik-Bilgisayar Eğitimi Anabilim Dalında yüksek lisansa başladı.