

T.C.

GEBZE YÜKSEK TEKNOLOJİ ENSTİTÜSÜ
MÜHENDİSLİK VE FEN BİLİMLERİ ENSTİTÜSÜ



ÇOK KATMANLI İLERİ BESLEMELİ
YAPAY SİNİR AĞINDA GERİ YAYILIM
(BACK PROPAGATION) ALGORİTMASININ
KULLANIMIYLA DÖVİZ KURLARININ
TAHMİNİ

98261

Eyüp Selim YILDIRIM
YÜKSEK LİSANS TEZİ
ELEKTRONİK MÜHENDİSLİĞİ
ANABİLİM DALI

TEZ DANIŞMANI
Y.Doç.Dr. Ertuğrul KARAÇUHA

GEBZE

2000

T.C. YÜKSEKÖĞRETİM BAKANLIĞI
DOKÜMANİZASYON MERKEZİ

Eyyup Selim YILDIRIM'in tez çalışması, G.Y.T.E. Mühendislik ve Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 28.02.2000 tarih ve 1000/16 sayılı kararıyla oluşturulan jüri tarafından Elektronik Müh. Anabilim Dalında YÜKSEK LİSANS tezi olarak kabul edilmiştir.

JÜRİ

ÜYE
(Tez Danışmanı)

Elmanur

ÜYE

: Yrd. Doç. Dr. M. Hakan Hocaoglu

[Signature]

ÜYE

[Signature]

ONAY

G.Y.T.E. Mühendislik ve Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 16.05.2000 tarih ve 1000/11 sayılı kararı.



ÖZET



Bu çalışmada, Çok Katmanlı İleri Beslemeli Yapay Sinir Ağına, Eğitim Algoritması olarak Backpropagation (Geri-Yayılım) algoritmasının uygulanmasıyla değişik yabancı paralar için döviz kuru alış fiyatlarının gelecek dönemdeki değerlerinin tahmini yapılmaktadır.

Geçmiş dönem kayıtları kullanılarak yapılan hesaplamalar sonucunda, Geri Yayılım algoritmasının uygulandığı Yapay Sinir Ağı bünyesindeki parametreler ve katman özelliklerinin değişik değerleri için üretilen tahmini değerler, gerçekleşen değerlerle kıyaslanmakta ve bu yolla, uygulanan tekniğin verimliliği ölçülmektedir.



ABSTRACT

In this work, Multi Layer Feedforward Artificial Neural Network Structure trained by the backpropagation algorithm is applied to the forecasting of future values of foreign currency exchange rates.

Through the calculations which are made by using the previous real values of the exchange rates for the subject foreign currencies, real future values are compared with the values extracted with the multilayer feedforward artificial neural network trained by backpropagation algorithm for different network parameters; and the efficiency of the method applied for forecasting the future exchange rates are measured.



TEŞEKKÜR

Gebze Yüksek Teknoloji Enstitüsü'nde yüksek lisans eğitimine başladığım tarihten itibaren yardımlarını esirgemeyen danışman hocam Sayın Yrd. Doç. Dr. Ertuğrul KARAÇUHA Beye...,

Asistanlığım müddetince birlikte çalıştığımız Elektronik Bölümü akademik ve idari personeline...,

Ve, haklarını hiçbir zaman ödeyemeyeceğim, maddi ve manevi desteklerini her zaman yanımda hissettiğim aileme...,

EN İÇTEN TEŞEKKÜRLERİMİ SUNARIM.

İÇİNDEKİLER

VII

ÖZET.....	IV
ABSTRACT.....	V
TEŞEKKÜR.....	VI
İÇİNDEKİLER DİZİNİ.....	VII
SİMGELER VE KISALTMALAR DİZİNİ.....	IX
ŞEKİLLER DİZİNİ.....	X
TABLolar DİZİNİ.....	XII
1. GİRİŞ	1
1.1. Konu ve Önemi	1
1.2. Tezin Amacı ve İçeriği.....	2
1.3. Yapay Sinir Ağları ile İlgili Temel Bilgiler.....	3
1.3.1. Biyolojik YSA Modelleri	4
1.3.2. Uygulamaya Yönelik Modeller	7
1.3.3. Yapay Sinir Ağlarının Sınıflandırılması.....	12
1.4. Kullanılan Notasyon	13
1.5. Çalışmayla İlgili Limitler.....	13
2. PROBLEMİN FORMÜLASYONU VE ÇÖZÜMÜ.....	15
2.1. Döviz Kurlarının Tamini.....	15
2.1.1. Çıkış Değerleri ve Amaç.....	16
2.1.2. Giriş Değerleri.....	16
2.2. YSA Mimarisi ve Backpropagation Algoritması.....	21
2.2.1. Çok Katmanlı İleri Beslemeli Ağ.....	21
2.2.2. Backpropagation (Geri-Yayılm Algoritması).....	24
2.2.2.1. Aktivasyon Fonksiyonu.....	33
2.2.2.2. Öğrenim Oranı.....	35

2.3. Eğitim Verilerinin İşlenmesi.....	36
2.3.1.Eğitim Verilerinin Normalleştirilmesi.....	39
2.3.2. Hedef Değerler.....	41
3. YSA UYGULAMA PROGRAMI.....	42
3.1. Uygulama Programının Kullanımı	42
3.1.1. Eğitim Programı.....	42
3.1.2.Test Programı.....	44
3.1.3.Ek Program ve Fonksiyonlar.....	46
4. OPTIMAL PARAMETRELERİN SEÇİMİ VE PROBLEMİN ÇÖZÜMÜNÜN SAYISAL OLARAK GERÇEKLEŞTİRİLMESİ.....	47
4.1. Parametreler İçin Optimal Değerlerin Seçimi.....	47
4.1.1. Eğitim İçin Uygulanan Çevrim (İterasyon) Sayısı “N”.....	47
4.1.2. Öğrenim Oranına Ait Parametreler.....	51
4.1.3. Ağ Mimarisi.....	55
4.2. Problemin Çözümünün Sayısal Olarak Gerçekleştirilmesi.....	60
4.2.1. ABD Doları Alış Kuru İçin Yapay Sinir Ağıyla Tahmin.....	60
4.2.2. Alman Markı Alış Kuru İçin Yapay Sinir Ağıyla Tahmin.....	62
4.2.3. Japon Yeni Alış Kuru İçin Yapay Sinir Ağıyla Tahmin.....	63
4.2.4. Avusturalya Doları Alış Kuru İçin Yapay Sinir Ağıyla Tahmin.....	67
5. SONUÇLAR.....	69
KAYNAKLAR.....	70
ÖZGEÇMİŞ.....	71

EK'LER

EK 1

SİMGELER VE KISALTMALAR DİZİNİ

Simgeler	Açıklama
BP	Back Propagation
İMKB	İstanbul Menkul Kıymetler Borsası
TCMB	Türkiye Cumhuriyeti Merkez Bankası
YSA	Yapay Sinir Ağı
YZ	Yapay Zeka



ŞEKİLLER DİZİNİ

Şekil	Sayfa
1.1. Biyolojik Nöron	5
1.2. Nöronun Matematiksel Modeli	7
1.3. Eşik Fonksiyonu	10
1.4. Rampa Fonksiyonu	10
1.5. Sigmoid Fonksiyonu	11
1.6. Yapay Sinir Ağlarının Sınıflandırılması	12
2.1. ABD Doları Alış Kuru	18
2.2. Alman Markı Alış Kuru	18
2.3. Japon Yeni Alış Kuru	19
2.4. Avusturalya Doları Alış Kuru	19
2.5. İstanbul Altın Borsası Günlük İşlem Hacmi	20
2.6. İMKB Kapanışa Göre Günlük İşlem Hacmi	20
2.7. TCMB Günlük Net İşlem Hacmi	21
2.8. Çok Katmanlı Yapay Sinir Ağı	22
2.9. YSA Bilgi Akış Yönleri	24
2.10. Döviz Kurları İçin Ham Veri Grupları	38
2.11. Yardımcı Veriler İçin Ham Veri Grupları	38
2.12. Döviz Kurları İçin Normalleştirilmiş Veri Grupları	40
2.13. Yardımcı Veriler İçin Normalleştirilmiş Veri Grupları	40
3.1. Ortalama Karesel Hata Fonksiyonu	44
3.2. Dolar Kuru İçin Test Sonuçları	45

Şekil	Sayfa
4.1. Ortalama Karesel Hata'nın N'e göre Değişimi (Mimari =[7 1])	48
4.2. Ortalama Karesel Hata'nın N'e göre Değişimi (Mimari =[7 10 1])	49
4.3. Ortalama Karesel Hata'nın N'e göre Değişimi (Mimari =[7 10 5 1])	50
4.4. Ortalama Karesel Hata'nın (α, n) 'e göre Değişimi (Mimari =[7 1])	52
4.5. Ortalama Karesel Hata'nın (α, n) 'e göre Değişimi (Mimari =[7 10 1])	53
4.6. Ortalama Karesel Hata'nın (α, n) 'e göre Değişimi (Mimari =[7 10 5 1])	54
4.7. Ortalama Karesel Hata'nın Mimariye Göre Değişimi: 2 ve 3 Katman İçin	56
4.8. Ortalama Karesel Hata'nın Mimariye Göre Değişimi: 3 Katman İçin	57
4.9. Ortalama Karesel Hata'nın Mimariye Göre Değişimi: 3'den Fazla Katman İçin	58
4.10. ABD Doları İçin Ortalama Karesel Hata	61
4.11. ABD Doları İçin Test Sonuçları	62
4.12. Alman Markı için Ortalama Karesel Hata	63
4.13. Alman Markı İçin Test Sonuçları	64
4.14. Japon Yeni İçin Ortalama Karesel Hata	65
4.15. Japon Yeni İçin Test Sonuçları	66
4.16. Avusturalya Doları İçin Ortalama Karesel Hata	67
4.17. Avusturalya Doları İçin Test Sonuçları	68

TABLOLAR DİZİNİ

Tablo	Sayfa
4.1. YSA Parametreleri	47
4.2. Eğitim Sayısına Göre Değişim, (Mimari = [7 1])	48
4.3. Eğitim Sayısına Göre Değişim, (Mimari = [7 10 1])	49
4.4. Eğitim Sayısına Göre Değişim, (Mimari = [7 10 5 1])	50
4.5. (α, n) 'e Göre Değişim (Mimari = [7 1])	52
4.6. (α, n) 'e Göre Değişim (Mimari = [7 10 1])	53
4.7. (α, n) 'e Göre Değişim (Mimari = [7 10 5 1])	54
4.8. Mimariye Göre Değişim, 2 ve 3 Katmanlı Ağlar	56
4.9. Mimariye Göre Değişim, 3 Katmanlı Ağlar	57
4.10. Mimariye Göre Değişim, 3'ten Fazla Katman	58
4.11. Ortalama Değerler	59
4.12. Optimal Parametreler	60

1. GİRİŞ

1.1. Konu ve Önemi

Dünya borsalarında, özellikle Amerikan Wall Street borsasında işlem yapan kurumsal yatırımcılar son dönemlerde Yapay Zeka (YZ), ve YZ tekniklerinin kendi portföylerinde ve ekonominin genel eğilimi ve temel göstergelerinin tahminindeki uygulamaları üzerinde yoğunlaşmaktadırlar. Yatırım planlarında kullanılan YZ tabanlı yazılımlar çoğu kez, ilgili kuruluş tarafından gizli tutulmakta; öyleki bu finans kuruluşları amaçlarına uygun yazılımı kendi bünyelerinde çalıştırdıkları bilgi-işlem uzmanlarıyla hazırlamakta ve dışarıya bilgi sızdırmamaktadırlar. Akademik çevrelere bile ancak konuyla ilgili genel felsefe aktarılmaktadır. Dolayısıyla, YZ tekniklerinin yatırımcı kuruluşlar açısından çok önemli bir rekabet unsuru olarak görüldüğü açıktır [14].

YZ teknikleri, özellikle Yapay Sinir Ağlarının (YSA) kullanımı, finansal uygulamalar dışında sinyal ve/veya verinin var olduğu yada üretildiği, işlendiği her türlü kurum, fiziksel olay yada sistemin (fiziksel, sosyal) bilgi işleme proseslerine, problemin ilgili YZ tekniğine uygun bir biçimde kavramlaştırılmasının derecesine bağlı olarak başarılı bir biçimde uygulanabilmektedir.

YZ'nın başarıyla uygulandığı alanlara örnek olarak; Finansal uygulamalarla birlikte, Kontrol, Haberleşme, Radar Sinyallerinin işlenmesi, örüntü tanıma, genel olarak DSP, non-lineer sistemler, ... vb, verilebilir [1].

Finansal uygulamalarda, genellikle YSA Backpropagation (Geri Yayılım) algoritması uygulanmaktadır.

Algoritmaya temel olan bir tahmin modeli geliştirmek için aşağıdaki aşamalar temel olarak alınmaktadır;

1. Hedefin belirlenmesi, ve bu hedefe yönelik bir yapay sinir ağının aşağıdaki aşamaların gereklerine göre tasarlanması,
2. Tahmin edilecek zaman aralığının belirlenmesi,



3. Problemin tanım kümesiyle ilgili bilginin elde edilmesi,
4. İhtiyaç duyulan bilginin elde edilmesi, ve her bir giriş bilgisinin varılmak istenen hedefle ilişkisinin tanımlanması,
5. Ağın kullanacağı bilgilerin belirlenmesi için, verilerin işlenmesi,
6. Verinin uygun bir biçimde dönüştürülmesi,
7. Verinin, ağın ihtiyaçlarına uygun olarak ölçeklendirilmesi,
8. Giriş verilerindeki çok boyutluluğun mümkün olduğu kadar azaltılması,
9. Ağ mimarisinin tasarlanması (topoloji, katmanların sayısı, katmanların büyüklüğü, parametreler, öğrenme paradigması)
10. Ağ tasarımının geliştirilmesi için "eğitim-test-yeniden tasarım" döngüsünün kullanılması,
11. 10.cu aşamada iken birbiriyle ilişkili giriş verilerinin mümkün olduğu kadar azaltılması,
12. Ağın yeni verilerle beslenmesi, test edilmesi, ve ihtiyaç halinde mimarisi üzerinde değişiklikler yapılması.

Verilen aloritmadan da görüldüğü üzere; YSA ile gerçekleştirilen tahmin işlemlerinde, ağın yapısı veri gruplarını en iyi işleyebilecek biçimde seçilmekte ve seçim işlemi gittikçe daha iyi sonuçlar verecek mimarilerin aşama aşama geliştirilmesine dayanmaktadır.

1.2. Tezin Amacı ve İçeriği:

Çalışmamızda; ileri beslemeli yapay sinir ağlarına eğitim algoritması olarak backpropagation (geriyayılım) algoritması uygulanacak ve eğitim aşaması sonunda elde edilen eğitilmiş YSA'nın, yabancı ülke döviz kurlarının tahmininde kullanımının etkinliği gerçek döviz kuru değerleri ve tahmin edilen döviz kuru değerleri karşılaştırmalarının yapılması suretiyle incelenecektir.

Bu kapsamda konuların tez içerisinde işleniş sırası aşağıdaki gibidir.

1. Bölüm'de, konuyla ilgili giriş bilgileri ve YSA ve YSA'nın temel elemanı olan nöron (sinir hücresi) ile ilgili giriş bilgileri sunulacaktır.

2. Bölüm’de, tahmin sürecinde kullanılacak yapay sinir ağı mimarisi ve uygulanan backpropagation (geriyayılım) algoritmasıyla ilgili formülasyon problemin çözümüne temel teşkil edecek şekilde ele alınacak, tahmin için kullanılan veri grupları ve verilerin ağıya uygulanış biçimi sunulacaktır.

3. Bölüm’de, geliştirilen uygulama programı ve kullanımıyla ilgili ayrıntılar incelenecektir.

4. Bölüm’de, tahmin sürecinde kullanılacak YSA için optimal parametrelerin seçimi uygulama programının kullanımı vasıtasıyla belirlenecek, ve bu optimal parametrelere göre oluşturulmuş YSA’nın, döviz kurlarının tahmininde kullanılması yoluyla elde edilen veriler sunulacaktır.

5. Bölüm’de, uygulanan teknik kapsamında elde edilen sonuçlar sunulacaktır.

1.3. Yapay Sinir Ağları ile İlgili Temel Bilgiler

Yapay Sinir Ağı (YSA); birbiriyle akson bağlantılarıyla haberleşen nöron (sinir hücresi) birimlerinin oluşturduğu gerçek sinir sistemlerinin soyut olarak simüle edilmiş özel bir halidir. YSA, kendi kendini örgütleyebilmesi ve adaptif yapısı nedeniyle, geleneksel paralel işleme tekniklerinden kullanım kolaylığı ve hataları tolere edebilme yönleriyle ayrılmaktadır.

Sinir ağlarının temel modeli ilk olarak 1943 yılında McCulloch ve Pitts tarafından, sinir sistemi aktivitesinin bilgisayar bilimi yaklaşımıyla modellenmesiyle gerçekleştirilmiştir. McCulloch ve Pitts’in modeline göre “nöron” ikili sayısal sisteme sahip bir cihazdır ve her bir nöronun kendi eşik değerine sahip olması nedeniyle; basit eşik mantığına (threshold logic) göre çalışmaktadır. D. O. Hebb, 1947’de yayınlanan “Davranışın Örgütlenmesi” (Organisation of Behavior) isimli çalışmasında; nöronların, kendi kendini örgütleyebilmeleri özellikleri sayesinde uygun biçimde birbirleriyle bağlantılar oluşturduklarını; nöronların ve aralarındaki bağlantıların,

canlının öğrenme sürecine bağlı olarak sürekli bir biçimde değiştiğini yada yenilendiğini ileri sürmüştür [1].

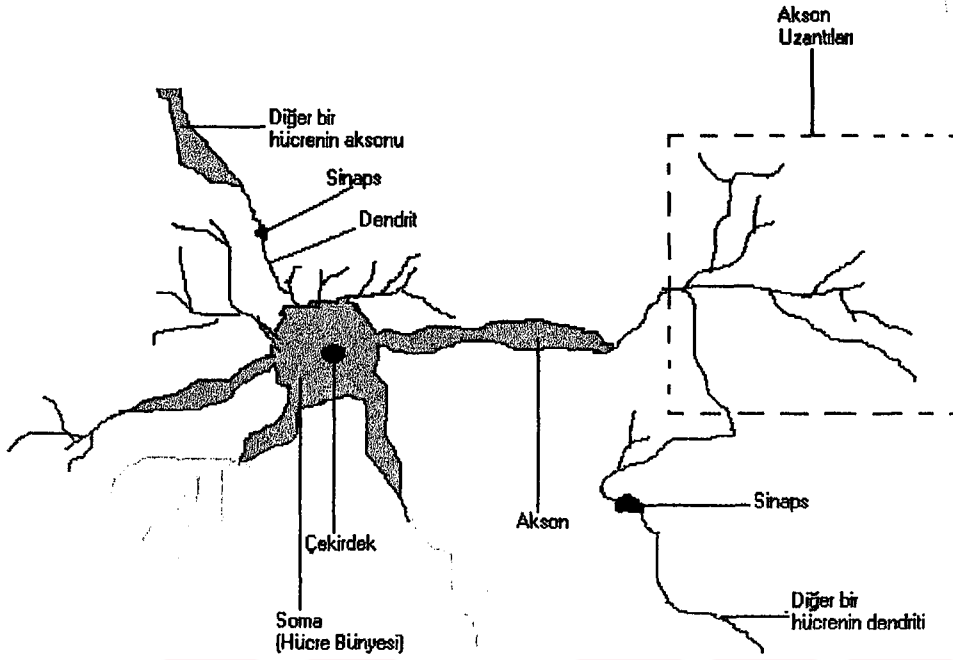
Yapay Sinir Ağı (YSA) modelleri, temel olarak iki ana grup altında toplanabilir:

1. **Biyolojik YSA Modelleri:** Nöroloji bilimiyle aynı çizgiyi takip ederek, YSA'nın canlı varlıkların sinir sistemlerinin değişik bölümlerinin taklit edilerek modellenmesi biçiminde ele alınması,
2. **Uygulamaya Yönelik Modeller:** Yapay Sinir Ağı modellerinin, üzerinde çalışılan uygulamanın tipik ihtiyaçları doğrultusunda oluşturulmasıdır. Bu tür modeller, "Bağlantıcı" (Connectionist) modeller olarak ta bilinirler.

1.3.1. Biyolojik YSA Modelleri:

İnsan beyninin yaklaşık yüz milyar (10^{11}) sinir hücresi (nöron) ve bu hücrelerin birbiriyle bağlantılarını sağlayan 10^{14} sinaps'tan (bağlantı noktası) oluştuğu tahmin edilmektedir. Beyin anatomisi üzerine yapılan çalışmalara göre, ortalama olarak her bir nöronun giriş ve çıkışında 1000'den fazla sinaps vardır. Günümüz bilgisayar sistemleriyle, insan beynini kıyasladığımızda; nöronların anahtarlama sürelerinin (mili saniyeler ölçeğinde) günümüz bilgisayarlarının temel elemanlarının anahtarlama sürelerinden (piko saniyeler ölçeğinde) milyonlarca kez daha uzun olduğunu, buna karşılık nöronlar arası bağlantıların bilgisayarın temel elemanları arasındaki bağlantıdan ortalama 1000 kat daha fazla olduğunu görmekteyiz [4].

Biyolojik temelli YSA modellemelerinin temel amacı, canlı varlıkların sinir sistemleri üzerinde yürütülen çalışmalarda ileri sürülen hipotezlerin geçerliliğini değerlendirebilecek bir model elde edilmesi olarak açıklanabilir. Oluşturulan YSA modelleri, günümüz endüstrisinin bilgi – işlem sorunlarına bir çözüm getirmek amacıyla değil; bilimsel çalışmalarda kullanılabilecek sentetik bir veri üretim modeli olarak kullanılmaktadır [1].



Şekil 1.1 Biyolojik Nöron

Nöronlar ve nöronlar arası bağlantı noktaları olan sinapslar, YSA modellerinin temel unsurlarını oluşturmaktadır. Çoğu nöron, diğer nöronlardan gelen bilgileri sinapslar vasıtasıyla almalarını sağlayan, ağaç dalı biçiminde yapılanmış dendrit adı verilen yapılara sahiptir. Çoğu nöronun sadece yakınındaki birkaç nöronla bu tip bağlantılara sahip olmasına rağmen, bazı nöronlar yakın ve uzak mesafedeki binlerce nöronla bağlantı noktalarına sahip olabilmektedir. Canlı (Biyolojik) sinir hücrelerinin basitleştirilmiş bir çizimi için lütfen Şekil1.1'e bakınız [1].

Şekil1.1'dede görüldüğü gibi, bir sinir hücresi temel olarak üç bölümden oluşmaktadır: Hücre Gövdesi, Dendrit, ve Akson. Dendritler, nöronun diğer nöronlardan gelen girişleri almalarını sağlamakta, akson ise nöronun çıkışını diğer hücrelere taşımaktadır. Sinapslar ise; akson ve dendritler arasındaki bağlantıları sağlayan ara elamanlardır.

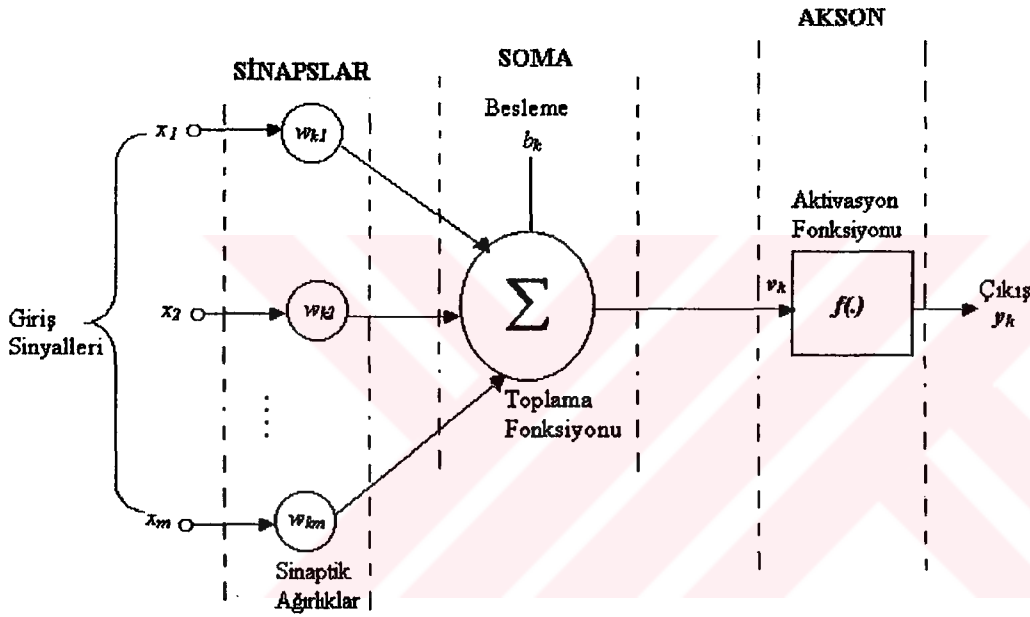
Biyolojik sinir hücreleri tekil işleyiş biçimleri bakımından çok basit olarak şu şekilde açıklanabilir:

- **Dendritler (Giriş Elemanları):** Diğer hücrelerin akson uzantılarının, nöronun giriş birimi olan dendritin uzantılarına bağlantı noktaları olan sinapslar vasıtasıyla bilgi aktarmalarını sağlayan sinir hücresi uzantılarıdır.
- **Soma (Hücre Gövdesi):** Dendritler vasıtasıyla ulaşan voltaj potansiyelleri veya akım değerlerinin arttırıcı ve eksiltici uyarımlarının toplanarak değerlendirildiği ve bu değerlendirme sonucuna göre çıkışı (akson) besleyen hücre gövdesidir.
- **Akson(Çıkış Elemanı):** Somadan gelen beslemeye bağlı olarak diğer sinir hücrelerinin dendrit uzantılarıyla akson uzantıları arasındaki sinapslar aracılığıyla çıkış bilgisi sağlayan hücre uzantısıdır.

Sinir hücresi sinapslar vasıtasıyla gelen uyarımların arttırıcı ve eksiltici özelliklerini bünyesinde toplar. Eğer gelen uyarımların toplamının artı özelliği baskın ise nöron gövdesi (soma) bu toplam mesajı; akson uzantılarının diğer nöronların dendrit uzantılarıyla oluşturduğu sinapslar vasıtasıyla aktarmaktadır. Genelde bu biçimde olduğu gözlemlenen nöron hücresinin işleyiş biçimi bir “eşik fonksiyonu” ile modellenenilmektedir.

1.3.2. Uygulamaya Yönelik Modeller:

Şekil.1.2’de YSA (Yapay Sinir Ağları) tasarımında temel eleman olarak kullanılan nöronun matematiksel bir modellemesi blok diyagram olarak verilmektedir [2].



Şekil 1.2. Nöronun Matematiksel Modeli

Bu temel blok diyagramında; biyolojik sinir hücresinin sinaps, soma, ve akson kısımlarının işleyişe katkıları matematiksel olarak şu şekilde ifade edilmiştir:

1. **Sinapslar:** Bir sinapsa gelen uyarım (giriş sinyali), somaya (hücre gövdesi) aktarılmadan önce biyolojik özelliği niteleyen bir ağırlık katsayısıyla çarpılmakta ve daha sonra somaya toplam etkinin bulunması amacıyla aktarılmaktadır. Şekil 1.2’de; m . sinapsa gelen m . uyarım (giriş sinyali); bu sinapsın, sinapsın ait olduğu dendritin, ve somanın (örnekte k

numaralı somadır) biyolojik özellikleriyle belirlenen w_{km} katsayısıyla çarpılmaktadır.

2. **Soma:** Somanın fonksiyonu lineer bir biçimde bağlantısının bulunduğu sinapslara gelen giriş sinyallerinin ağırlık katsayısıyla (w_{km}) çarpılmış değerlerinin ve somaya aktarılan besleme sinyalinin (şekilde b_k olarak gösterilmiştir) toplamı olarak ifade edilmiştir.
3. **Akson:** Aksonun fonksiyonu, soma çıkışının (v_k) belirli düzeyde sınırlanmasını sağlayan aktivasyon fonksiyonu (f_k) olarak belirtilmiştir.

Cebirsel matematik terimlerle k numaralı bir nöronun operasyonu aşağıdaki şekilde gösterilebilir:

$$u_k = \sum_{j=1}^m w_{kj} x_j \quad (1.1)$$

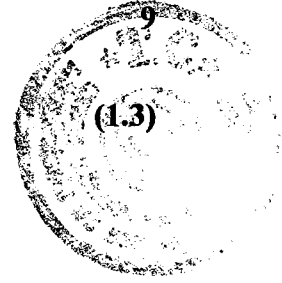
$$y_k = f(u_k + b_k) \quad (1.2)$$

Burada; $x_1, x_2, \dots, x_m$ giriş sinyalleri; $w_{k1}, w_{k2}, \dots, w_{km}$, k numaralı nörona ait sinaptik ağırlık katsayıları; u_k giriş sinyallerinin lineer toplamı; b_k besleme sinyali; $f(\cdot)$ aktivasyon fonksiyonu; ve y_k nöronun çıkış sinyali olarak verilmiştir.

Besleme sinyali b_k , sinaps bağlantıları vasıtasıyla gelen giriş sinyallerinden farklı olarak; hücreyi besleyen harici bir sinyaldir. Besleme sinyalinin değerine bağlı olarak hücre gövdesine biriken toplam potansiyel değişmektedir. Dolayısıyla aktivasyon fonksiyonunun giriş değeri olan u_k ve b_k sinyallerinin toplam değerine göre ürettiği çıkışı harici olarak kontrol eden bir parametre olarak kullanılabilir.

Denklem 1.1 ve 1.2, besleme sinyalini dikkate alarak aşağıdaki biçimde yeniden formüle edebiliriz:

$$v_k = \sum_{j=0}^m w_{kj} x_j$$



$$y_k = f(v_k)$$

(1.4)

Besleme sinyali b_k 'nin;

$$x_0 = +1$$

(1.5)

$$w_{k0} = b_k$$

(1.6)

(1.5) ve (1.6) denklemlerinde görüldüğü biçimde; besleme sinyalinin, giriş değeri +1 sabit değerinde olan, $w_{k0}=b_k$ şinaptik ağırlığına sahip yeni bir sinaps olarak ifade edildiği açıktır.

Şekil 1.2'de verilen aktivasyon fonksiyonu $f(v)$; şinaptik ağırlıklar, giriş sinyalleri ve besleme sinyallerinin toplam değerine bağlı olarak sinir hücresinin çıkış değerini belirlemektedir. Aktivasyon fonksiyonu olarak; Eşik (Şekil.1.3), Rampa (Şekil.1.4), ve Sigmoid fonksiyonları (Şekil.1.5) yaygın olarak kullanılmaktadır[4].

Eşik Fonksiyonu:

$$f(v) = \begin{cases} 1 & v \geq 0 \\ 0 & v < 0 \end{cases} \quad (1.7)$$

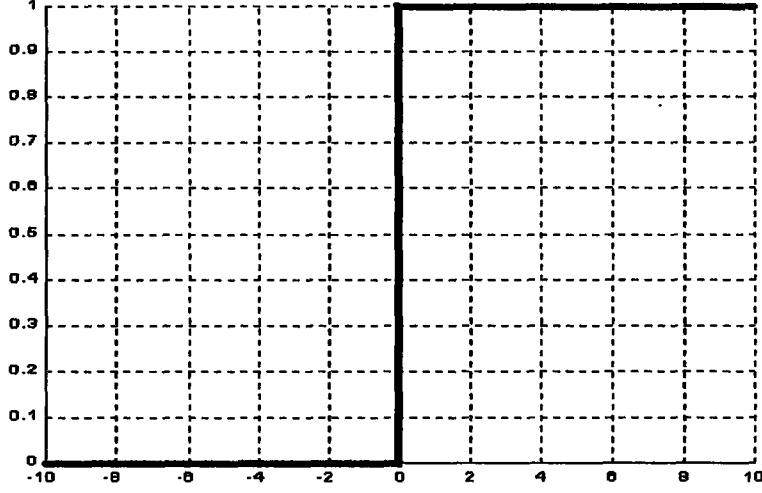
Rampa Fonksiyonu:

$$f(v) = \begin{cases} 1 & ; v \geq +1 \\ v & ; +1 > v > -1 \\ -1 & ; v \leq -1 \end{cases} \quad (1.8)$$

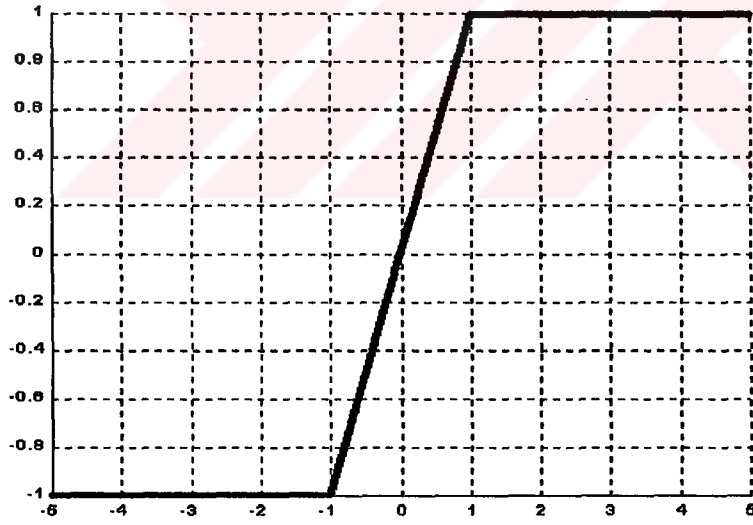
ve Sigmoid Fonksiyonu:

$$f(v) = \frac{1}{1 + \exp(-av)} \quad (1.9)$$

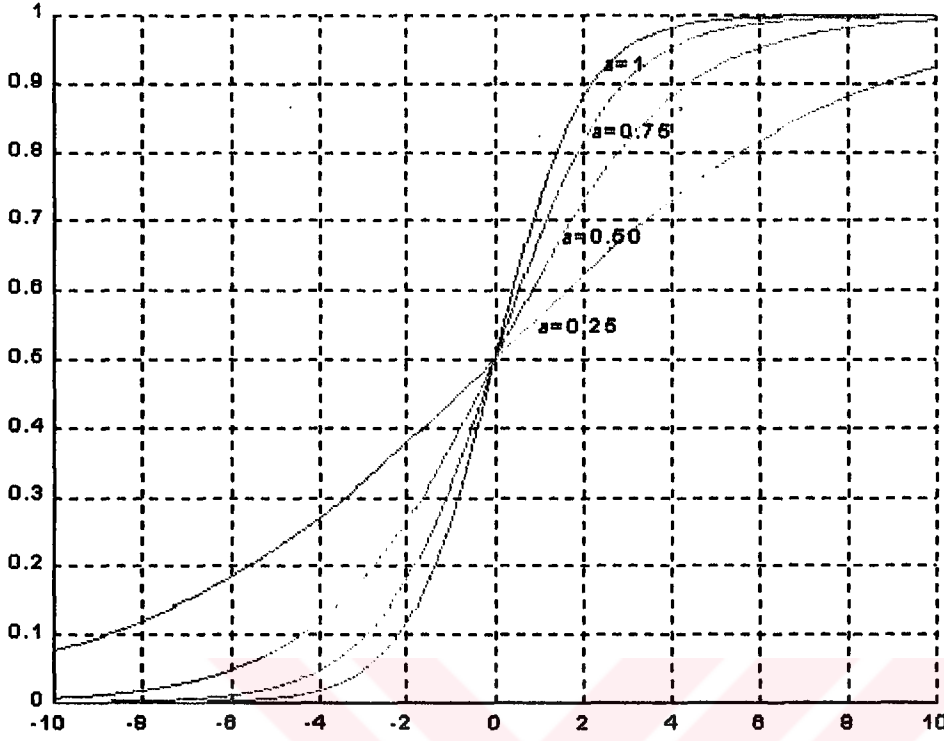
biçiminde ifade edilmektedir.



Şekil 1.3. Eşik Fonksiyonu



Şekil 1.4. Rampa Fonksiyonu



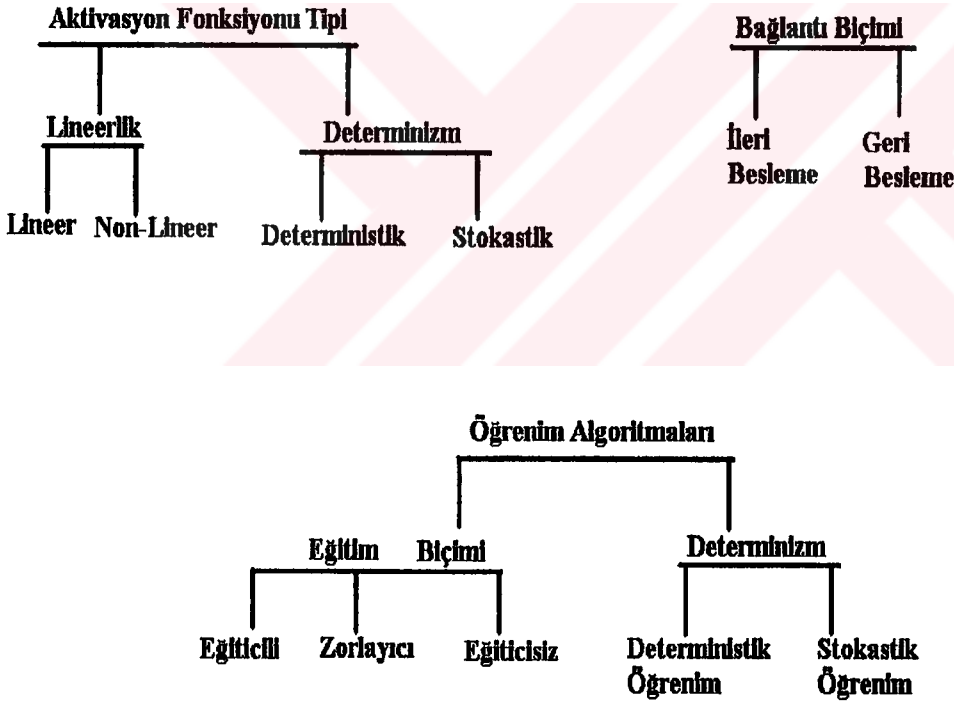
Şekil 1.5. Sigmoid Fonksiyonu

Nöron için verdiğimiz yukardaki matematiksel modeldeki fonksiyonlar ve parametreler, uygulamaya yönelik çalışmalarda doğrudan uygulamanın gerektirdiği amaçlar doğrultusunda seçilmektedir. Uygulamaya yönelik çalışmalarda, nöronun matematiksel modelindeki fonksiyonların matematiksel yaklaşım açısıyla seçilebilmeleri; diğer bir deyişle, matematiksel soyutlamanın getirdiği sınırsız yaklaşım biçimleri; modelin doğal kökeninden farklı olarak, uygulamanın ihtiyaçları yönünde yapılandırılabilmesini sağlamaktadır.

1.3.3. Yapay Sinir Ağlarının Sınıflandırılması:

Yapay Sinir Ağlarının sınıflandırılması, Şekil 1.6'da da belirtildiği gibi; kullanılan aktivasyon fonksiyonu, öğrenim algoritması, yada bağlantı biçimlerine göre değişik açılardan yapılabilir [4].

Kullanılan aktivasyon fonksiyonunun biçimi (Liner yada Non-Linear), Yapay Sinir Ağı boyunca gerçekleştirilen işlemleri sınırlandırması bakımından, YSA davranışını belirleyen en temel unsurdur.



Şekil 1.6. Yapay Sinir Ağlarının Sınıflandırılması

Hücreler arası bağlantı biçimi; YSA'nın genel davranışını belirleyen diğer bir temel unsurdur. Ağ içerisinde "Geri Besleme'nin" bulunması; YSA işleyiş biçimini, hücre çıkışlarını zamanla değişir hale getirmesi açısından dinamik hale getirmektedir. Geri Beslemenin kullanılmadığı mimarilerde (İleri Besleme); ağ boyunca sinyallerin

ilerleyişi, giriş hücrelerinden çıkış hücrelerine doğru; çıkış değerlerinin girişe herhangi bir arttırıcı veya eksiltici etkisi olmadan gerçekleşmektedir.

Geri Beslemeli YSA mimarileri, “Kararlılık”, “Yakınsama”, ve “Depolama Kapasitesi” gerektiren uygulamalarda kullanılmaktadır. İleri Besleme mimarileri ise genelde yaklaşık değer belirleme, tahmin, ve ayırım-sınıflandırma uygulamalarında kullanılmaktadır.

Yapay Sinir Ağlarının sınıflandırılmasında kullanılan bir diğer temel ölçüt de, kullanılan “Öğrenim Algoritmaları”dır. Öğrenim Algoritmaları; aktivasyon fonksiyonları ve ağırlık bağlantı biçiminin sınırlaması içerisinde, ağırlık öğrenmek için kullandığı bilgi örneklerinin tipine bağlıdır.

Öğrenim Algoritmaları temelde; “Eğitiminin Kullanım Biçimi” ve algoritmaların öğrenim kurallarının (Deterministik yada Stokastik) yapısına göre sınıflandırılmaktadır.

1.4. Kullanılan Notasyon

Bu tezde kullanılan notasyon, yaygın olarak kullanılan standart notasyon olduğundan ayrıca belirtilmesine ihtiyaç duyulmamaktadır. Bununla beraber yeni tanımlandığı düşünülen bir terim için gerekli açıklamalar yerinde yapılacaktır.

1.5. Çalışmayla İlgili Limitler

Tez Çalışmasında kullanılan veri grupları; 1999 yılı ilk 140 iş gününe ait gerçek değerlerdir. İlk 100 iş gününe ait değerler YSA’nın eğitimi sürecinde; takip eden 40 iş gününe ait değerler ise, kurulan YSA mimarileri vasıtasıyla elde edilen tahmin değerlerinin test edilmesi amacıyla kullanılmıştır.

YSA vasıtasıyla elde edilen tahmin değerleri, ilgili döviz kurunun bir sonraki iş günündeki değerini vermekte olup tahmin süresi 1 iş günüdür.

Yapay Sinir Ağına ait optimal parametrelerin deneysel olarak belirlendiği 4. Bölüm’de, ilgili parametreler için literatürde yaygın olarak kullanılan değişim aralıkları kullanılmıştır (bakınız: [1], [2], [4],[11], [14]).



2. PROBLEMİN FORMÜLASYONU VE ÇÖZÜMÜ

2.1. Döviz Kurlarının Tahmini

İki ülkenin para birimleri arasındaki değişim oranı yada her iki para biriminden birisinin, diğeri cinsinden fiyatı “Döviz Kuru” olarak adlandırılmaktadır [5].

Döviz kurları; enflasyon, dış ticaret, ve ülkenin genel sosyo-ekonomik politikalarını etkilemesi açısından yerel hükümetlerce, ülkenin uluslararası ekonomik performansını geliştirmek için çeşitli düzenlemelere tabi tutulmaktadır.

Uluslararası parasal sistem, temelde üç biçimde düzenlenmektedir [9]:

- **Altın Standardı:** Ülkelerin para birimlerini sabit miktarda altınla tanımlamasıdır; dolayısıyla altın standardını kullanan diğer ülkelerle sabit kur üzerinden değişim yapılmaktadır.
- **Esnek Döviz Kuru:** Para birimleri arasındaki değişim oranının, herhangi bir düzenleyici unsur olmadan, global ekonominin arz – talep ihtiyaçları doğrultusunda kendiliğinden oluşmasına izin verilmesidir.
- **Karma Döviz Kuru:** Ülkelerin farklı para birimleri için; akışkan döviz kuru, altın standardı, yada değişim oranı üzerinde hükümet kontrolü uygulamasıdır.

Bilindiği üzere, 24 Ocak 1980 kararları sonrasında ülkemizde esnek döviz kuru uygulamasına geçilmiştir. 1999 yılı sonuna kadar uygulanan ekonomik politika, döviz kurlarının, ekonomik arz-talep ilişkisi içerisinde nispeten kendiliğinden oluşması biçiminde olmuştur.

Bu bölümde, Türk Lirası ile farklı para birimleri (Amerikan Doları, Alman Markı, Japon Yeni, ve Avustralya Doları) arasındaki değişim oranının (T.C. Merkez Bankası, Döviz Alış Kurları) tahmininde kullanılan Çok Katmanlı Yapay Sinir Ağı Modeli, ve uygulanacak Backpropagation (Geri Yayılım) Algoritması ile ilgili temel çözümleme ve kullanılan veriler üzerinde gerçekleştirilen işlemler incelenecektir.

2.1.1. Çıkış Değerleri ve Amaç

Amacımız; ABD Doları, Alman Markı, Japon Yeni, ve Avusturalya Doları alış kurlarına ait belirli bir iş günündeki fiyatların; bu iş gününden 1 sonraki iş gününde gerçekleşebilecek değerlerinin tahmin edilmesi için kullanımı; ve gerçek döviz alış kuru değerleriyle tahmin değerleri arasında kıyaslamalar yoluyla, uygulanan YSA mimarisi ve backpropagation algoritmasının etkinliğini incelemektedir.

2.1.2. Giriş Değerleri

Bilindiği üzere, Esnek Döviz Kuru politikasının uygulandığı ülkelerde; Döviz Kurunun değişimi pek çok etkene bağlıdır. Çalışmamızda, istenen çıkış değerlerini tahmin edebilecek Yapay Sinir Ağına ait parametrelerin oluşturulmasını sağlayacak eğitim (BackPropagation) algoritmasına aşağıdaki veri yığınları uygulanacaktır:

- ABD Doları – Döviz Alış (Frekans: İş Günü),
- Alman Markı – Döviz Alış (Frekans: İş Günü),
- Japon Yeni – Döviz Alış (Frekans: İş Günü),
- Avusturalya Doları– Döviz Alış (Frekans: İş Günü),
- İstanbul Altın Borsası İşlem Hacmi (Dolar/Ons) (Frekans: İş Günü),
- İstanbul Menkul Kıymetler Borsası (İMKB) günlük işlem hacmi (Frekans: İş Günü),
- T.C. Merkez Bankası tarafından gerçekleştirilen günlük net repo hacmi (Frekans: İş Günü).

Belirli bir döviz kurunun tahmini için geliştirilecek YSA mimarisindeki ilk aşama, ağa ait parametrelerin belirlenmesi, yani; ağ için en uygun sinaptik ağırlıklar ve besleme değerlerinin belirlenmesi biçiminde gerçekleştirilen YSA'nın eğitilmesi sürecidir.

Yapay Sinir Ağı'nın eğitim sürecinde, belirlenen giriş değerlerine ait örnekler, istenen çıkış değerleriyle birlikte kullanılan eğitim algoritmasına uygulanmaktadır. Ağa ait parametrelerin istenen kesinlikte sağlanmasıyla sonuçlanan

eğitim süreci sonunda, YSA'yı tanımlayan tüm parametreler, uygulanan giriş ve çıkış değerlerine bağlı olarak belirli bir kesinlikte oluşturulmaktadır.

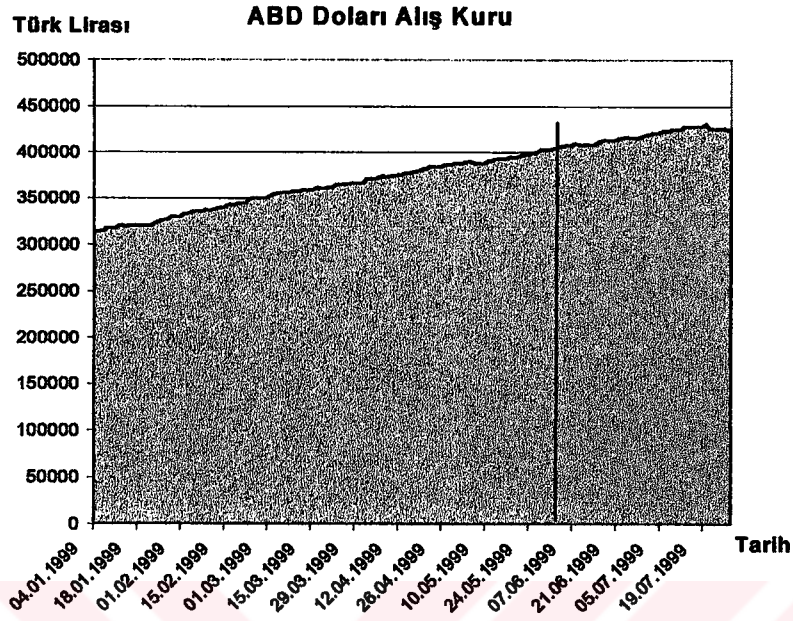
Eğitim aşaması sonrasında; ağın istenilen etkinlik için performansı, eğitim verilerinden farklı verilerin ağa uygulanması ve sonuçların, gerçek değerlerle kıyaslanmasıyla ölçülmektedir.

Ağa, eğitim sürecinde uygulanan veriler "Eğitim Verileri"; ve ağın performansının ölçülmesi için kullanılan verilerde "Test Verileri" olarak adlandırılmaktadır.

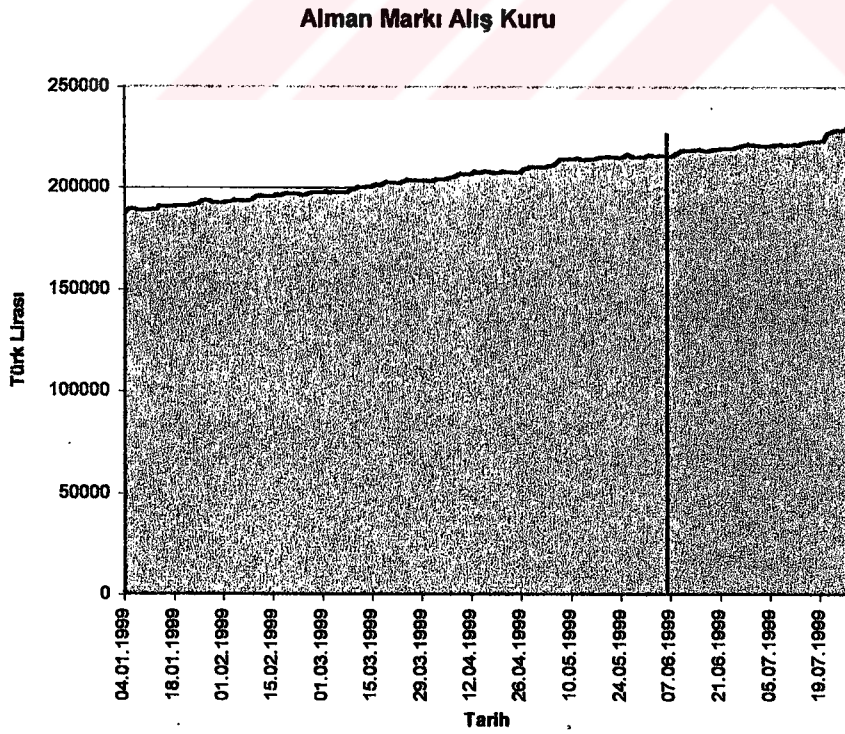
Çalışmamızda, ağa uygulanan veriler için ilk 100 iş gününe ait değerler (04 Ocak – 2 Haziran 1999 tarihleri arasında oluşan, gerçek değerler) eğitim verileri olarak kullanılacak; ağın eğitimi sonrasında, tahmin yeteneğinin ölçülmesi içinde ilk yüz günü takip eden 40 iş gününe ait değerler(03 Haziran – 28 Temmuz 1999 tarihleri arasında oluşan gerçek değerler), test verileri olarak kullanılacaktır.

Amerikan Doları, Alman Markı, Japon Yeni, ve Avusturalya Doları'nun 04 Ocak – 28 Temmuz 1999 tarihleri arasındaki değişimleri için, lütfen Şekil (2.1), Şekil (2.2), Şekil (2.3), ve Şekil (2.4)'e bakınız.

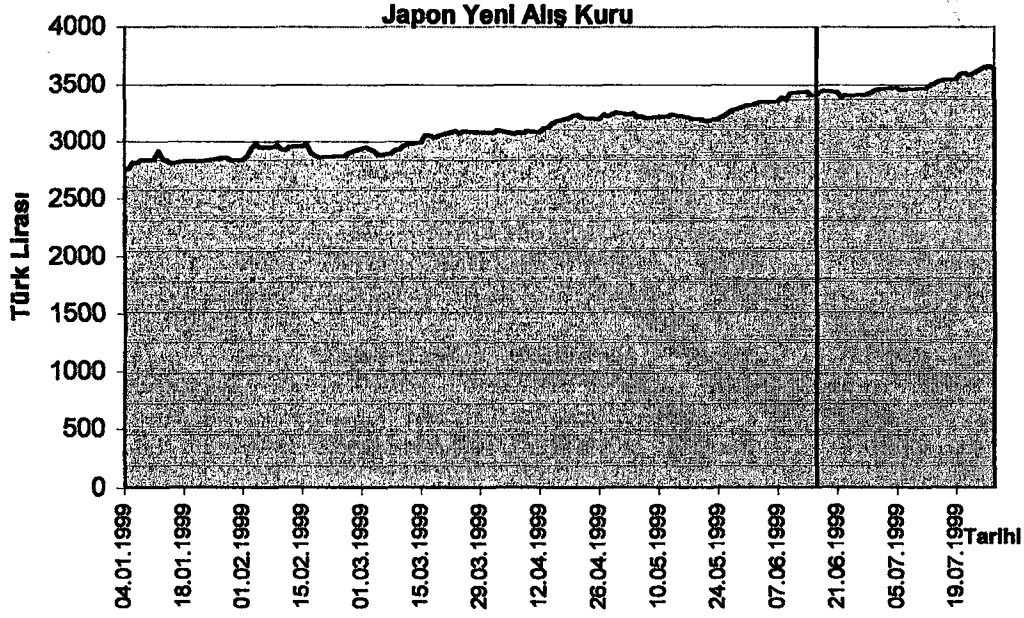
Döviz Kurlarının, diğer ekonomik parametrelere bağımlı olarak değişim gösterdiği açıktır. Çalışmamızda, frekansları bir iş günü olan: İstanbul Altın Borsası İşlem Hacmi (Dolar/Ons), İstanbul Menkul Kıymetler Borsası (İMKB) günlük işlem hacmi, ve T.C. Merkez Bankası tarafından gerçekleştirilen günlük net repo hacmi değerleri; döviz kurlarının değişimi üzerindeki etkileri nedeniyle; oluşturulacak YSA algoritması için giriş değerleri olarak kullanılacaktır. Söz konusu belirleyici parametrelerin 04 Ocak – 28 Temmuz 1999 tarihleri arasındaki değişimleri için, lütfen lütfen Şekil (2.5), Şekil (2.6), ve Şekil (2.7)'ye bakınız.



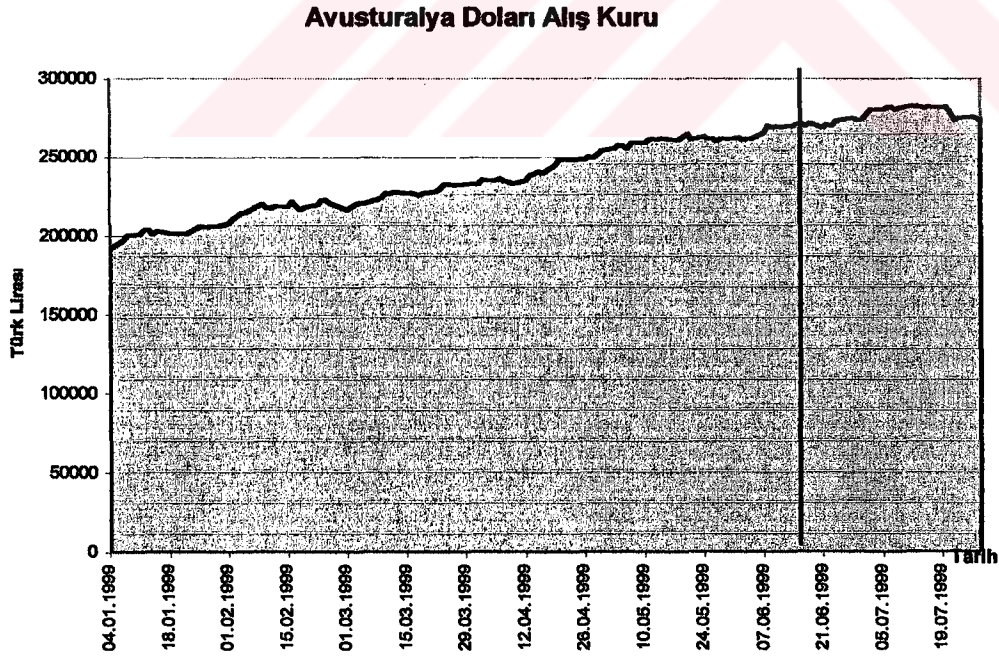
Şekil 2.1. ABD Doları Alış Kuru



Şekil 2.2. Alman Markı Alış Kuru

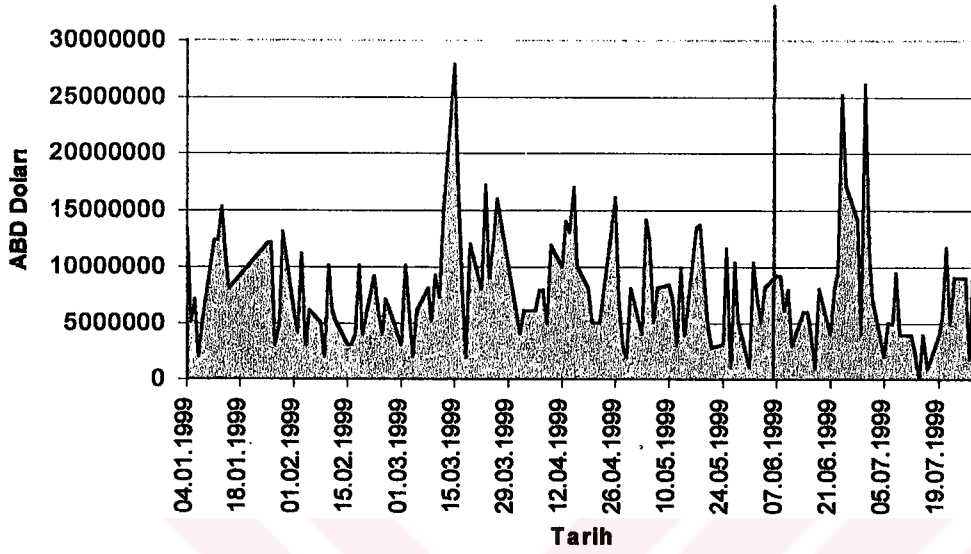


Şekil 2.3. Japon Yeni Alış Kuru



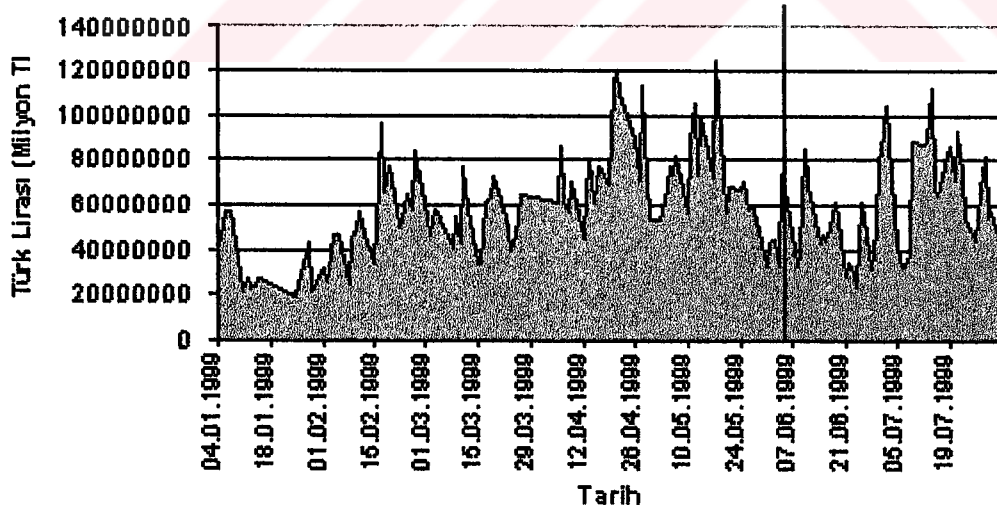
Şekil 2.4. Avusturalya Doları Alış Kuru

İstanbul Altın Borsası, Günlük İşlem Hacmi (\$/Ons)



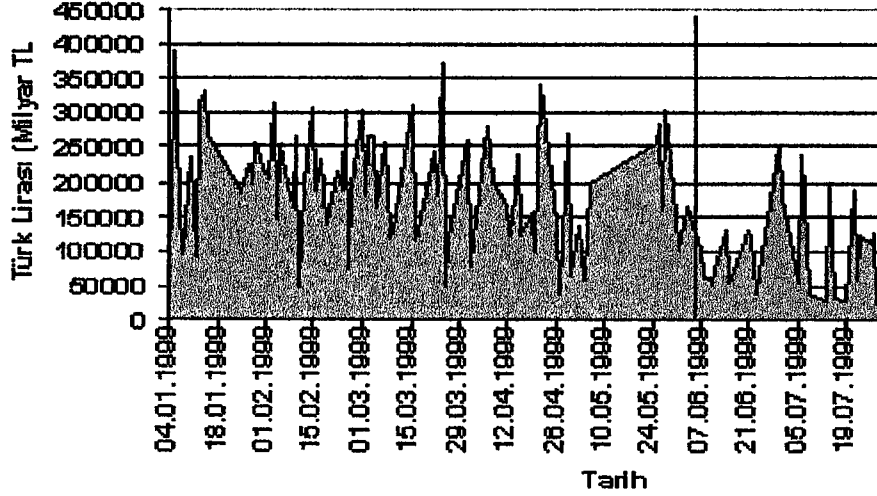
Şekil 2.5. İstanbul Altın Borsası Günlük İşlem Hacmi

İMKB Kapanışa Göre Günlük İşlem Hacmi



Şekil 2.6. İMKB Kapanışa Göre Günlük İşlem Hacmi

T.C. Merkez Bankası Günlük Net Repo İşlem Hacmi



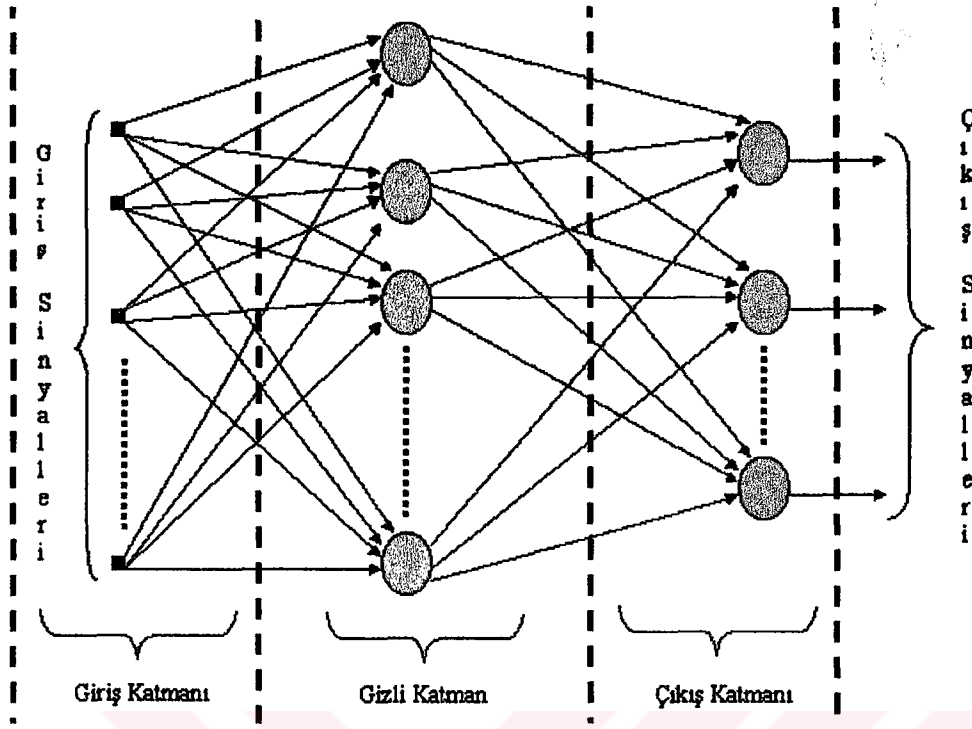
Şekil 2.7. TCMB Günlük Net İşlem Hacmi

2.2. YSA Mimarisi ve Backpropagation Algoritması

Çalışmamızda, önemli bir YSA mimarisi olan Çok Katmanlı İleri Beslemeli (Feedforward) ağ kullanılacaktır. Eğitim algoritması için, finansal uygulamalar ve tahmin içeren uygulamalarda yaygın olarak kullanılan Backpropagation (Geri Yayılım) algoritması kullanılacaktır[14].

2.2.1 Çok Katmanlı İleri Beslemeli Ağ

Çok katmanlı ileri beslemeli ağ, giriş değerlerinin aktarıldığı “giriş katmanı”; istenilen çıkış değerlerinin üretilmesini sağlayan “çıkış katmanı”, ve giriş çıkış katmanları arasında bulunan, bir veya daha fazla katmanlı “gizli katmanlardan” oluşmaktadır. Lütfen Şekil (2.8)’e bakınız [2].



Şekil 2.8. Çok Katmanlı Yapay Sinir Ağı

Backpropagation algoritması çok katmanlı ağlarda eğitim algoritması olarak yaygın bir biçimde kullanılmaktadır. Algoritma; ağdan istenen çıkış değerleriyle, ağın ürettiği gerçek çıkış değerlerinin kıyaslanması sonucu elde edilen fark değerlerinin, ağı ait parametrelerin belirlenmesi temeline dayanır.

Backpropagation eğitim algoritmasının uygulandığı ileri beslemeli çok katmanlı ağlarda; ağın işleyiş biçimi iki aşamada açıklanabilir. İlk aşamada; ağı gelen giriş sinyalleri, giriş katmanından çıkış katmanına doğru, ağın halihazırda sahip olduğu ağırlık katsayıları ve besleme sinyallerinin etkisi altında çıkışa doğru işlenerek aktarılmaktadır. İkinci aşamada ise, ilk aşamada elde edilen çıkış değerlerinin; ağın üretmesi beklenen değerlerle kıyaslanması yapılarak; aradaki fark, çıkış katmanından giriş katmanına doğru ters yönde, ağın halihazırdaki parametrelerinin yenilenmesi amacıyla aktarılmaktadır.

Ağı ardışık eğitim değerleri uygulanarak; giriş değerlerinden çıkış değerleri üretilmesi, ve üretilen çıkış değeriyle istenen çıkış değeri arasındaki farkın giriş

katmanına doğru aktarılması süreci sürekli tekrarlanarak; ağı ait parametrelerin belirli bir değere yakınsaması, yada belirli sınırlar içerisinde kalmasıyla eğitim süreci tamamlanmaktadır.

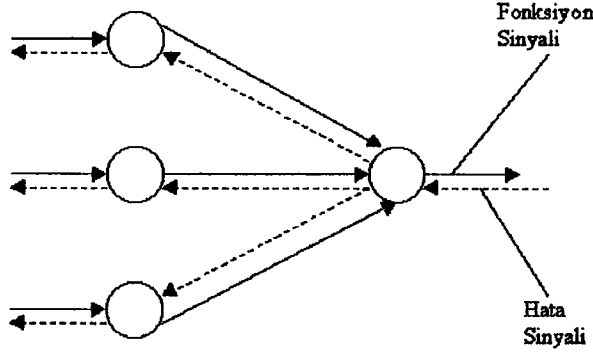
Eğitim aşamasından sonra, ağı test verileri uygulanmaktadır. Bu aşamada, ağı eğitim süresince kullanılan değerlerden farklı değerler giriş olarak uygulanmakta; böylelikle, ağı eğitim verilerinden farklı veriler için ürettiği tahmin değerlerinin performansı ölçülmektedir.

Çok Katmanlı İleri Beslemeli ağı belirleyici üç karakteristik özelliği aşağıdaki gibidir:

1. Ağ bünyesinde bulunan her bir nöron hücresi, non-lineer bir aktivasyon fonksiyonuna sahiptir. Burada not edilmesi gereken husus, ilerde ağıla ilgili temel çözümleme yapılırken de görüleceği üzere; aktivasyon fonksiyonunun türevinin alınabilir olmasıdır. Aktivasyon fonksiyonuyla ilgili bu özelliği sağlayan, ve en çok kullanılan fonksiyon; Bölüm (1.2.2)'de verilen sigmoid fonksiyonudur.
2. Çok Katmanlı ağ bünyesinde bir veya daha fazla sayıda gizli katman bulunmaktadır. Gizli katmanlar, giriş değerlerinde bulunan yapısal özelliklerin öğrenilmesini adaptif bir biçimde sağlamaktadır.
3. Çok katmanlı ağ, ağda bulunan sinapsların durumuna göre bir "bağlantılılık" (connectivity) seviyesine sahiptir. Ağı bağlantılılık seviyesindeki herhangi bir değişim, sinaptik bağlantıların sayısında yada ağırlık katsayılarında oluşan değişim anlamına gelmektedir.

Şekil (2.9)'da, Çok katmanlı ileri beslemeli ağ mimarisinde, backpropagation eğitim algoritmasının uygulanması durumunda bilgi ve sinyallerin akış yönleri gösterilmektedir. Fonksiyon sinyali, ağ girişine uygulanan giriş değerlerinin, ağ katmanları boyunca ilerleyişi boyunca, geçmiş hücreler, ve sinaptik ağırlıklara bağlı olarak değeri değişecek biçimde ilerlemekte, ve sonuçta çıkış sinyali olarak verilmektedir. Hata sinyali, bir çıkış hücresinin beklenen değeriyle, gerçekleşen değeri arasındaki fark olup; çıkış katmanından giriş katmanına doğru geri yönde

katman katman aktarılmakta, böylece, ağın istenen çıkış karakteristiğini sağlaması için sahip olması gereken parametrik özellikler bütün nöronlara bildirilmektedir.



Şekil 2.9. YSA Bilgi Akış Yönleri

2.2.2. Backpropagation (Geri-Yayılım) Algoritması

Çıkış katmanında bulunan “j” numaralı bir nöron için; n. eğitim örneğinin işlenmesi sırasındaki hata fonksiyonu aşağıdaki gibi tanımlanmaktadır [2]:

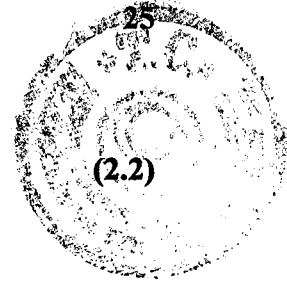
$$e_j(n) = d_j(n) - y_j(n) \quad (2.1)$$

Denklem(2.1)'deki $d_j(n)$; ağın vermesi gereken tepki sinyali olup, eğitim parametrelerinin oluşturulması aşamasında belirlenmektedir; $y_j(n)$ ise bir eğitim örneğinin işlenmesi sonucunda oluşan, çıkıştaki gerçek değerdir.

J numaralı bir nöron için, hata enerjisi, $\frac{1}{2}e_j^2(n)$ fonksiyonuyla verilmektedir.

Dolayısıyla, çıkış katmanındaki tüm nöronların bir eğitim örneğinin işlenmesi boyunca geçerli anlık hata sinyallerinin enerjileri toplamı olan $\varepsilon(n)$, fonksiyonu **denklem(2.2)** ile verilmektedir.

$$\varepsilon(n) = \frac{1}{2} \sum_{j \in C} e_j^2(n) \quad (2.2)$$



Burada C, çıkış katmanındaki hücrelerin oluşturduğu küme olarak verilmektedir.

Bir YSA'ya, eğitim sürecinde uygulanan, eğitim örneklerinin sayısı "N" ile gösterilirse; bu eğitim verilerin tamamı için, ortalama karesel hata enerji fonksiyonu olan ε_{ort} , denklem(2.3) ile verilmektedir.

$$\varepsilon_{ort} = \frac{1}{N} \sum_{n=1}^N \varepsilon(n) \quad (2.3)$$

Anlık hata enerji fonksiyonu $\varepsilon(n)$, ve ortalama karesel hata enerji fonksiyonu ε_{ort} , ağırlık sahip olduğu sinaptik ağırlıklar ve besleme seviyeleri gibi parametrelerin bir fonksiyonudur. YSA'nın eğitim sürecindeki amaç, ortalama karesel hata enerji fonksiyonu ε_{ort} minimum olacak şekilde, ağırlık parametrelerinin ayarlanmasıdır.

Bölüm(1.2.2.)'de verilen, Denklem (1.3)'te belirtildiği gibi; "j" numaralı bir nörona, girişinde bulunan diğer hücreler tarafından aktarılan sinyallerin toplamı, denklem(2.4) ile verilmektedir.

$$v_j(n) = \sum_{i=0}^m w_{ji}(n) y_i(n) \quad (2.4)$$

Denklem(2.4)'te; m, girişlerin toplam sayısını vermektedir. Dolayısıyla, "j" numaralı nöron için, n. eğitim örneğinin uygulanması sonucu elde edilen çıkış sinyali, denklem(2.5) ile verilebilir.

$$y_j(n) = f_j(v_j(n)) \quad (2.5)$$

Denklem(2.5)'te verilen $f_j(.)$ fonksiyonu, bu nörona ait aktivasyon fonksiyonudur.

Geri yayılım algoritmasında, hata'yı minimize etmek için, $w_{ji}(n)$ şinaptik ağırlığına; anlık hata enerjisinin, şinaptik ağırlığa göre kısmi türevi olan $\frac{\partial \varepsilon(n)}{\partial w_{ji}(n)}$ ile orantılı olan $\Delta w_{ji}(n)$ oranında bir düzeltme uygulanmaktadır. Uygulanan düzeltme değeri $\Delta w_{ji}(n)$ aşağıdaki denklemler kullanılarak türetilmektedir;

$$\frac{\partial \varepsilon(n)}{\partial w_{ji}(n)} = \frac{\partial \varepsilon(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \frac{\partial v_j(n)}{\partial w_{ji}(n)} \quad (2.6)$$

Dikkat edileceği üzere, denklem(2.6) daki kısmi türev için zincirleme kuralı uygulanmıştır.

Denklem(2.2)'nin her iki tarafının $e_j(n)$ 'e göre türevini alırsak:

$$\frac{\partial \varepsilon(n)}{\partial e_j(n)} = e_j(n) \quad (2.7)$$

Keza, denklem(2.1.)'in her iki tarafının $y_j(n)$ 'e göre türevini alırsak;

$$\frac{\partial e_j(n)}{\partial y_j(n)} = -1 \quad (2.8)$$

Denklem(2.5)'in $v_j(n)$ 'e göre türevi:

$$\frac{\partial y_j(n)}{\partial v_j(n)} = f'_j(v_j(n)) \quad (2.9)$$

verir.

Ayrıca, son olarak denklem(2.4)'ün $w_{ji}(n)$ 'e göre türevide aşağıdaki sonucu verir:

$$\frac{\partial v_j(n)}{\partial w_{ji}(n)} = y_i(n) \quad (2.10)$$

Denklem(2.7), Denklem(2.8), Denklem(2.9), ve Denklem(2.10)'u; Denklem(2.6)'da kullanırsak;

$$\frac{\partial \varepsilon(n)}{\partial w_{ji}(n)} = -e_j(n) f'_j(v_j(n)) y_i(n) \quad (2.11)$$

Sonucunu elde ederiz.

Uygulanan $\Delta w_{ji}(n)$ düzeltme oranı, denklem(2.12) ile verilmektedir.

$$\Delta w_{ji}(n) = -\eta \frac{\partial \varepsilon(n)}{\partial w_{ji}(n)} \quad (2.12)$$

Burada verilen η katsayısı backpropagation algoritmasının öğrenim oranı katsayısı olarak bilinmektedir. Denklem(2.11)'i, denklem(2.12)'de kullanırsak; uygulanan düzeltme oranı için denklem(2.13)'ü elde ederiz.

$$\Delta w_{ji}(n) = \eta \delta_j(n) y_i(n) \quad (2.13)$$

Burada, kullanılan $\delta_j(n)$, yerel (lokal) gradyan olup, denklem(2.14) ile ifade edilmektedir. Denklem (2.14)'den de görüleceği üzere, hata değirindeki artış,

gradyan değerinde azalmaya, hata değerindeki azalma ise gradyan'da artışa neden olmaktadır. Dolayısıyla $\delta_j(n)$, sinaptik ağırlıklar için gerekli olan değişim yönünü işaret etmektedir.

$$\delta_j(n) = - \frac{\partial \varepsilon(n)}{\partial v_j(n)} \quad (2.14)$$

Denklem(2.14), zincirleme kuralına göre aşağıdaki gibi yazılabilir;

$$\delta_j(n) = - \frac{\partial \varepsilon(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \quad (2.15)$$

Denklem(2.15)'deki kısmi türev ifadelerinin yerine, Denklem(2.7),Denklem(2.8), ve Denklem(2.9)'daki açık ifadeleri kullanırsak, Denklem(2.16)'yı elde ederiz.

$$\delta_j(n) = e_j(n) f'_j(v_j(n)) \quad (2.16)$$

Denklem(2.16)'yı, Denklem(2.13)'de kullanırsak, düzeltme katsayısı $\Delta w_{ji}(n)$ için, Denklem(2.17)'yi elde ederiz.

$$\Delta w_{ji}(n) = \eta e_j(n) f'_j(v_j(n)) y_i(n) \quad (2.17)$$

Sinaptik ağırlıklara uygulanan düzeltme oranı $\Delta w_{ji}(n)$, ilgili nöronun bir çıkış katmanı nöronu, yada gizli katmanda bulunan nöron olmasına göre farklı açılardan incelenmelidir.

Eğer düzeltmenin uygulandığı nöron çıkış katmanındaki bir hücreyse, istenen değerle, hücre çıkışında bulunan değer, nöronun çıkış katmanında bulunması nedeniyle kolayca hesaplanabilmektedir.

Eğer düzeltmenin uygulanacağı nöron, gizli katmanda bulunuyorsa, bu durumda, istenen hedef değerle, gizli katmanda bulunan bu nöronun çıkış değeri arasındaki fark kolayca bulunamayacağından, aşağıdaki prosedür takip edilmelidir.

Denklem(2.15); aşağıdaki biçimde yazılabilir:

$$\delta_j(n) = - \frac{\partial \varepsilon(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \quad (2.18)$$

Denklem(2.18)'de denklem(2.9)'u kullanırsak;

$$\delta_j(n) = - \frac{\partial \varepsilon(n)}{\partial y_j(n)} f'_j(v_j(n)) \quad (2.19)$$

Denklem(2.19)'daki ilk kısmi türevi hesaplamak için, aşağıdaki işlemler gerçekleştirilmektedir.

$$\varepsilon(n) = \frac{1}{2} \sum_{k \in C} e_k^2(n) \quad (2.20)$$

Denklem(2.20), gizli katmanda bulunan j nöronundan bir sonraki katmanda bulunan nöronların anlık hata enerjisi fonksiyonlarını vermektedir. Bu durumda, denklem(2.19)'daki ilk kısmi türev denklem(2.21)'deki gibi hesaplanabilir:

$$\frac{\partial \varepsilon(n)}{\partial y_j(n)} = \sum_k e_k(n) \frac{\partial e_k(n)}{\partial y_j(n)} \quad (2.21)$$

Denklem(2.21)'deki kısmi türevi Denklem(2.22)'deki gibi yazabiliriz.

$$\frac{\partial \varepsilon(n)}{\partial y_j(n)} = \sum_k e_k(n) \frac{\partial e_k(n)}{\partial v_k(n)} \frac{\partial v_k(n)}{\partial y_j(n)} \quad (2.22)$$

Gizli katmanda bulunan j nöronunun sağındaki katmanda bulunan k nöronu için, hata fonksiyonu , denklem(2.23)'de verilmiştir.

$$e_k(n) = d_k(n) - y_k(n) \quad (2.23)$$

Denklem(2.23)'deki $d_k(n)$, k numaralı nöronun üretmesi istenen değeri, $y_k(n)$ ise n. Eğitim verisinin işlenmesi sonucunda üretilen gerçek değeri vermektedir.

Denklem(2.23)'deki $y_k(n)$ değerini aşağıdaki gibi yazabiliriz.

$$y_k(n) = f_k(v_k(n)) \quad (2.24)$$

Bu durumda, denklem (2.22)'deki ilk kısmi türev aşağıdaki gibi ifade edilebilir.

$$\frac{\partial e_k(n)}{\partial v_k(n)} = -f'_k(v_k(n)) \quad (2.25)$$

Keza, k numaralı nörona aktarılan toplam potansiyeli denklem(2.26)'daki gibi ifade edebiliriz:

$$v_k(n) = \sum_{j=0}^m w_{kj}(n) y_j(n) \quad (2.26)$$

Burada, m sayısı, k numaralı nörondan önceki gizli katmanda bulunan nöronların sayısını vermektedir. Bu durumda, denklem(2.22)'nin sağ tarafındaki ikinci kısmi türev denklem(2.27)'deki gibidir.

$$\frac{\partial v_k(n)}{\partial y_j(n)} = w_{kj}(n) \quad (2.27)$$

Denklem(2.25) ve Denklem(2.27)'yi denklem(2.22)'de kullanırsak, denklem(2.28)'i elde ederiz.

$$\frac{\partial \varepsilon(n)}{\partial y_j(n)} = - \sum_k e_k(n) f'_k(v_k(n)) w_{kj}(n) \quad (2.28)$$

Denklem(2.28), denklem(2.16) yardımıyla, denklem(2.29) biçiminde yazılabilir.

$$\frac{\partial \varepsilon(n)}{\partial y_j(n)} = - \sum_k \delta_k(n) w_{kj}(n) \quad (2.29)$$

Denklem(2.29)'u denklem(2.19)'da kullanırsak, denklem(2.30)'u elde ederiz.

$$\delta_j(n) = f'_j(v_j(n)) \sum_k \delta_k(n) w_{kj}(n) \quad (2.30)$$

Yukarıda verilen açıklamalar ışığında, gizli katmanda bulunan j numaralı bir nörona ait sinaptik katsayıya uygulanacak düzeltme oranı $\Delta w_{ji}(n)$ için;

$\Delta w_{ji}(n) = \eta \delta_i(n) y_i(n)$ olduğu göz önüne alınırsa, düzeltme oranı, denklem(2.30) yardımıyla, denklem(2.31) biçiminde ifade edilmektedir.

$$\Delta w_{ji}(n) = \eta \left(f'_j(v_j(n)) \sum_k \delta_k(n) w_{kj}(n) \right) y_i(n) \quad (2.31)$$

YSA bünyesinde bulunan bir nörona uygulanacak düzeltme katsayısı $\Delta w_{ji}(n)$ 'in türetilmesiyle ilgili bu tartışmalardan sonra, backpropagation algoritmasının uygulandığı Çok Katmanlı İleri Beslemeli Ağ için, eğitim prosedürünü özet olarak aşağıdaki maddelerle ifade edebiliriz:

1. YSA'ya uygulanacak giriş verileri ve bu giriş verilerinden türetilen hedef çıkış verileri, ağın yapısına uygun olarak üretilir. Bu aşama, ağ tarafından işlenecek verilerin hazırlanması olarak bilinir.
2. YSA'daki nöronlara ait parametreler ve öğrenim oranı η gibi değerler için ilk değer atamaları (initialization) gerçekleştirilir.
3. Eğitim verilerinden "n." grubun işlenmesi sırasında; giriş sinyallerinin ağ yapısında ileri doğru işlenmesi aşamasında, halihazırdaki parametreler kullanılır, ve buna göre çıkışlar üretilir.
4. Eğitim verilerinden "n." grubun işlenmesi sırasında; üretilen gerçek çıkış değerleriyle, nöronların üretmesi beklenen hedef değerler arasındaki fark, ve yukarda denklem (2.1) ve Denklem(2.32)'ye kadar çıkartılan sonuçlar doğrultusunda elde edilen nöronlara ait sinaptik ağırlıklara uygulanacak düzeltme oranı $\Delta w_{ji}(n)$ 'nin kullanılmasıyla; sinaptik ağırlıklar aşağıdaki denklem(2.32) yardımıyla yenilenir. Sinaptik ağırlıklara uygulanan düzeltme

oranı, ilgili nöronun çıkış katmanında yada gizli katmanlarda bulunmasına göre denklem(2.17) veya denklem(2.31)'e göre hesaplanır.

$$w_{ji}(n+1) = \Delta w_{ji}(n) + w_{ji} \quad (2.32)$$

5. 3. ve 4. aşamalar, eğitim verilerindeki örnek değerlerin tamamı için, örnek değerler rastgele kullanılacak biçimde “rastgele” (random) olarak tekrarlanır. Eğitim sürecinin bitişi, ilgili parametrelerin belirli bir değere yakınsaması, veya gerçek çıkış değerleriyle, hedef çıkış değerleri arasındaki farkın minimize edildiği duruma kadar tekrarlanır.

2.2.2.1 Aktivasyon Fonksiyonu

Backpropagation öğrenim algoritmasını uygulayacağımız Çok Katmanlı İleri beslemeli ağdaki her bir nöron için, kullanılacak aktivasyon fonksiyonlarının; $\Delta w_{ji}(n)$ düzeltme katsayısı için türetilen denklem(2.17) ve Denklem(2.31)'den anlaşıldığı gibi, birinci dereceden türevinin alınabilir olması gerekmektedir. olmalıdır. Matematiksel anlamda, türevleri alınabilir fonksiyonlar, “düzgün” fonksiyonlar olarak adlandırılmaktadır.

Biz bu çalışmamızda, finansal uygulamalarda çok yaygın olarak kullanılan, sigmoid fonksiyonunu, aktivasyon fonksiyonu olarak kullanacağız. Sigmoid fonksiyonunun denklemi, 1. bölümde, denklem(1.9) ile verilmektedir. Burada, uygulanan prosedürün gösterimi açısından aynı denklem tekrarlanacaktır.

$$f_j(v_j(n)) = \frac{1}{1 + \exp(-av_j(n))} \quad (2.33)$$

Denklem(2.33)'ün, v 'ye göre türevini alırsak;

$$f_j'(v_j(n)) = \frac{a \cdot \exp(-a \cdot v_j(n))}{[1 + \exp(-a v_j(n))]^2} \quad (2.34)$$

Denklem(2.34), aşağıdaki gibi ifade edilebilir.

$$f_j'(v_j(n)) = a \cdot f_j(v_j(n)) \{1 - f_j(v_j(n))\} \quad (2.35)$$

Denklem(2.35)'deki $f_j(v_j(n))$ ifadesi, "j" nöronuna ait çıkış değeri olup, $y_j(n)$ olarak ifade edilir. Bu durumda, denklem(2.34) ve denklem(2.35) yardımıyla, sigmoid aktivasyon fonksiyonuna sahip bir nöron için, aktivasyon fonksiyonu $f_j(n)$ 'in türevi; hücre çıkışında verilen $y_j(n)$ değeri kullanılarak denklem(2.36)'daki gibi ifade edilebilir.

$$f_j'(v_j(n)) = a \cdot y_j(n) \{1 - y_j(n)\} \quad (2.36)$$

Denklem(2.36) kullanılarak, "j" nöronuna uygulanan düzeltme değeri $\Delta w_{ji}(n)$ değeri, eğer "j" nöronu çıkış katmanında ise, çıkış katmanında bulunan nöronlar için geçerli denklem(2.17) kullanılarak denklem(2.37)'de verildiği gibi ifade edilebilir.

$$\Delta w_{ji}(n) = [\eta(d_j(n) - y_j(n))] a \cdot y_j(n) [1 - y_j(n)] y_i(n) \quad (2.37)$$

Denklem(2.37)'deki $y_j(n)$ ile $y_i(n)$ ifadelerini birbiriyle karıştırmamak için; "j" numaralı nörona ait çıkış değerini $o_j(n)$ ile ifade edersek; çıkış katmanındaki "j"

nöronunun şinaptik ağırlıklarına uygulanacak düzeltme değeri için, denklem(2.38)'i elde ederiz.

$$\Delta w_{ji}(n) = [\eta(d_j(n) - o_j(n))] a.o_j(n)[1 - o_j(n)] y_i(n) \quad (2.38)$$

Keza, gizli katmanda bulunan “j” nöronu içinde, denklem(2.31)'i kullanarak, şinaptik ağırlıklara uygulanan $\Delta w_{ji}(n)$ değerini, denklem(2.39)'daki gibi ifade edebiliriz. Burada $y_j(n)$ ifadesi için, herhangi bir karışıklığa yol açmamak amacıyla, gizli katmandaki “j” nöronu için $g_j(n)$ ifadesi kullanılmıştır.

$$\Delta w_{ji}(n) = \eta(a.g_j(n)[1 - g_j(n)]) \cdot \left[\sum_k \delta_k(n) w_{kj}(n) \right] y_i(n) \quad (2.39)$$

2.2.2.2 Öğrenim Oranı

Backpropagation öğrenim algoritmasında kullanılan öğrenim oranı η 'nın (Bkz. denklem 2.12) küçük değerleri için, denklem(2.17) ve denklem(2.31)'dende anlaşılacağı üzere, eğitim örneklerinin kullanıldığı her bir eğitim çevriminde, nöronlara ait şinaptik ağırlık katsayılarının az miktarlarda değişmesine yol açacaktır. Ancak bu durumda, ağırlık katsayıları yeterli hızda ulaşmaları gereken değere ulaşamayacaklarından, YSA'nın öğrenme süresi uzayacaktır. Öğrenim oranının çok büyük olarak seçilmesi durumunda ise, her bir çevrimde, ağırlık katsayılarındaki değişim büyüyeceğinden, sistemin kararsız duruma gelmesi, yani ağırlık katsayılarının herhangi bir minimum değere yakınsayamaması söz konusu olacaktır.

Öğrenim oranının değerini, sistemde kararsızlığa yol açmadan arttırabilmenin bir yolu da, denklem(2.38) ve denklem(2.39)'da verilen, düzeltme değerinin denklem(2.40) ve denklem(2.41)'deki gibi düzenlenmesidir.

$$\Delta w_{ji}(n) = \alpha \Delta w_{ji}(n-1) + \eta (d_j(n) - o_j(n)) \cdot a.o_j(n) [1 - o_j(n)] y_i(n) \quad (2.40)$$

$$\Delta w_{ji}(n) = \alpha \Delta w_{ji}(n-1) + \eta (a.g_j(n) [1 - g_j(n)]) \cdot \left[\sum_k \delta_k(n) w_{kj}(n) \right] y_i(n) \quad (2.41)$$

α katsayısı, momentum sabiti olarak adlandırılmaktadır ve genelde pozitif bir değer olup, öğrenim oranından kaynaklanabilecek kararsızlıkların sınırlanmasını sağlamaktadır.

2.3. Eğitim Verilerinin İşlenmesi

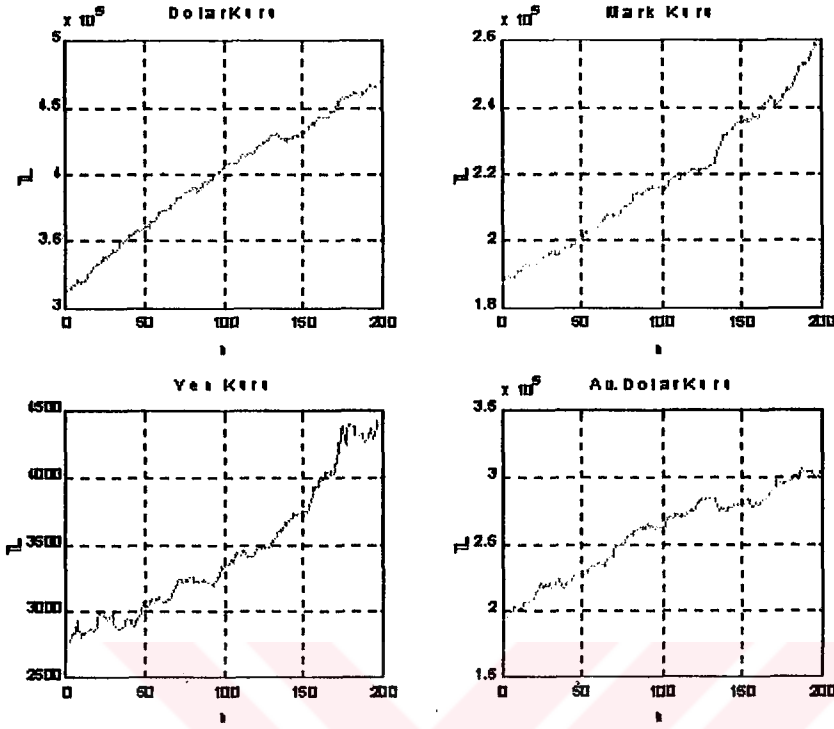
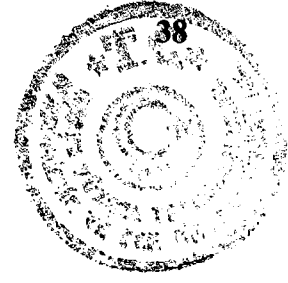
Eğitim ve test verilerinden oluşan veri gruplarının, YSA tarafından işlenebilmesi için uygun forma getirilmesi gerekmektedir. Örneğin ağ bünyesinde bulunan nöronlar için, sigmoid aktivasyon fonksiyonunun kullanılması ve bu fonksiyonun değerinin [0,1] aralığında değişmesi, ham veri gruplarının, ağ yapısına uygun değerlere dönüştürülmesini gerektirmektedir.

Bölüm(2.1.2)'de de verildiği gibi, döviz kurlarının tahmininde kullanılacak değerler aşağıda tekrar edilmiştir.

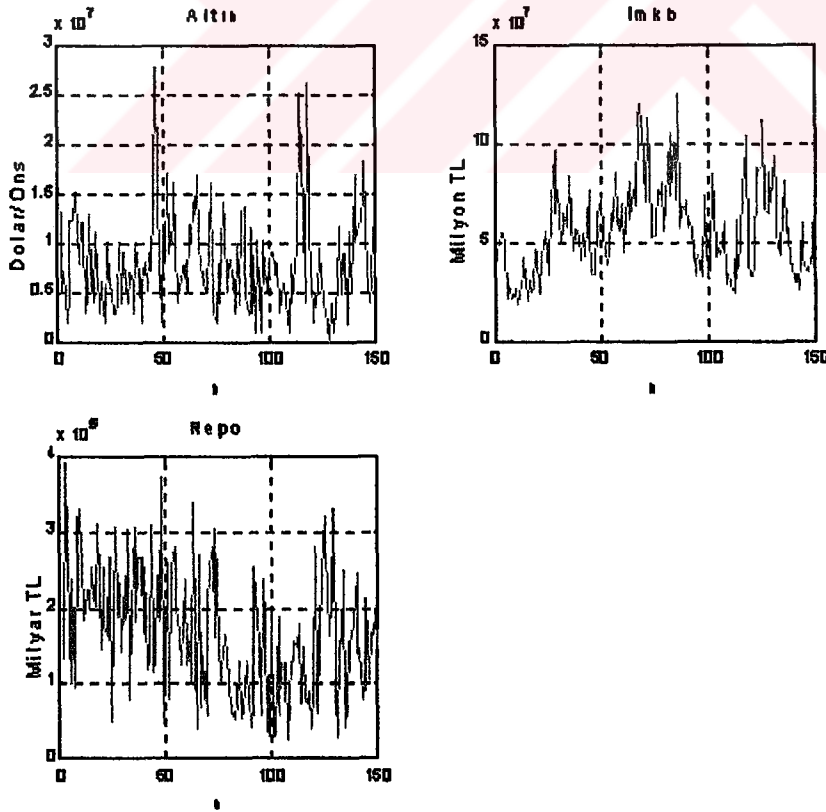
- ABD Doları – Döviz Alış,
- Alman Markı – Döviz Alış,
- Japon Yeni – Döviz Alış,
- Avusturalya – Döviz Alış,
- İstanbul Altın Borsası İşlem Hacmi (Dolar/Ons),
- İstanbul Menkul Kıymetler Borsası (İMKB) günlük işlem hacmi,
- T.C. Merkez Bankası tarafından gerçekleştirilen günlük net repo hacmi.

Ham veri gruplarının, örnek numarası “n”e göre grafikleri; döviz kurları için Şekil(2.10) ve döviz kurlarının gelecekteki değerlerinin tahmininde yardımcı veri olarak kullanacak veriler için Şekil(2.11)’de verilmektedir.





Şekil 2.10. Döviz Kurları İçin Ham Veri Grupları



Şekil 2.11. Yardımcı Veriler İçin Ham Veri Grupları

2.3.1. Giriş Değerlerinin Normalleştirilmesi

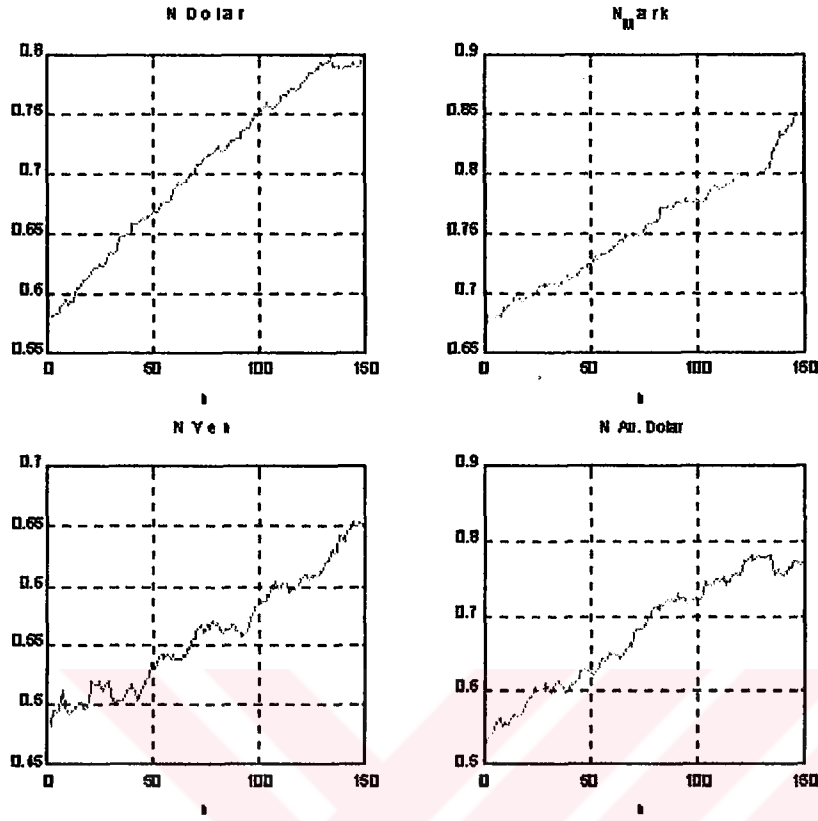
Şekil(2.10) ve Şekil(2.11)'dede görüleceği üzere, döviz kurları, ve yardımcı verilerin değişim aralığı, Çok Katmanlı İleri Beslemeli Ağ yapısında kullanılan nöronların, sigmoid aktivasyon fonksiyonlarıyla ele alınabilecek değerlerden çok yüksektir.

Döviz kurlarının, ve yardımcı verilerin; gerçek değerlerindeki bu fazlalık, aşağıda verilen denklem(2.42) yardımıyla normalleştirilebilir.

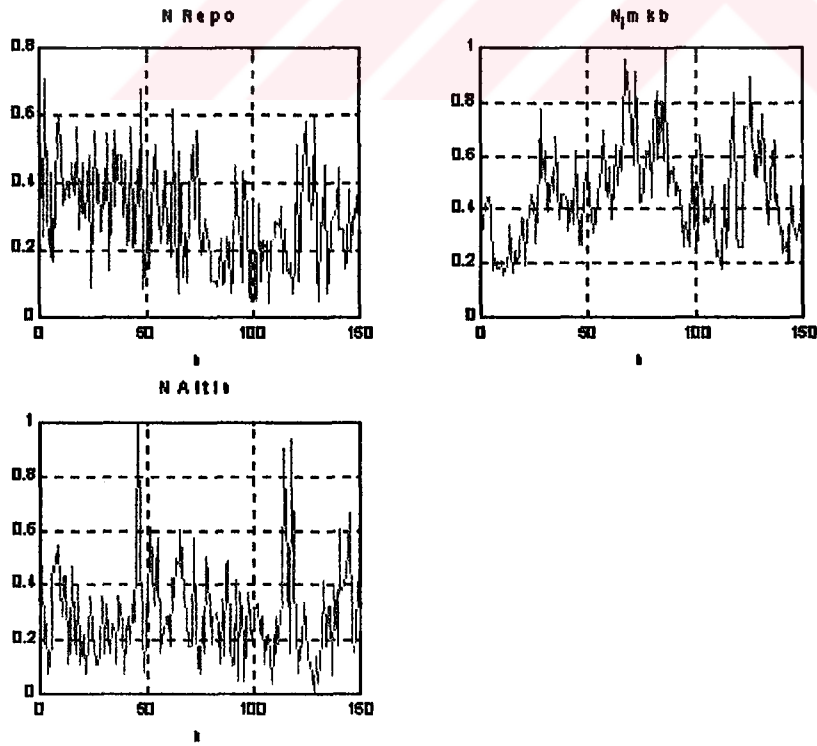
$$\text{Yeni Değer} = (\text{Eski Değer}) / (\text{Maksimum Değer}) \quad (2.42)$$

Denklem(2.42)'de Eski Değer, ham verinin gerçek değerini, Maksimum Değer, ilgili veri grubu içerisindeki maksimum değeri, Yeni Değer ise Denklem(2.42)'ye göre hesaplanan değeri vermektedir.

Denklem(2.46)'nın kullanımıyla; Dolar, Mark, Yen, ve Avusturalya Doları döviz kurlarının değerleri ile repo, altın, İmkb yardımcı verileri için; Şekil(2.12), ve Şekil(2.13)'de verilen normalleştirilmiş değerler için N_Dolar, N_Mark, N_Yen, ve N_Av.Dolar ; ile N_repo, N_altın, ve N_İmkb grafikleri oluşturulmuştur (Baştaki N_ ibaresi; ilgili veri grubunun normalleştirilmiş halini ifade etmek için kullanılmıştır).



Şekil 2.12. Döviz Kurları İçin Normalleştirilmiş Veri Grupları



Şekil 2.13. Yardımcı Veriler İçin Normalleştirilmiş Veri Grupları

2.3.2. Hedef Değerler

Backpropagation algorimasının uygulanacağı YSA için ilgili giriş değerlerini ürettikten sonra (normalleştirilmiş giriş değerleri; toplam 7 adet giriş değeri bulunmaktadır), YSA'nın çıkış olarak vermesi istenilen döviz kurlarına ait hedef değer, 1 iş günü sonraki değer olarak belirlenmektedir.

Örneğin Amerikan Doları döviz alış kuruna ait tahmin için kullanılacak veriler aşağıdaki gibidir.

n. gündeki giriş verileri:

N_Dolar(n),
 N_Mark(n),
 N_Yen(n),
 N_Av.Dolar(n),
 N_repo(n),
 N_altın(n),
 N_İmkb(n)

Hedef Veri:

N_Dolar(n+1)

Dikkat edileceği üzere hedef değerler normalleştirilmiş veriler içerisinden seçilmektedir, dolayısıyla YSA yapısında işlemeye hazır halde bulunmaktadırlar.

3. YSA UYGULAMA PROGRAMI

3.1. Uygulama Programının Kullanımı

Uygulama programının açıklaması üç kısımda yapılacaktır.

Birinci kısımda (Bölüm. 3.1.1.); yapay sinir ağının kurulumu, ilk değer atamaları, ve eğitim verilerinin kullanımı vasıtasıyla ağırlık katsayılarının belirlendiği Eğitim Programı açıklanacaktır.

İkinci kısımda (Bölüm.3.1.2); Eğitim aşamasında üretilen parametreleri kullanan Test Programının işleyişi açıklanacaktır.

Üçüncü kısımda (Bölüm 3.1.3); Eğitim ve Test Programları tarafından kullanılan ek programlar açıklanacaktır.

3.1.1. Eğitim Programı

MATLAB (Version 5.1) ile programlanan Eğitim Programı: Yapay Sinir Ağının yapısal özelliklerinin tanımlanmasını, ilk değer atamalarını; ve Eğitim verilerinin rastgele düzen içerisinde çeşitli çevrim (iterasyon) sayılarında kullanımıyla, YSA parametrelerinin ortalama karesel hata fonksiyonunun kabul edilebilir değerlere yakınsamasıyla sonuçlanacak aşamaları içermektedir.

Programın işleyişi sırasında; kullanıcının belirlemesi istenilen parametreler (Ağ mimarisi, Eğitim Sayısı, Öğrenim Oranı) giriş olarak alınmakta, daha sonra bu parametrelere göre kurulan yapay sinir ağı vasıtasıyla gerçekleştirilen hesaplamalar sonucunda, ortalama karesel hata fonksiyonunun değerleri istenilen eğitim çevrimi sayısına göre hesaplanmaktadır.

Eğitim Programının İşleyişiyle ilgili bir örnek aşağıda sunulmaktadır:

»

»

» **EGITIM**

Katman Sayısını Giriniz: 3

Lütfen

i =

2

. katmandaki nöron sayısını giriniz: 10

Öğrenim oranına ait alfa momentum sabitini giriniz, alfa = 0.5

Öğrenim oranına ait eta sabitini giriniz, eta = 0.4

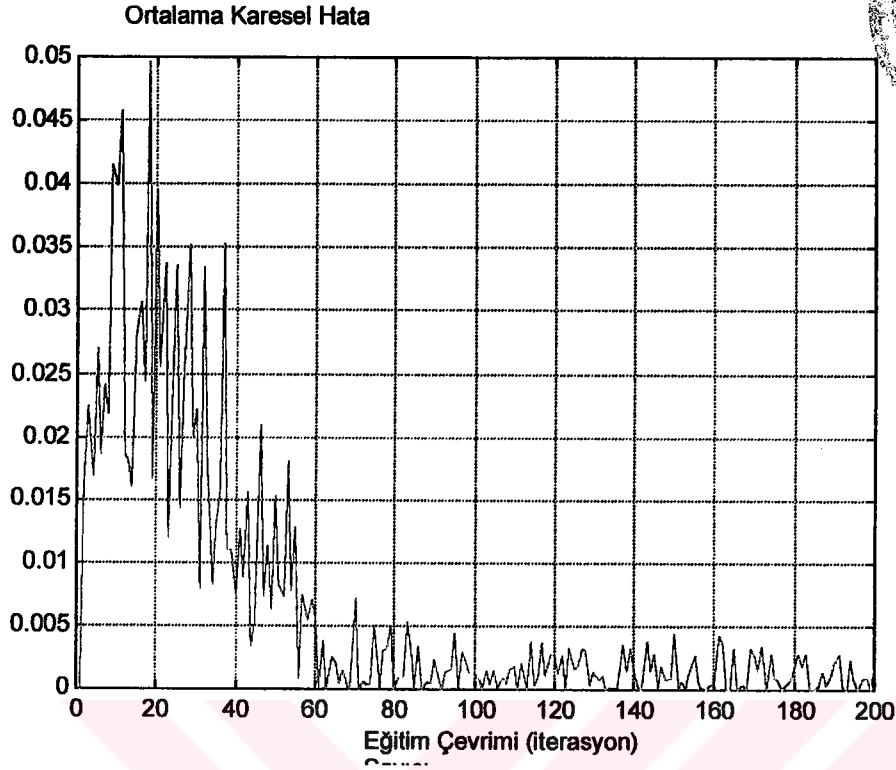
Eğitim Çevrimi Sayısını Giriniz: 200

»

Kullanıcı EGITIM programını başlattıktan sonra;

1. YSA'nın kaç katmanlı olacağı, ve ilgili katmanlardaki hücre sayıları sorulmaktadır, daha sonra bu değerlere göre ağırlık matrisine rastgele ilk değer atamaları gerçekleştirilmekte ve mimari kurulmaktadır. Örnekte verilen değerler için oluşturulan mimari matrisi; $mimari = [7 \ 10 \ 1]$ 'dir. Burada matrisin birinci sütunu birinci katmana, ikinci sütunu ikinci katmana ve her bir sütunda verilen sayı ilgili katmandaki nöron sayısına karşılık gelmek üzere her bir sütun yapay sinir ağına ait katmanları belirlemektedir (örnekte; 2. Katmanda 10 gizli nöron bulunmaktadır). Burada not edilmesi gereken husus, YSA'ya ait giriş değerlerinin 7 adet (ABD Doları, Alman Markı, Japon Yeni, Avusturalya Doları, İstanbul Altın Borsası İşlem Hacmi, İMKB günlük işlem hacmi, TCMB günlük net repo işlem hacmi) olması ve tahmini edilecek döviz kurunda 1 adet olması nedeniyle YSA'nın giriş ve çıkış katmanlarının önceden belirlenmiş olduğudur.
2. Momentum katsayısı "alfa" ve öğrenim oranı "eta" değerleri girilmekte,
3. Eğitim Aşamasında YSA'ya uygulanacak eğitim çevrimi sayısı "N" (iterasyon sayısı) girilmektedir.

Program, hesaplamaları yaptıktan sonra; belirlenen mimari ve parametreler için ilgili döviz kuruna ait "Ortalama Karesel Hata Foksiyonu" değerlerini eğitim aşaması sonunda grafik olarak vermektedir. (Bknz. Şekil 3.1)



Şekil 3.1.Ortalama Karesel Hata Fonksiyonu

Eğitim aşamasında, YSA mimarisi ve ilgili parametrelere bağlı olarak ortalama karesel hata fonksiyonunun belirli değerlere yakınsaması ,kullanıcının tercihiine göre, kabul edilebilir sınırlara ulaştığında eğitim aşaması bitirilmektedir. Bu şekilde eğitilen YSA daha sonra Test Değerlerine uygulanmakta ve ağın performansı değerlendirilmektedir.

EGITIM Programının kod dökümü EK 1’de verilmektedir. :

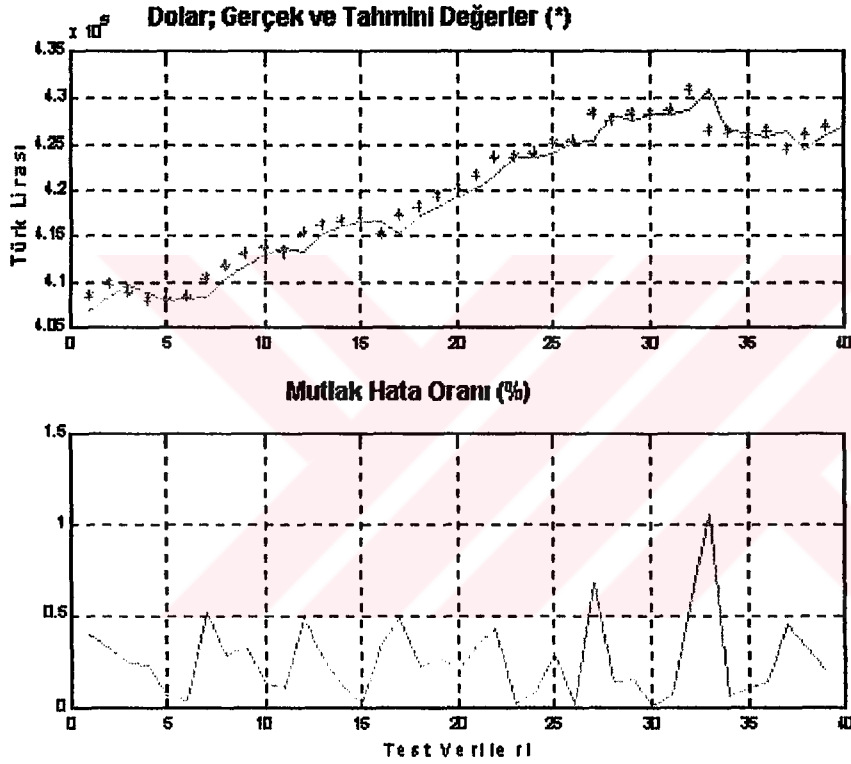
3.1.2. Test Programı

Test Programı, Eğitim aşaması sonucunda elde edilen YSA’ya ait ağırlık katsayılarını ve diğer parametreleri kullanarak, sistemin eğitim verileri dışındaki veriler için ürettiği değerleri kullanıcının dikkatine sunar. Böylelikle, Ağı tahmin sürecinde kullanılma yeterliliği değerlendirilmiş olur.

Test Programının İşleyişiyle ilgili bir örnek aşağıda sunulmaktadır (EGITIM programını takiben)

»
 »
 » TEST
 »

Kullanıcı TEST Programını başlattığında; program, Test Dönemi içerisinde; ilgili döviz kurunun gerçek değerlerini, YSA ile belirlenen tahmini değerleri, ve mutlak hatanın oranını % olarak grafik üzerinde vermektedir. (Bakınız Şekil.3.2)



Şekil 3.2. Dolar Kuru İçin Test Sonuçları

Yüzdelik Mutlak Hata Oranı hesaplanırken aşağıdaki denklem.3.1. kullanılmaktadır:

$$\text{Mutlak Hata Oranı (\%)} = 100 \cdot (\text{Tahmin} - \text{Gerçek Değer}) / (\text{Tahmin}) \quad (3.1)$$

TEST Programının kod dökümü EK 1'de verilmektedir.

3.1.3. Ek Program Ve Fonksiyonlar

Hem EGITIM hemde TEST programları; “nöron fonksiyonunu” ; ve “veriler” programını kullanmaktadır.

“Nöron” fonksiyonu, Yapay Sinir Ağı bünyesinde kullanılan nöronlara ait parametreleri içermekte ve YSA’nın eğitim ve test aşamalarında ilgili programlar tarafından kullanılmaktadır.

Nöron Fonksiyonunun kod dökümü EK 1’de verilmektedir.

“Veriler” programı; Yapay Sinir Ağı tarafından kullanılan eğitim ve test verilerinin yüklenmesini sağlamaktadır. Ayrıca, ilgili veri gruplarına ait normalleştirilmiş değerler bu program tarafından oluşturulmaktadır.

“Veriler” programına ait kod dökümü EK 1’de verilmektedir.

4. OPTİMAL PARAMETRELERİN SEÇİMİ VE PROBLEMİN ÇÖZÜMÜNÜN SAYISAL OLARAK GERÇEKLEŞTİRİLMESİ

Bu bölümde, Yapay Sinir Ağı mimarisinin; geriyayılım (backpropagation) algoritmasının kullanımı ve ağı ait; katman sayısı, gizli katmanlardaki nöronların sayısı, öğrenim oranı, momentum sabiti, uygulanan eğitim çevrim sayısı gibi çeşitli parametrelerinin değişik değerlerinin ağı tahmin yeteneği üzerindeki etkileri incelenecek; ve döviz kurları için optimal ağ parametreleri belirlenecektir.

4.1. Parametreler için Optimal Değerlerin Seçimi

Problemin çözümü, pratik – deneysel açıdan en iyi yakınsamayı sağlayan ağ parametrelerinin seçimiyle sağlanacaktır.

Tablo 4.1.de YSA mimarisinde kullanılan parametreler verilmektedir.

Parametre	Sembol	Tipik Değişim Aralığı
YSA Mimarisi	mimari	$(2, \infty)$
Öğrenim Oranı	η	$(0,1)$
Momentum Sabiti	α	$(0,1)$
Eğitim Çevrim Sayısı	N	$(1, \infty)$

Tablo 4.1 YSA Parametreleri

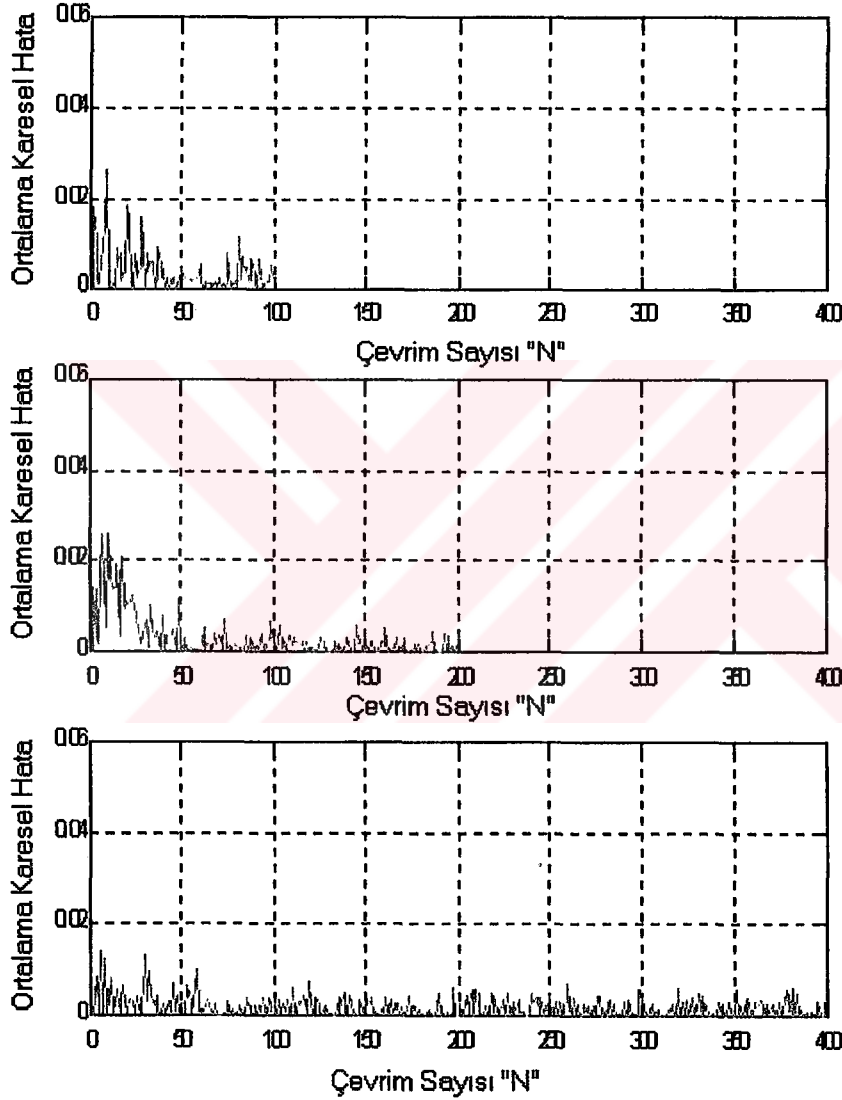
4.1.1. Eğitim İçin Uygulanan Çevrim (İterasyon) Sayısı “N”

Bu kısımda, ağı ait diğer parametrelerin sabit değerleri için, Eğitim için uygulanan çevrim sayısı “N” ‘nin değişimine göre ortalama karesel hata fonksiyonunun değişimi incelenecektir.

Tablo 4.2’de verilen sabit değerler için; Ortalama Karesel Hata Fonksiyonunun, çevrim sayısı N’ye göre değişim grafikleri Şekil 4.1 de verilmektedir.

N =100; 200; 400	Mimari	[7 1]
	η	0.1
	α	0.9

Tablo 4.2. Eğitim Çevrim Sayısına Göre Değişim, (Mimari = [7 1])

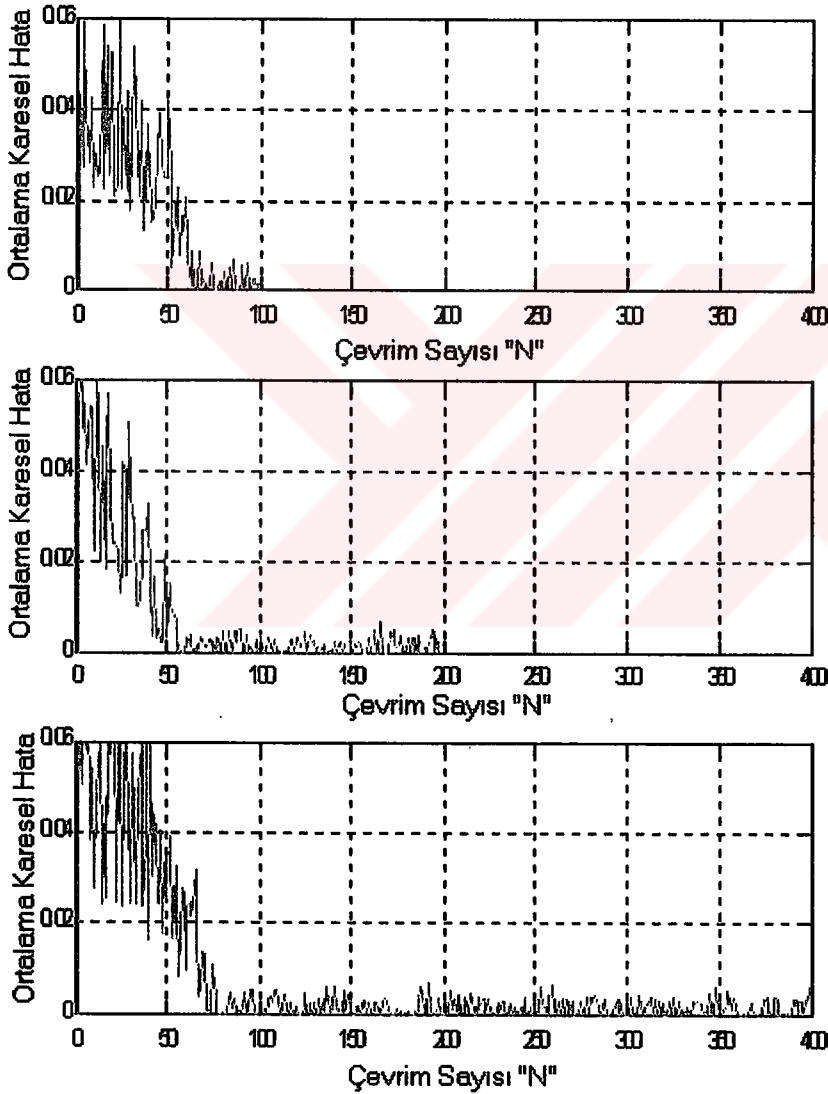


Şekil 4.1. Ortalama Karesel Hata'nın N'e göre Değişimi (Mimari =[7 1])

Tablo 4.3'de verilen sabit değerler için; Ortalama Karesel Hata Fonksiyonunun, Eğitim Sayısı N'ye göre değişim grafikleri Şekil 4.2 de verilmektedir.

N =100; 200; 400	Mimari	[7 10 1]
	η	0.1
	α	0.9

Tablo 4.3. Eğitim Çevrim Sayısına Göre Değişim, (Mimari = [7 10 1])

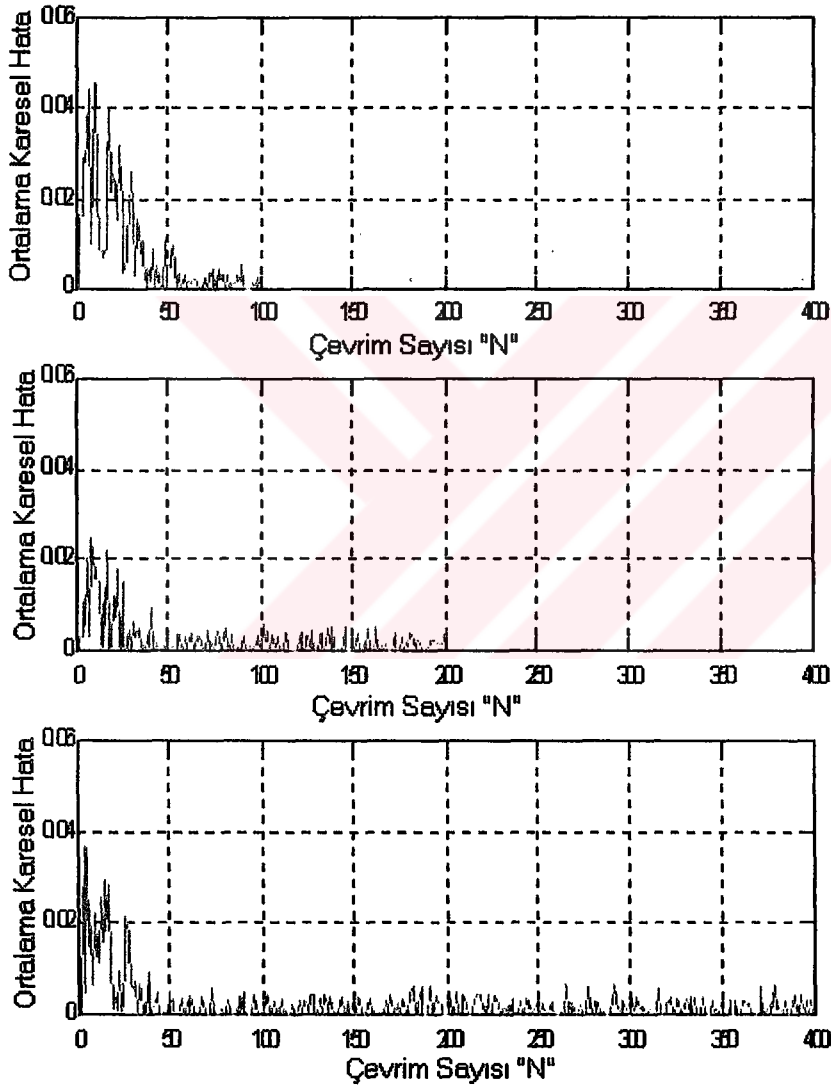


Şekil 4.2. Ortalama Karesel Hata'nın N'e göre Değişimi (Mimari =[7 10 1])

Tablo 4.4'de verilen sabit değerler için; Ortalama Karesel Hata Fonksiyonunun, Eğitim Sayısı N'ye göre değişim grafikleri Şekil 4.3 de verilmektedir.

N =100; 200; 400	Mimari	[7 10 5 1]
	η	0.1
	α	0.9

Tablo 4.4. Eğitim Çevrim Sayısına Göre Değişim, (Mimari = [7 10 5 1])



Şekil 4.3. Ortalama Karesel Hata'nın N'e göre Değişimi (Mimari =[7 10 5 1])

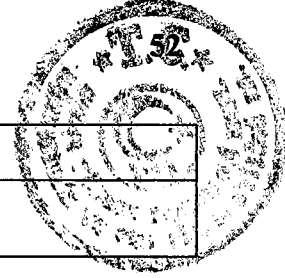
Şekil 4.1., Şekil 4.2. ve Şekil 4.3 'de görüldüğü üzere, N = 200 Eğitim Sayısı değeri için YSA eğitim aşamasında maksimum ortalama hata değeri; 0.005'den küçük

değerlere ulaşmaktadır. Bu, eğitim aşamasında 200 örnek verinin uygulanması sonucu; özellikle eğitimin son aşamalarında 5 / 1000 'den daha az bir ortalama karesel hata anlamına gelmektedir. Dolayısıyla, döviz değerlerinin tahmininde kullanılacak algoritmada, $N = 200$ değerinin kullanımı uygun bulunmaktadır.

4.1.2. Öğrenim Oranına Ait Parametreler

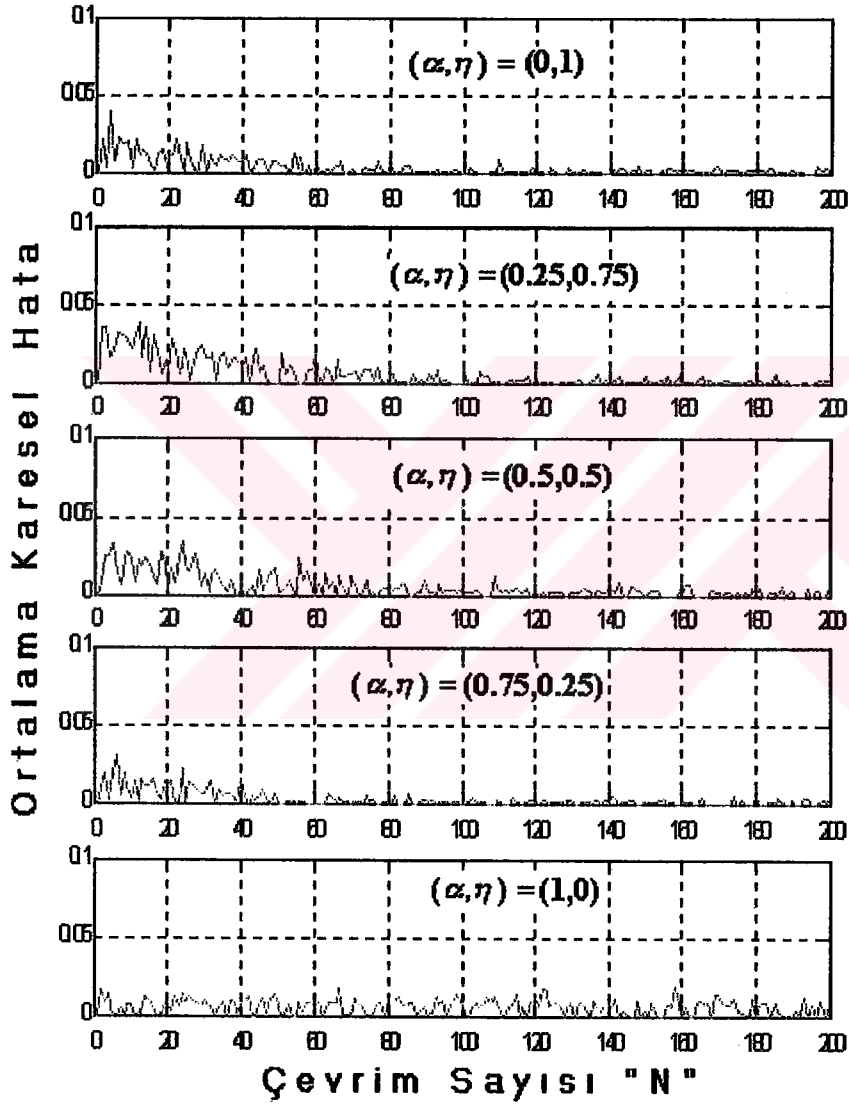
Bu kısımda, ağa ait diğer parametrelerin sabit değerleri için, Öğrenim Oranına ait parametrelerin (α, η) değişimine göre ortalama karesel hata fonksiyonunun değişimi incelenecektir.

Tablo 4.5'de verilen sabit değerler için; Ortalama Karesel Hata Fonksiyonunun, öğrenim oranına ait parametrelerin (α, η) değişimine göre değişim grafikleri Şekil 4.4 de verilmektedir.



$(\alpha, \eta) = (0,1); (0.25,0.75);$	Mimari	[7 1]
$(0.5,0.5); (0.75,0.25); (1,0)$	N	200

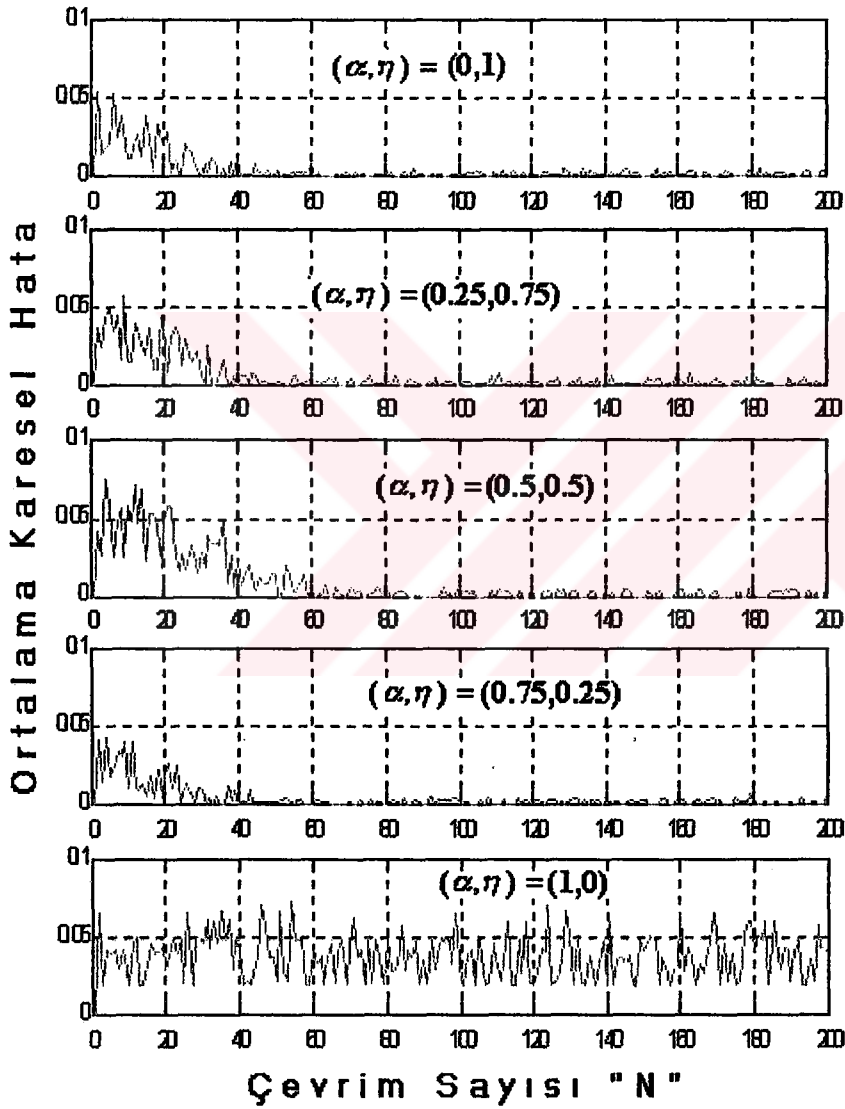
Tablo 4.5. (α, n) 'e Göre Değişim (Mimari = [7 1])



Şekil 4.4. Ortalama Karesel Hata'nın (α, n) 'e göre Değişimi (Mimari =[7 1])

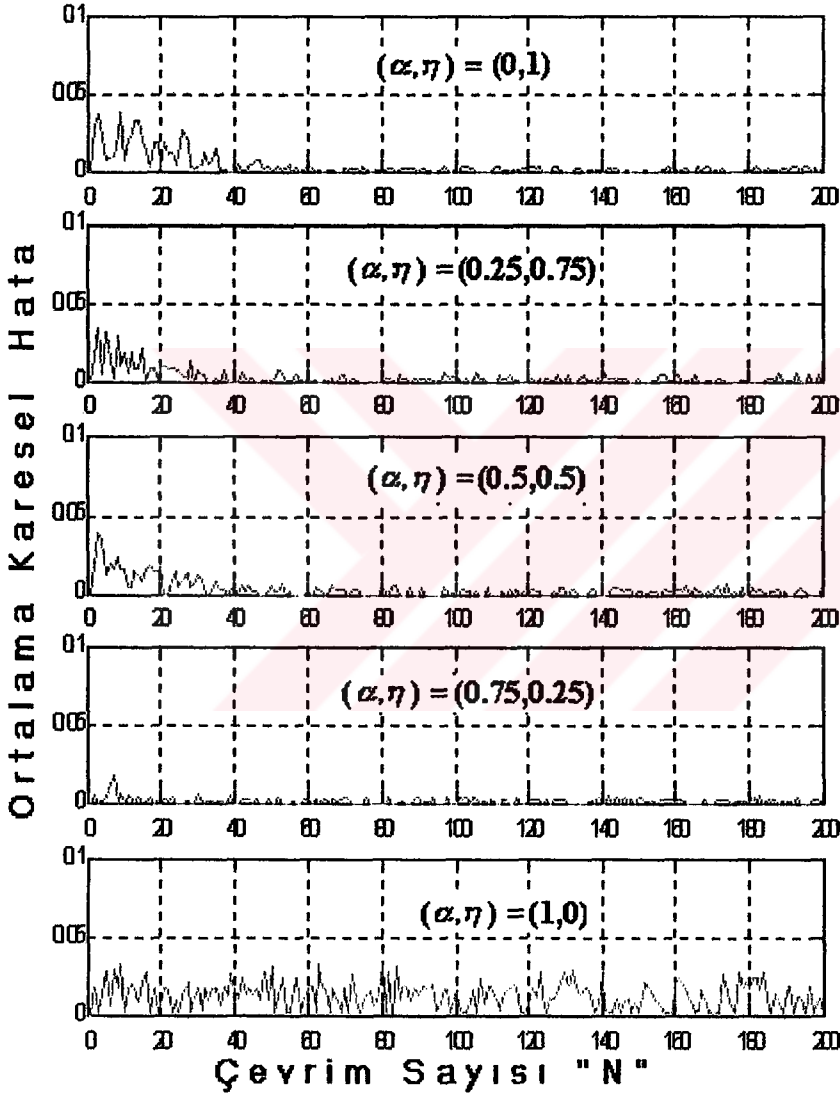
Tablo 4.6'da verilen sabit değerler için; Ortalama Karesel Hata Fonksiyonunun, öğrenim oranına ait parametrelerin (α, η) değişimine göre değişim grafikleri Şekil 4.5 de verilmektedir.

$(\alpha, \eta) = (0,1); (0.25,0.75);$	Mimari	[7 10 1]
$(0.5,0.5); (0.75,0.25); (1,0)$	N	200

Tablo 4.6. (α, η) 'e Göre Değişim (Mimari = [7 10 1])Şekil 4.5. Ortalama Karesel Hata'nın (α, η) 'e göre Değişimi (Mimari = [7 10 1])

Tablo 4.7'da verilen sabit değerler için; Ortalama Karesel Hata Fonksiyonunun, öğrenim oranına ait parametrelerin (α, η) değişimine göre değişim grafikleri Şekil 4.6 de verilmektedir.

$(\alpha, \eta) = (0,1); (0.25,0.75);$	Mimari	[7 10 5 1]
$(0.5,0.5); (0.75,0.25); (1,0)$	N	200

Tablo 4.7. (α, η) 'e Göre Değişim (Mimari = [7 10 5 1])Şekil 4.6. Ortalama Karesel Hata'nın (α, η) 'e göre Değişimi (Mimari = [7 10 5 1])

Şekil 4.4.; Şekil 4.5 ve Şekil 4.6 dan görüldüğü üzere; öğrenim oranı $\eta \rightarrow 0$ doğru azaltılıp; momentum sabiti $\alpha \rightarrow 1$ doğru yaklaştıkça; YSA çıkışında verilen ortalama karesel hata değerleri gittikçe daha çabuk yakınsamaktadır. Ancak; belirli bir aşamadan sonra, momentum sabiti α 'nın değerindeki artış, ve η 'nın değerindeki

azalış; YSA ortalama karesel hata değerin artmasına ve bir noktadan sonra salınım girmesine neden olmaktadır.

Ayrıca; ağ mimarisindeki katman sayısı arttıkça (gizli nöronların sayısı arttıkça) ; momentum sabiti α 'nın ancak daha büyük değerleri için ortalama karesel hata fonksiyonun daha çabuk yakınsadığı da gözlemlenmektedir.

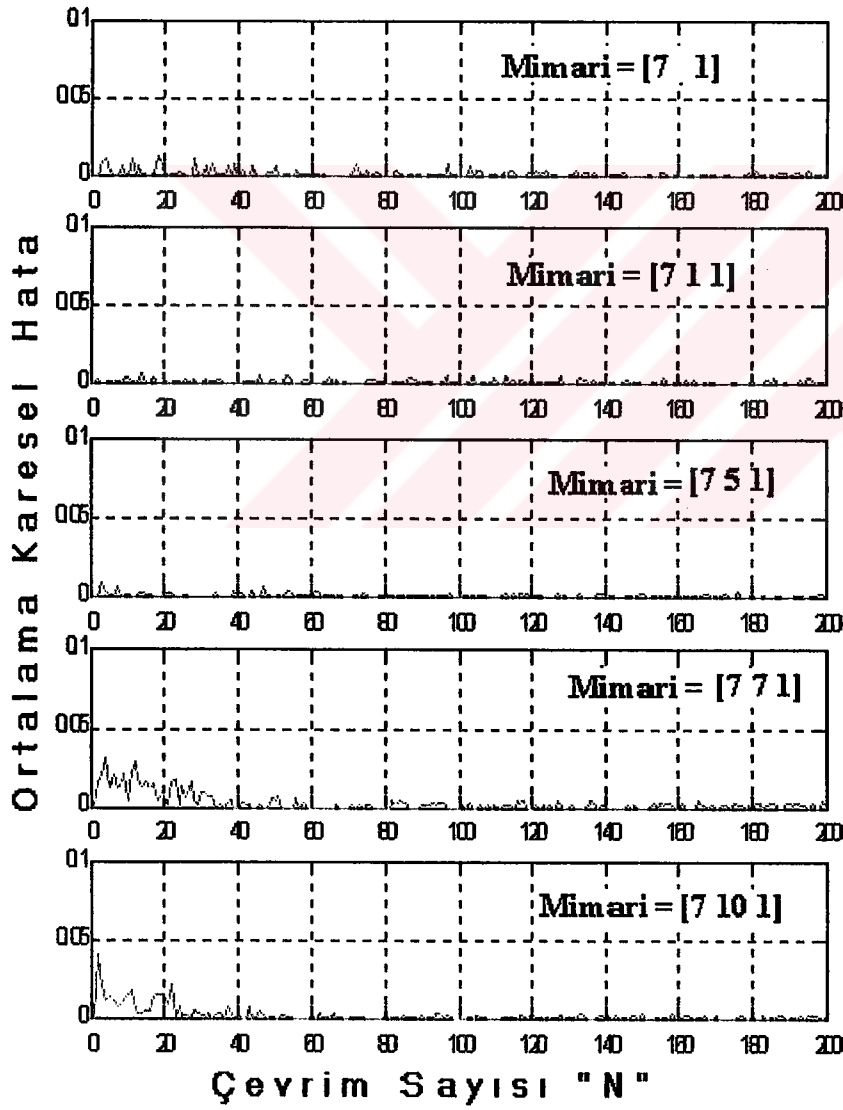
4.1.3. Ağ Mimarisi

Bu kısımda, ağa ait diğer parametrelerin sabit değerleri için, Ağ Mimarisinin değişimine göre ortalama karesel hata fonksiyonunun değişimi incelenecektir.

Tablo 4.8'de verilen sabit değerler için; Ortalama Karesel Hata Fonksiyonunun, Ağ Mimarisinin değişimine göre değişim grafikleri Şekil 4.7 de verilmektedir.

Mimari =	(α, η)	(0.25, 0.75)
[7 1]	N	200
[7 1 1]		
[7 5 1]		
[7 7 1]		
[7 10 1]		

Tablo 4.8. Mimariye Göre Değişim, 2 ve 3 Katmanlı Ağlar

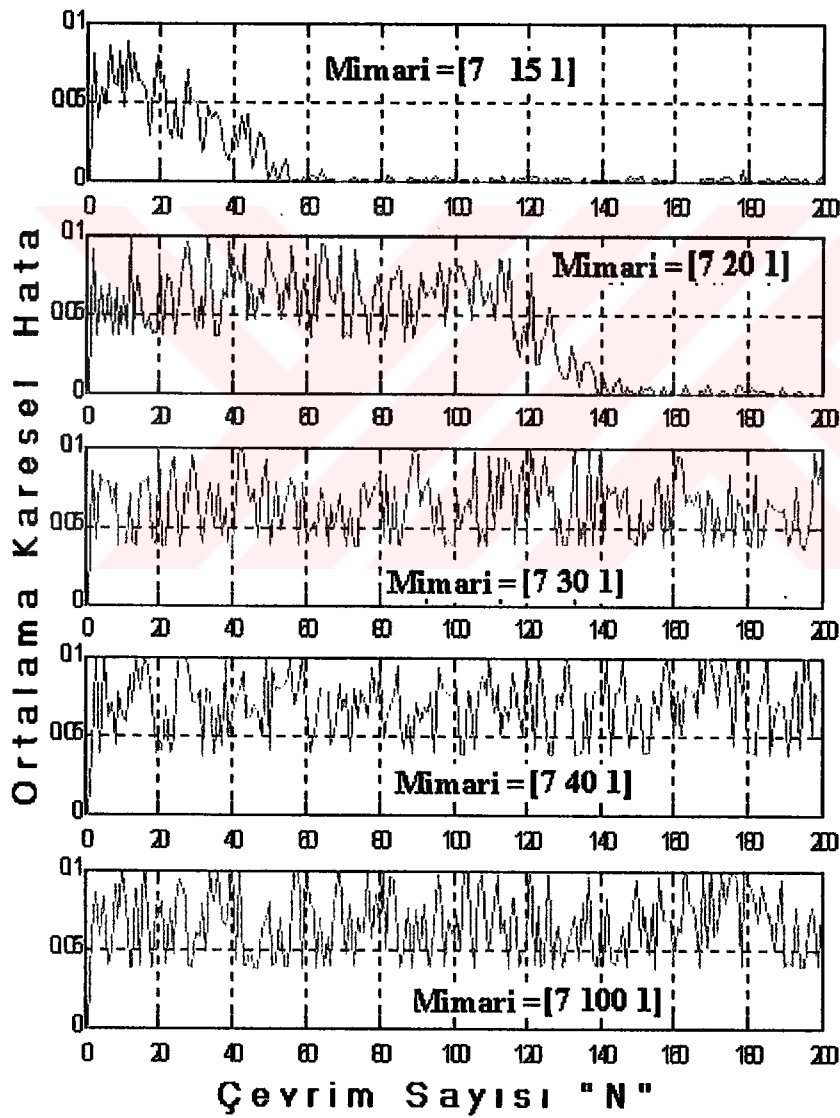


Şekil 4.7. Ortalama Karesel Hata'nın Mimariye Göre Değişimi: 2 ve 3 Katman İçin

Tablo 4.9'da verilen sabit değerler için; Ortalama Karesel Hata Fonksiyonunun, Ağ Mimarisinin değişimine göre değişim grafikleri Şekil 4.8 de verilmektedir.

Mimari =	(α, η)	$(0.25, 0.75)$
[7 15 1]	N	200
[7 20 1]		
[7 30 1]		
[7 40 1]		
[7 100 1]		

Tablo 4.9. Mimariye Göre Değişim, 3 Katmanlı Ağlar

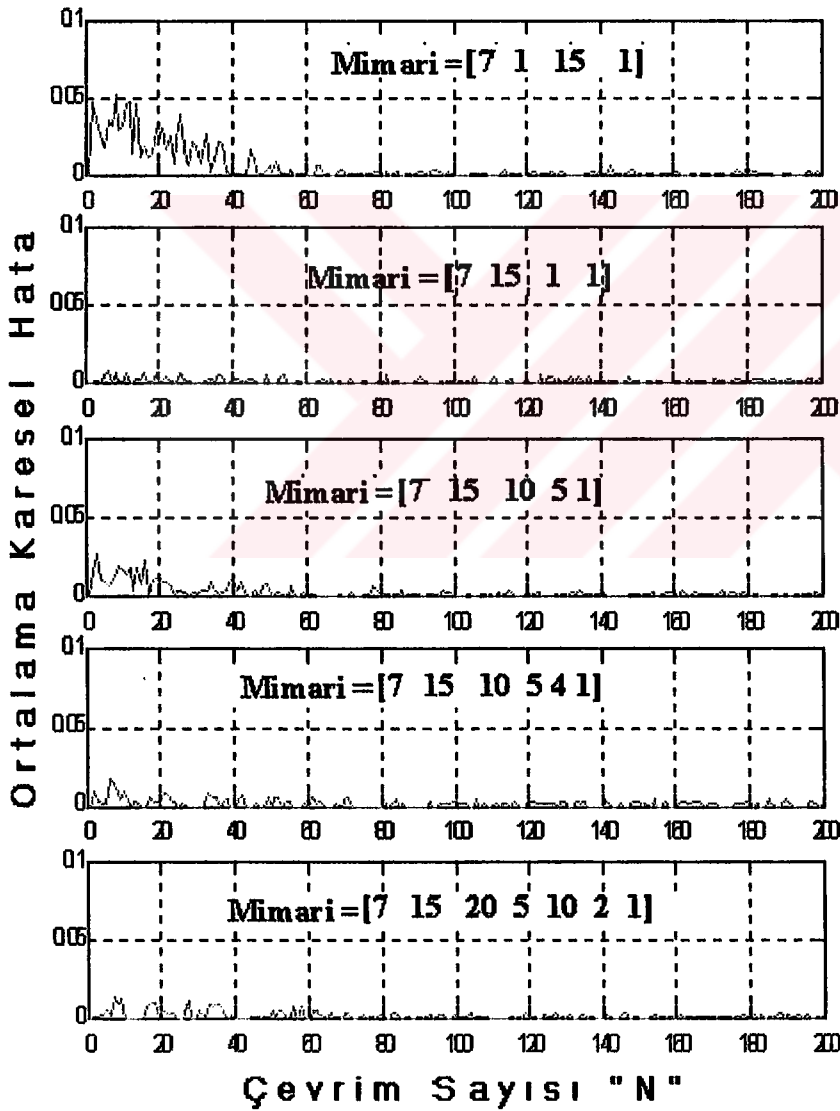


Şekil 4.8. Ortalama Karesel Hata'nın Mimariye Göre Değişimi: 3 Katman İçin

Tablo 4.10'da verilen sabit değerler için; Ortalama Karesel Hata Fonksiyonunun, Ağ Mimarisinin değişimine göre değişim grafikleri Şekil 4.9 da verilmektedir.

Mimari =	(α, η)	(0.25, 0.75)
[7 1 15 1]	N	200
[7 15 1 1]		
[7 15 10 5 1]		
[7 15 10 5 4 1]		
[7 15 20 5 10 2 1]		

Tablo 4.10. Mimariye Göre Değişim, 3'ten Fazla Katman



Şekil 4.9. Ortalama Karesel Hata'nın Mimariye Göre Değişimi: 3'den Fazla Katman

Şekil 4.7., Şekil 4.8., ve Şekil 4.9'da gözlemlendiği üzere; hesaplanan bütün mimari yapılar için ortalama karesel hata; 200 çevrim (iterasyon) boyunca 0.1 değerini

aşmamaktadır. Özellikle 100. çevrimden (iterasyon) sonraki ortalama karesel hata miktarı çoğu mimari için 0.005 değerinin altına düşmektedir. Döviz Kurlarının tahmini için kullanacağımız optimal mimariyi belirlemek için ortalama karesel hata fonksiyonunun son 100 çevrimde (101 den 200'e kadar olan çevrimler) aldığı değerlerin aritmetik ortalamasını denklem 4.1.'e göre hesaplırsak ortalama karesel hatanın son 100 çevrimde aldığı ortalama değerleri buluruz (Lütfen Tablo 4.11'e bakınız)

$$Ortalama = \left(\frac{1}{100} \right) \cdot \sum_{i=101}^{200} ortalama_hata(i) \quad (4.1)$$

Mimari	Son 100 çevrim için Ortalama Karesel Hata değerine ait ortalama:
[7 1]	0.0022
[7 1 1]	0.0016
[7 5 1]	0.0018
[7 7 1]	0.0015
[7 10 1]	0.0020
[7 15 1]	0.0014
[7 20 1]	0.0021
[7 30 1]	0.0654
[7 40 1]	0.0667
[7 100 1]	0.0661
[7 1 15 1]	0.0035
[7 15 1 1]	0.0016
[7 15 10 5 1]	0.0017
[7 15 10 5 4 1]	0.0017
[7 15 20 5 10 2 1]	0.0016

Tablo 4.11. Ortalama Değerler

Tablo 4.11 ile ilgili olarak dikkate edilmesi gereken husus; elde edilen değerlerin, YSA'nın ilk değer atamalarına bağımlı olduğudur (Lütfen Bölüm 3'e bakınız).

Tablo 4.11'de verilen sonuçlara göre; YSA mimarisi için optimal yapı olarak ikinci katmanda 15 nöronun kullanımı uygun bulunmaktadır (mimari = [7 15 1]).

4.2. Problemin Çözümünün Sayısal Olarak Gerçekleştirilmesi

Bu kısımda, Bölüm.4.1.'de elde edilen optimal parametrelerin kullanımıyla oluşturulacak Yapay Sinir Ağı; eğitim verileriyle eğitilecek, daha sonra test verilerinin eğitilmiş ağı uygulanmasıyla; Dolar, Mark, Japon Yeni, ve Avusturalya Doları için Yapay Sinir Ağının tahmin performansı değerlendirilecektir.

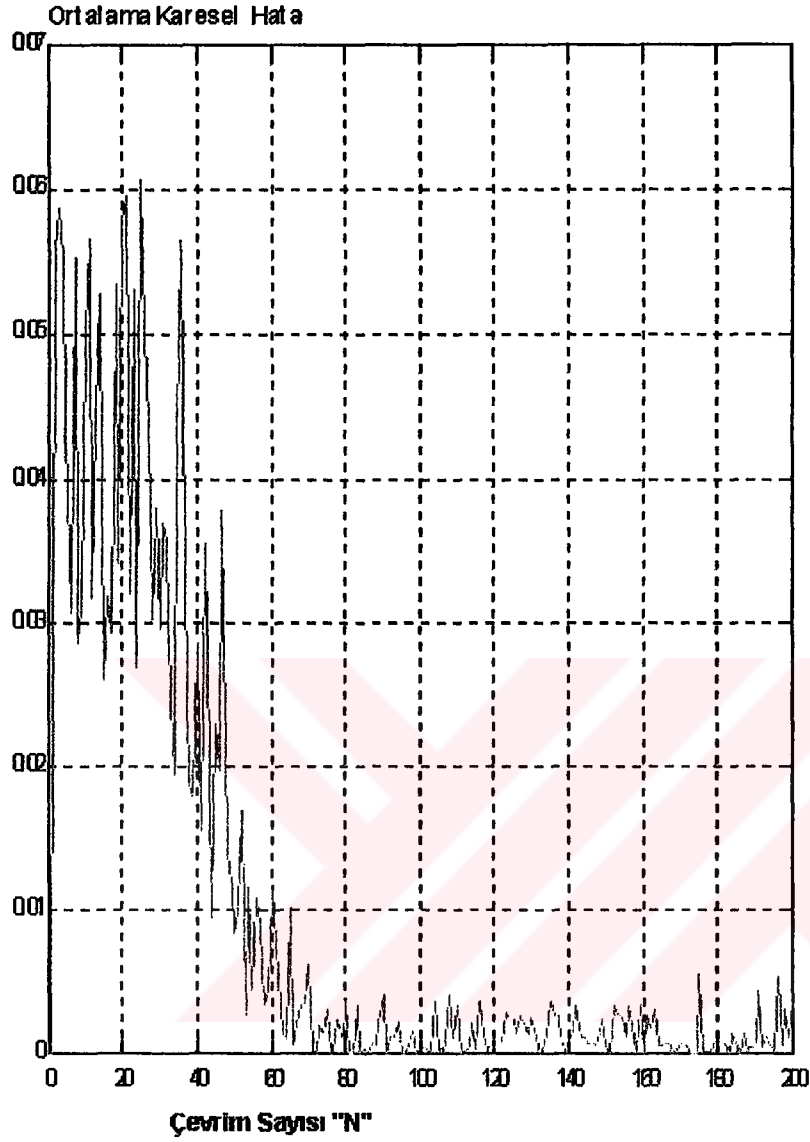
Tahmin için kullanılacak Yapay Sinir Ağına ait optimal parametreler, Tablo.4.11'de verilmektedir.

Parametre	Sembol	Optimal Değerler
YSA Mimarisi	Mimari	[7 15 1]
Öğrenim Oranı	η	0.75
Momentum Sabiti	α	0.25
Eğitim Çevrim Sayısı	N	200

Tablo 4.12. Optimal Parametreler

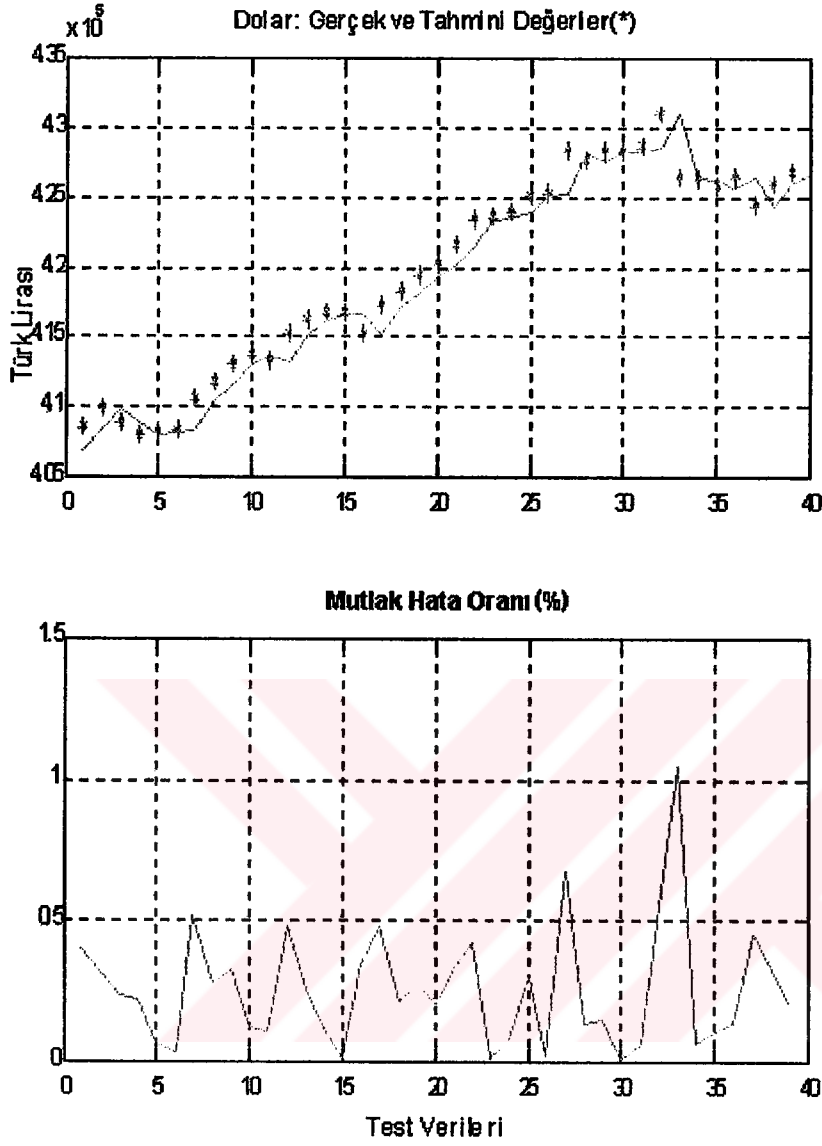
4.2.1. ABD Doları Alış Kuru için Yapay Sinir Ağıyla Tahmin

ABD Doları alış kuru için; Tablo.4.12'de verilen optimal değerlere göre oluşturulmuş Yapay Sinir Ağı'nın çıkış katmanında verilen "Eğitim Aşamasındaki" Ortalama Karesel Hata fonksiyonunun uygulanan çevrim sayısı N'ye göre değişimi Şekil 4.10'da verilmektedir.



Şekil 4.10. ABD Doları İçin Ortalama Karesel Hata

Şekil 4.11’de Tablo 4.12 deki optimal parametrelere göre oluşturulmuş Yapay Sinir Ağının “Test Verileri” için ürettiği tahmini değerler; ve tahmini değerlerle gerçek değerler arasındaki farkın mutlak değer olarak yüzdelik değişimleri verilmektedir. Görüldüğü üzere, tahmin edilen döviz kuru değeri ile gerçek döviz kuru değeri arasındaki fark maksimum %1 civarındadır.

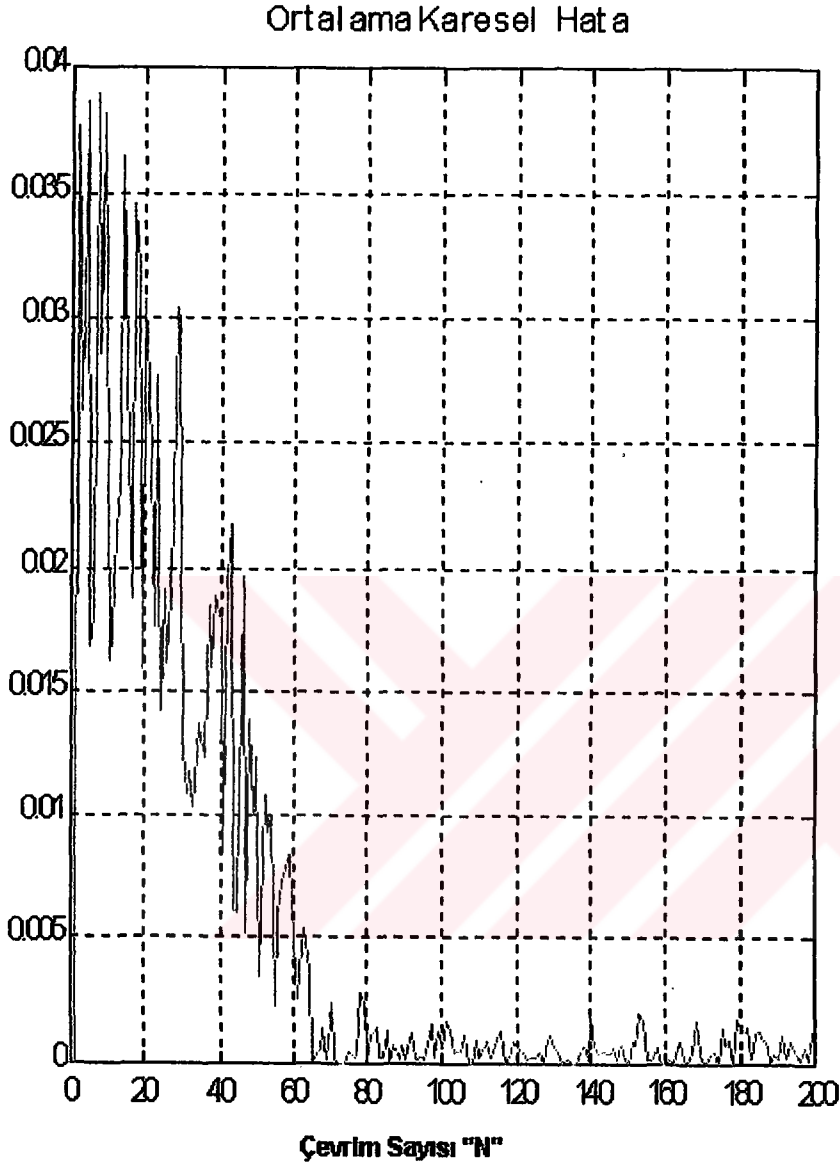


Şekil.4.11. ABD Doları İçin Test Sonuçları

4.2.2. Alman Markı Alış Kuru için Yapay Sinir Ağıyla Tahmin

Alman Markı alış kuru için; Tablo.4.12’de verilen optimal değerlere göre oluşturulmuş Yapay Sinir Ağı’nın çıkış katmanında verilen “Eğitim Aşamasındaki”

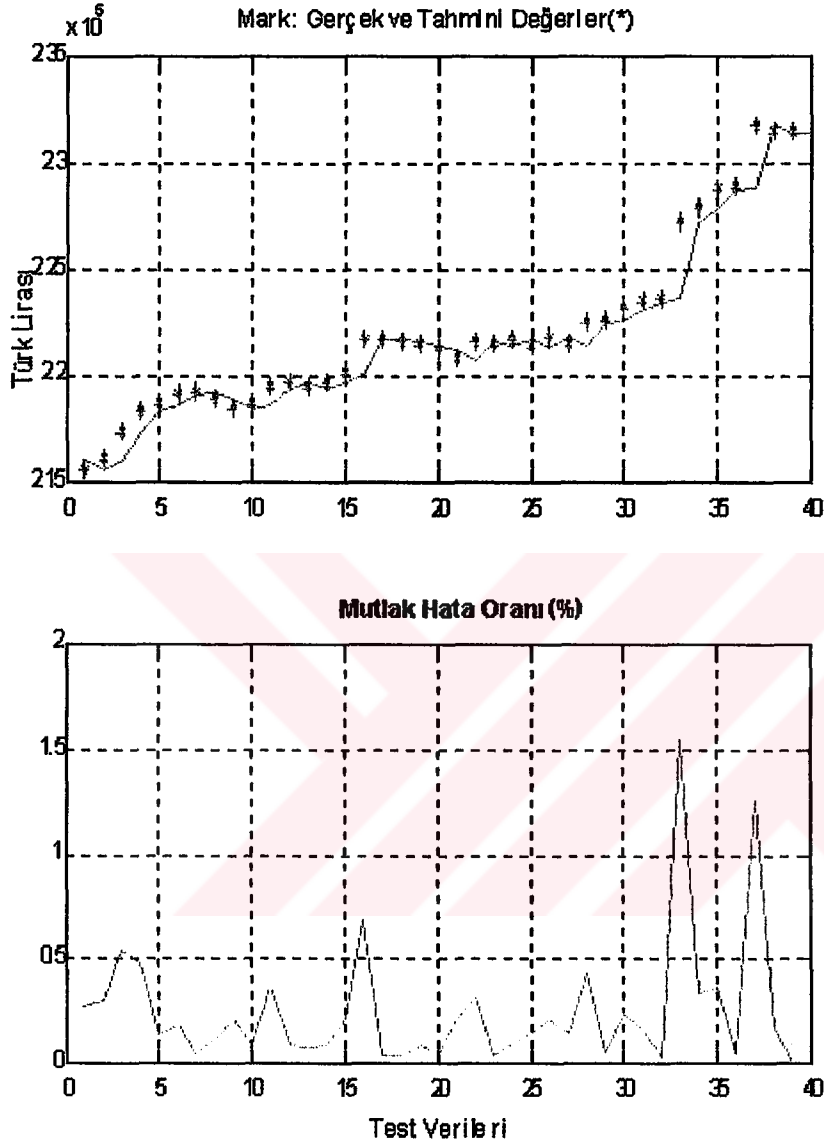
Ortalama Karesel Hata fonksiyonunun uygulanan eğitim örneği sayısı N 'ye göre değişimi Şekil 4.12'de verilmektedir.



Şekil 4.12. Alman Markı için Ortalama Karesel Hata

Şekil 4.13'de Tablo 4.12 deki optimal parametrelere göre oluşturulmuş Yapay Sinir Ağının "Test Verileri" için ürettiği tahmini değerler; ve tahmini değerlerle gerçek değerler arasındaki farkın mutlak değer olarak yüzdelik değişimleri verilmektedir.

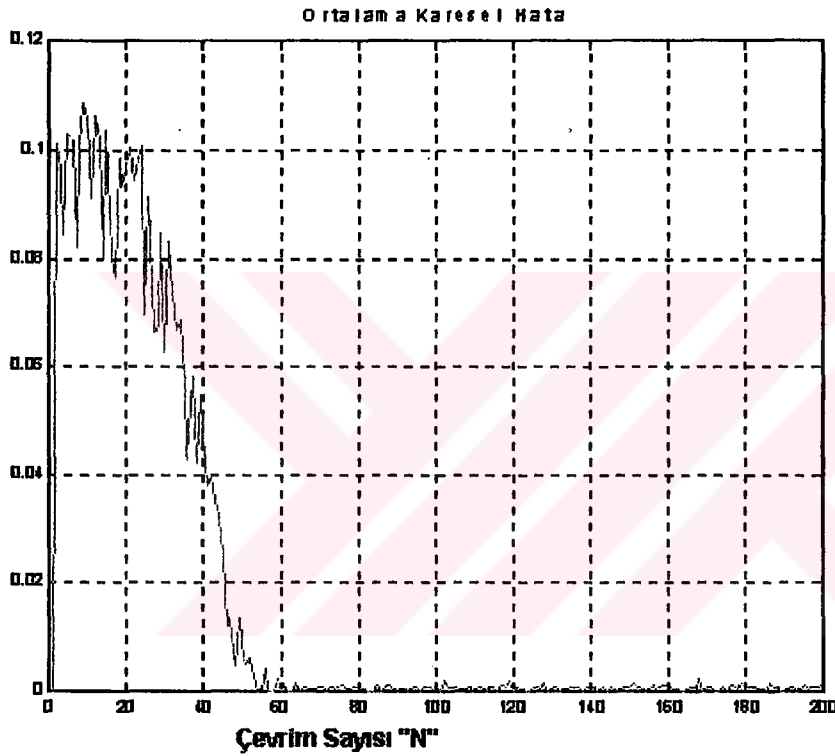
Görüldüğü üzere, tahmin edilen döviz kuru değeri ile gerçek döviz kuru değeri arasındaki fark maksimum %1.5 civarındadır.



Şekil 4.13. Alman Markı İçin Test Sonuçları

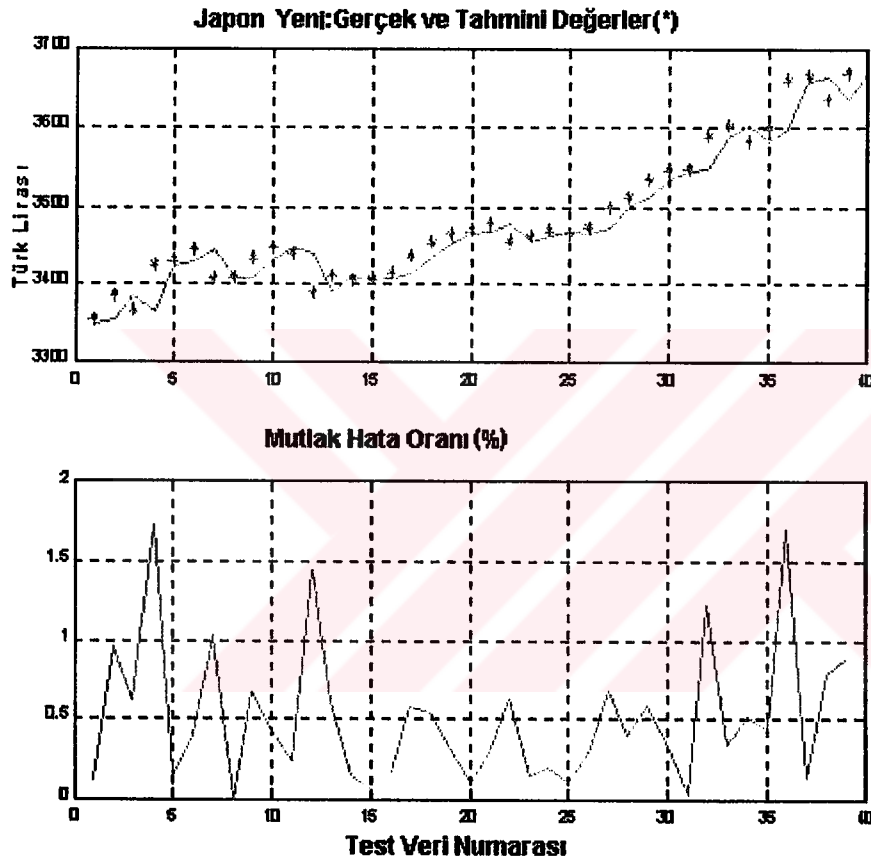
4.2.3. Japon Yeni Alış Kuru için Yapay Sinir Ağıyla Tahmin

Japon Yeni alış kuru için; Tablo.4.12’de verilen optimal değerlere göre oluşturulmuş Yapay Sinir Ağı’nın çıkış katmanında verilen “Eğitim Aşamasındaki” Ortalama Karesel Hata fonksiyonunun uygulanan çevrim sayısı N’ye göre değişimi Şekil 4.14’de verilmektedir.



Şekil 4.14. Japon Yeni İçin Ortalama Karesel Hata

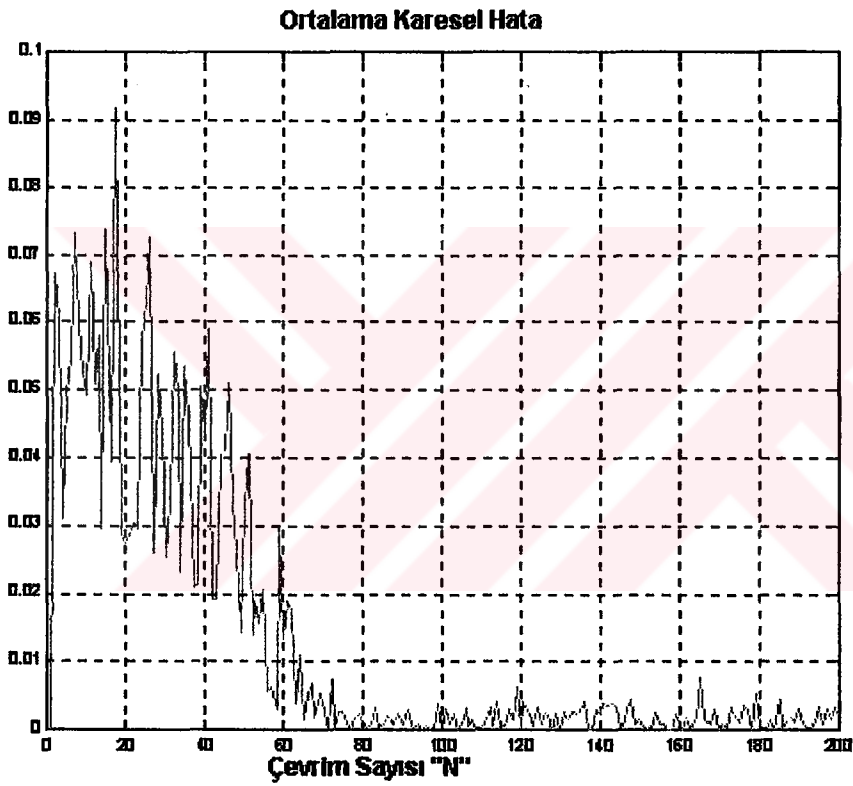
Şekil 4.15’de Tablo 4.12’ deki optimal parametrelere göre oluşturulmuş Yapay Sinir Ağının “Test Verileri” için ürettiği tahmini değerler; ve tahmini değerlerle gerçek değerler arasındaki farkın mutlak değer olarak yüzdelik değişimleri verilmektedir. Görüldüğü üzere, tahmin edilen döviz kuru değeri ile gerçek döviz kuru değeri arasındaki fark maksimum %1.5 civarındadır.



Şekil 4.15. Japon Yeni İçin Test Sonuçları

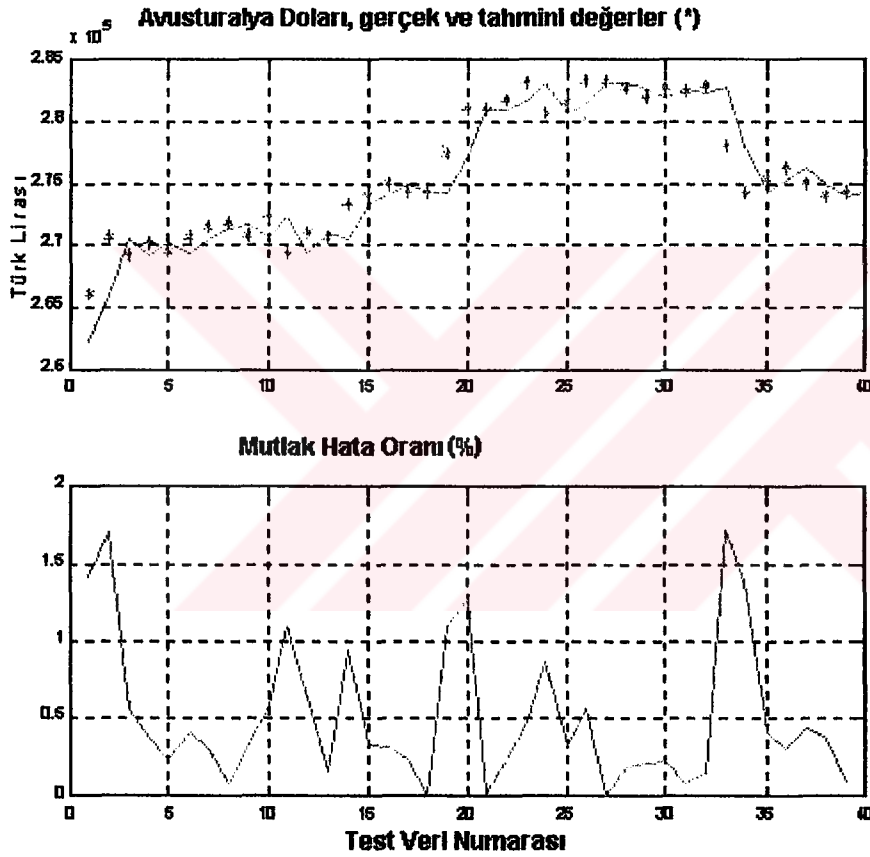
4.2.4. Avusturalya Doları Alış Kuru için Yapay Sinir Ağıyla Tahmin

Avusturalya Doları alış kuru için; Tablo.4.12’de verilen optimal değerlere göre oluşturulmuş Yapay Sinir Ağı’nın çıkış katmanında verilen “Eğitim Aşamasındaki” Ortalama Karesel Hata fonksiyonunun uygulanan çevrim sayısı N’ye göre değişimi Şekil 4.16’da verilmektedir.



Şekil 4.16. Avusturalya Doları İçin Ortalama Karesel Hata

Şekil 4.17’de Tablo 4.11’deki optimal parametrelere göre oluşturulmuş Yapay Sinir Ağının “Test Verileri” için ürettiği tahmini değerler; ve tahmini değerlerle gerçek değerler arasındaki farkın mutlak değeri olarak yüzdelik değişimleri verilmektedir. Görüldüğü üzere, tahmin edilen döviz kuru değeri ile gerçek döviz kuru değeri arasındaki fark maksimum %1.5 civarındadır.



Şekil 4.17. Avusturalya Doları İçin Test Sonuçları

5. SONUÇLAR

Yapay Sinir Ağıyla gerçekleştirilen tahmin işlemi aşamalarında; tahmini gerçekleştirilecek değer ve tahmin sürecinde kullanılan yardımcı veriler arasındaki ilişki için herhangi bir matematiksel model kullanılmamıştır. Yapay Sinir Ağı vasıtasıyla üretilen tahmin değerleri; ve bu değerlerin hesaplanmasıyla ilgili olarak kullanılan yardımcı veriler arasındaki ilişkilerin belirlenmesi için herhangi bir modelin kurulmak zorunda olmayışı; veriler arasındaki ilişkinin eğitim aşamasında otomatik olarak sinir ağı parametrelerinin ayarlanması / adapte edilmesi vasıtasıyla kendiliğinden oluşması; kullanılan tekniğin yapısı gereği sağladığı bir avantajdır.

ABD Doları, Alman Markı, Japon Yeni, ve Avusturalya Doları döviz kurlarının alış değerlerinin tahmini için kullandığımız; optimal yapay sinir ağı parametreleriyle kurulan Çok Katmanlı İleri Beslemeli Yapay Sinir Ağı mimarisi, ve eğitim algoritması olarak kullanılan Geri Yayılım algoritmasıyla elde edilen sonuçlar Bölüm 4.2’de sunulmaktadır.

Optimize edilmiş YSA’nın kullanımıyla; ABD Doları için Şekil 4.11; Alman Markı için Şekil 4.13; Japon Yeni için Şekil 4.15, ve Avusturalya Doları için Şekil 4.17’de grafiksel olarak sunulan sonuçlara göre; tahmin edilen döviz kuru alış değerleriyle; gerçek döviz kuru alış değerleri arasındaki fark %2’yi aşmamaktadır. Dolayısıyla, kullanılan tekniğin maksimum %2 hata ile tahmin değerleri ürettiği görülmektedir. .

KAYNAKLAR

- 1] S. Y. YUNG. Digital Neural Networks. -PTR Prentice Hall, 1993.
- 2] Simon HAYKIN. Neural Networks.-Prentice Hall, INC., Second Edition,1999
- 3] G. G. JUDGE, R. C. HILL, W. E. GRIFFITS, H. LÜTKEPOHL, T. C.LEE, Introduction to the Theory and Practice of Econometrics, -John Wiley & Sons, Second Edition, 1988.
- 4] P. MEHRA, B. W. WAH, Artificial Neural Networks, -IEEE Computer Society Press, 1992.
- 5] A. KILIÇBAY, Türk Ekonomisi, -Türkiye İş Bankası Kültür Yayınları, 3. Baskı, 1991.
- 6] N.K. BOSE, P. LIANG, Neural Network Fundamentals with Graphs, Algorithms, and Applications, -McGraw-Hill International Editions, 1996.
- 7] R. S. PINDIYCK, D. L. RUBINFELD, Econometric Models & Economic Forecasts, - McGraw – Hill International Editions, Third Edition, 1991
- 8] Robert A. HAUGEN, Modern Investment Theory, -Prentice Hall International, Inc., International Edition, 1997
- 9] P. A. SAMUELSON, W. D. NORDHAUS, Economics, -McGraw-Hill International Editions, Fourteenth Edition, 1992
- 10] T. ÖÇAL, Ö. F. ÇOLAK, S. TOGAY, K. ESER, Para Banka Teori ve Politika, -Gazi Kitabevi, 1. Baskı, 1997
- 11] J. S. R. JANG, C. T. SUN, E. MIZUTANI, Neuro-Fuzzy and Soft Computing, - Prentice Hall, Inc. 1997
- 12] Z. MICHALEWICZ, Genetic Algorithms + Data Structures = Evolution Programs, - Springer – Verlag Berlin Heidelberg New York, Third, Revised and Extended Version, 1996
- 13] P. H. WINSTON, Artificial Intelligence, - Addison – Wesley Publishing Company, Third Edition, 1992
- 14] V. RAO, H. RAO, C++ Neural Networks and Fuzzy Logic, MIS Press, Second Edition, 1995
- 15] D. C. MONTGOMERY, L. A. JOHNSON, J. S. GARDNER, Forecasting & Time Series Analysis,- MCGraw-Hill International Editions, Second Edition, 1990
- 16] V. F. SAVAŞ, İktisatın Tarihi, -Avcıol Basım Yayın, 2. Baskı, İstanbul 1998

ÖZGEÇMİŞ

Eyüp S. YILDIRIM, 1972 yılında Diyarbakır'da doğdu. İlk ve Orta öğrenimini tamamladıktan sonra Hacettepe Üniversitesi; Mühendislik Fakültesi, Elektrik Elektronik Mühendisliği Bölümü, Elektronik dalından mezun oldu. 1997 yılı bahar döneminde, Gebze Yüksek Teknoloji Enstitüsü, Elektronik Mühendisliği Bölümü'nde yüksek lisans eğitime ve aynı bölümde araştırma görevliliğine başladı. Gebze Yüksek Teknoloji Enstitüsü'nde asistanlık görevinin yanısıra Türkiye Bilimsel ve Teknik Araştırmalar Kurumu (Tübitak); Marmara Araştırma Merkezi, Uzay Teknolojileri Bölümü, Türk Ukrayna Ortak Laboratuvarı'nda (T.U.J.R.L: Turkish Ukrainian Joint Research Laboratory) yarı zamanlı araştırmacı olarak çalıştı. 1998 yılı güz dönemi sonundan itibaren akademik görevlerinden ayrılan Eyüp S. YILDIRIM, halen Ankara'da yerleşik özel bir firmada Sistem Mühendisi olarak çalışmaktadır.

EK'LER



EK 1:**UYGULAMA PROGRAMI KOD DÖKÜMLERİ**

```
%-----
% "EĞİTİM" PROGRAMI
%-----
%-----
```

```
veriler; %T(8,100) Eğitim değerleri
```

```
b= input('Katman Sayısını Giriniz: ');
```

```
mimari(1)=7; %-----
```

```
for i = 2:b-1;
```

```
    disp('Lütfen ');
```

```
    i
```

```
    mimari(i) = input(' . katmandaki nöron sayısını giriniz: ');
```

```
end
```

```
mimari(b)=1;
```

```
% mimari = Ağ mimari matrisi
```

```
r = max(mimari);
```

```
w = rand(r,r,b-1); %Ağırlık matrisine rastgele ilk değer atamaları
```

```
x = 0 .* rand(r,b); %nöron çıkış değerleri
```

```
bk = ones(r,b); %her bir nöron için besleme sinyalleri...
```

```
%-----
```

```
% Düzeltme oranı = delta_w
```

```
% Öğrenim oranı parametreleri: alfa : momentum sabiti, eta: öğrenim oranı
```

```
% N = %eğitim sayısı
```

```
a=0.5; %nöron parametresi;
```

```

alfa = input('Öğrenim oranına ait alfa momentum sabitini giriniz, alfa = ');
eta = input('Öğrenim oranına ait eta sabitini giriniz, eta = ');
N = input('Eğitim Sayısını Giriniz: ');
gradyan = 0 .* rand(r,b);
% her bir hücreye ait lokal gradyan değerlerinin kayıt edildiği matris.
hata = 0 .* rand(N,1);
% "hata" matrisi, çıkış hücresinde oluşan hata değerlerinin kayıt edildiği matris.

delta_w = w;
delta_w(:, :, N) = 0; %ağırlık matrisi düzeltme değerleri
delta_w = 0 .* delta_w;
%-----
for n = 2:N;
    %-----
    rastgele = round(100 * rand); %rastgele eğitim örneği seçimi
    while rastgele == 0;
        rastgele = round(100 * rand); %örnek numarasının 0 olmasını engelleyen kısım
    end
    %-----
    x(1:7,1) = T(1:7,rastgele);
    hedef = T(8,rastgele);
    %-----FORWARD PATH
    for k = 1:b-1;
        x(1:mimari(k+1),k+1) = w(1:mimari(k+1),1:mimari(k),k) * x(1:mimari(k),k);
        x(1:mimari(k+1),k+1) = x(1:mimari(k+1),k+1) + bk(1:mimari(k+1),k);
        x(1:mimari(k+1),k+1) = esy_neuron( x(1:mimari(k+1),k+1),a);

```

end

%-----BACKWARD PATH-----

% Çıkış hücresine ait ağırlık katsayılarına uygulanan düzeltme oranları:

for j = 1 : mimari(b-1);

z(j) = alfa * delta_w(1,j,b-1,n-1);

u(j) = (eta * (hedef - x(1,b))) * a * x(1,b) * (1 - x(1,b)) .* x(j,b-1);

delta_w(1,j,b-1,n) = z(j) + u(j);

end

%-----

hata(n) = hedef - x(1,b); % Çıkış hücresinde oluşan hata değeri

rpt = 0;

for i = 1:n;

rpt = rpt + 0.5 .* hata(n).^2;

end

o_hata(n) = rpt / n;

%-----Gizli katmandaki nöronlara ait ağırlık katsayılarına uygulanan düzeltme oranları

gradyan(1,b) = (hedef - x(1,b)) * a * x(1,b) * (1 - x(1,b));

%-----

for k = b-1:-1:1;

gradyan(1:mimari(k),k) = w(1:mimari(k),1:mimari(k+1),k) *
gradyan(1:mimari(k+1),k+1);

end

%-----

for k = 1:b-2;

```

for j = 1:mimari(k+1);
    for i = 1:mimari(k);
        delta_w(j,i,k,n) = eta * a * x(j,k+1) * (1 - x(j,k+1)) * gradyan(j,k+1) * x(i,k);
        delta_w(j,i,k,n) = alfa * delta_w(j,i,k,n-1) + delta_w(j,i,k,n);
    end
end

end

%-----
for k = 1:b-1;
    for i=1:r;
        for j = 1:r;
            w(i,j,k) = delta_w(i,j,k,n) + w(i,j,k);
        end
    end
end

end

end

plot(o_hata);grid;
xlabel('Eğitim Sayısı');ylabel('Ortalama Karesel Hata');
%-----
%-----
%-----

```

```

%-----

% TEST PROGRAMI
%-----

mimari;

% mimari = Ağ mimari matrisi

r = max(mimari);

w = rand(r,r,b-1); %Ağırlık matrisine rastgele ilk değer
atamaları

x = 0 .* rand(r,b); %nöron çıkış değerleri

bk = ones(r,b); %her bir nöron için besleme sinyalleri...

hata = 0 .* rand(40,1);

% "hata" matrisi, çıkış hücresinde oluşan hata
değerlerinin kayıt edildiği matris.

tahmin = 0 .*rand(40,1);

%-----
%-----

for n = 2:40;

    x(1:7,n) = G(1:7,n);

    hedef = G(8,n);

    %-----FORWARD PATH

    for k = 1:b-1;

        x(1:mimari(k+1),k+1) =
w(1:mimari(k+1),1:mimari(k),k) * x(1:mimari(k),k);

        x(1:mimari(k+1),k+1) = x(1:mimari(k+1),k+1) +
bk(1:mimari(k+1),k);

        x(1:mimari(k+1),k+1) = esy_neuron(
x(1:mimari(k+1),k+1),a);

    end

```

```

    hata(n) = hedef - x(1,b); % Çıkış hücresinde oluşan
hata değeri

    tahmin(n) = hata(n) + x(1,b);
    rpt = 0;

    for i = 1:n;

        rpt = rpt + 0.5 .* hata(n).^2;
    end

    o_hata(n) = rpt / n;

end
%-----
for m = 2:40;

    hatalar(m) = 100 * abs( (tahmin(m) - dolar_n(m+100)) /
tahmin(m));
end
%-----

subplot(2,1,1),plot(max(dolar) *
transpose(dolar_n(102:141)));grid on;hold on;

subplot(2,1,1),plot(max(dolar) * tahmin(2:40),'*');grid
on;

ylabel('T ü r k   L i r a s ı');

title('D o l a r :   G e r ç e k   v e   T a h m i n i   D e
ğ e r l e r ( * ) ');

subplot(2,1,2),plot(hatalar(2:40));grid on;

xlabel('T e s t   V e r i l e r i');

title('M u t l a k   H a t a   O r a n ı ;      % ');

%-----
%-----
%-----

```

```
%-----  
%-- NÖRON fonksiyonu ---  
%-----  
%-----
```

```
function h = esy_neuron(x,a);
```

```
v = x; %nörona gelen sinyallerin toplamı
```

```
h = 1 ./ (1 + exp(-a .* v));
```

```
%-----  
%-----  
%-----
```

```
%-----  
%-- Veriler Programı ---  
%-----  
%-----
```

```
clear;  
  
load altin.m;  
  
load dolar.m;  
  
load yen.m;  
  
load avdolar.m;  
  
load imkb.m;  
  
load mark.m;  
  
load repo.m;  
  
%-----  
  
altin_n = altin / max(altin);  
dolar_n = dolar / max(dolar);  
yen_n = yen / max(yen);  
  
avdolar_n = avdolar / max(avdolar);  
  
imkb_n = imkb / max(imkb);  
mark_n = mark / max(mark);  
repo_n = repo / max(repo);  
  
%-----  
  
T = 0 .* rand(8,100);  
  
T(1,1:100) = transpose(dolar_n(1:100));  
T(2,1:100) = transpose(altin_n(1:100));  
T(3,1:100) = transpose(yen_n(1:100));  
T(4,1:100) = transpose(mark_n(1:100));  
T(5,1:100) = transpose(imkb_n(1:100));
```



```
T(6,1:100) = transpose(avdolar_n(1:100));
```

```
T(7,1:100) = transpose(repo_n(1:100));
```

```
T(8,1:100) = transpose(dolar_n(2:101));
```

```
%-----
```

```
%Test Verileri
```

```
G = 0 .* rand(8,40);
```

```
G(1,1:40) = transpose(dolar_n(101:140));
```

```
G(2,1:40) = transpose(altin_n(101:140));
```

```
G(3,1:40) = transpose(yen_n(101:140));
```

```
G(4,1:40) = transpose(mark_n(101:140));
```

```
G(5,1:40) = transpose(imkb_n(101:140));
```

```
G(6,1:40) = transpose(avdolar_n(101:140));
```

```
G(7,1:40) = transpose(repo_n(101:140));
```

```
G(8,1:40) = transpose(dolar_n(102:141));
```

```
%-----
```

```
%-----
```

```
%-----
```