

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**SINIFLANDIRMADA YAPAY SİNİR AĞLARI VE KARAR AĞAÇLARI
KARŞILAŞTIRMASI: ÖĞRENCİ BAŞARILARI ÜZERİNE BİR UYGULAMA**

**YÜKSEK LİSANS TEZİ
Süleyman TOSUN**

Anabilim Dalı : ENDÜSTRİ MÜHENDİSLİĞİ

Programı : MÜHENDİSLİK YÖNETİMİ

HAZİRAN 2007

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**SINIFLANDIRMADA YAPAY SİNİR AĞLARI VE KARAR AĞAÇLARI
KARŞILAŞTIRMASI: ÖĞRENCİ BAŞARILARI ÜZERİNE BİR UYGULAMA**

**YÜKSEK LİSANS TEZİ
Süleyman TOSUN
(507041224)**

**Tezin Enstitüye Verildiği Tarih : 7 Mayıs 2007
Tezin Savunulduğu Tarih : 11 Haziran 2007**

**Tez Danışmanı : Yrd. Doç.Dr. Mehmet Mutlu YENİSEY
Diğer Jüri Üyeleri Öğr. Gör. Dr. Gülgün KAYAKUTLU (İ.T.Ü.)
Yrd. Doç Dr. Şule ÖNSEL (D.Ü.)**

HAZİRAN 2007

ÖNSÖZ

Teknolojinin gelişmesi ve insanlığın her alanda bir adım daha ilerlemesi birbirini tetikleyici iki unsur haline geldi. İnsanların bu gelişimde en etkili araç olarak kullandıkları bilgisayar ile insan yapısını birleştirmek istemesi de yapay sinir ağlarının temelini oluşturdu. Yapay sinir ağları günümüzde bir çok alanda kullanılıyor ve her geçen gün hayatımıza daha fazla girmeye devam ediyor.

Geleceğin teknoloji çağının temellerinin atıldığı yapay zeka ve yapay sinir ağları yolundaki bu çalışmalarda insanlık bir devrim yaratmaya devam ediyor. Doğrusal programlama yapısını beyin modelleri ve sinir ağlarını inceleyerek geliştiren ve programlamayı doğrusallıktan kurtaran bu bilim dalı ile birlikte tüm verilerin karmaşasına ve veri sayısının çok fazla olmasına aldırmadan hareket edebilir ve sonuca ulaşabilirsiniz.

Gelecekte bir çok alanda hayatımızda yer alacak yapay sinir ağları uygulamasını akademik olarak ele almak istememin temel nedeni de bu gelişime ve geleceğe olan merakım ile şekillendi. Çalışmanın temelinde hem yapay sinir ağları uygulaması gerçekleştirmek hem de uygulamayı farklı metotlarla karşılaştırma fikri yer alıyor. Bu sebeple üniversite öğrencilerinin başarılarını etkileyen faktörlerin sınıflandırılması ile ilgili bir karar ağaçları uygulamasını temel alınarak ilerlenildi. Çalışma içerisinde yapay sinir ağları hakkında teorik bilgiler aktardıktan sonra daha önce gerçekleştirilmiş olan karar ağaçları çalışması hakkında bilgilere yer verilmektedir. Tüm bu bilgiler sonrasında yapay sinir ağları uygulamasının nasıl yapıldığı ve sonuçları yer almaktadır. Son bölümde ise her iki uygulamanın sonuçlarına yer verilerek yapay sinir ağlarının başarısı ortaya konulmuştur.

Çalışmam esnasında rehberliği ve yardımları ile yön gösteren Sayın Yrd. Doç. Dr. Mehmet Mutlu YENİSEY'e, manevi desteği ile her zaman yanımda olan aileme ve bitanem Purnur FIRAT'a teşekkür ederim.

Haziran 2007

Süleyman TOSUN

İÇİNDEKİLER

KISALTMALAR	v
TABLO LİSTESİ	vi
ŞEKİL LİSTESİ	viii
SEMBOL LİSTESİ	x
ÖZET	xi
SUMMARY	xii
1. YAPAY SINIR AĞLARI	1
1.1. Yapay Sinir Ağları	1
1.2. Biyolojik Sinir Sistemleri	2
1.3. Yapay Sinir Ağlarının Tarihsel Gelişimi	4
1.4. Genel Olarak Yapay Sinir Ağları Tanımı	6
1.5. Yapay Sinir Ağlarında Genel Yapı	7
1.6. Yapay Sinir Ağlarının Özellikleri	8
1.7. Yapay Sinir Ağlarının Uygulama Alanları	9
2. HÜCRE VE AĞ YAPISI	11
2.1. Statik Hücre Modeli	11
2.2 Hareket Fonksiyonları	12
2.2.1. Doğrusal Hareket Fonksiyonu	13
2.2.2. Parçalı Doğrusal Fonksiyon	13
2.2.3. Sigmoid Hareket Fonksiyonu	14
2.2.4. Eşik Hareket Fonksiyonu	15
2.2.5. Hiperbolik Tanjant Fonksiyonu	15
2.3. Dinamik Hücre Modelleri	16
2.3.1 FIR Filtre Ağırlıklı Dinamik Hücre Modeli	16
2.3.2. Birinci Dereceden Bir Filtreli Dinamik Hücre Modeli	17
2.4. Yapay Sinir Ağ Modelleri	18
2.4.1. İleri Beslemeli Yapay Sinir Ağları	18
2.4.2. Geri Beslemeli Yapay Sinir Ağları	19
2.4.3. Radyal Tabanlı Fonksiyon Ağları (RTFA)	20
2.4.4. Çağrışimli Bellek Ağları	21
2.5. Diğer Yapay Sinir Ağları	22
2.5.1. Vektör Kuantalama	22
2.5.2. Hopfield Ağı	24
2.5.3. Elman ve Jordan Ağları	25
2.5.4. Kohonen Ağı	27

3. ÖĞRENME VE ÖĞRENME KURALLARI	28
3.1 Yapay Sinir Ağlarında Öğrenme	28
3.1.1. Gözetimli Öğrenme Yöntemi	29
3.1.2. Gözetimsiz Öğrenme Yöntemi	30
3.1.3. Karma Öğrenme	30
3.2. Öğrenme Parametreleri	31
3.3. Öğrenme Kuralları	32
3.3.1. Genel Eğitici Öğrenme Kuralı	33
3.3.2. Hebb Öğrenme Kuralı	33
3.3.3. Delta Öğrenme Kuralı	34
3.3.4. Widrow-Hoff Öğrenme Kuralı	34
3.3.5. Algılayıcı Öğrenme Kuralı	35
3.4. Geriye Yayılmalı Öğrenme Algoritması	35
3.5. Delta Öğrenme Kuralı	40
3.6. Genelleştirilmiş Delta Öğrenme Kuralı	41
3.7. Eğitim Zamanını Azaltan Yöntemler	42
3.7.1 Adaptif Öğrenme Oranı	42
3.7.2. Daha İyi İlk Ağırlık Seçme Yöntemi	43
3.7.3. Momentum Metodu	43
4. KARAR AĞAÇLARI UYGULAMASI SONUÇLARI	45
4.1. Üniversite Öğrencilerinin Başarılarını Etkileyen Faktörlerin Sınıflandırılması	45
4.1.1. Demografik Özellikler:	46
4.1.2. Eğitim Soruları (Üniversite Öncesi)	55
4.1.3. Akademik Sorular (Üniversite Öğrenimi Sırasında)	57
4.1.4 Kişisel Sorular	60
4.2. Karar Ağaçları Uygulaması	66
5. YAPAY SİNİR AĞLARI UYGULAMASINDA KULLANILAN YAZILIMLAR	69
5.1 Yapay Sinir Ağları Uygulamasında Grafikselsel Arayüz Kullanımı	71
5.2 NeuroSolutions	80
6. YAPAY SİNİR AĞLARI UYGULAMASI	86
7. HER İKİ METODUN KARŞILAŞTIRILMASI	88
8. SONUÇ ve ÖNERİLER	93
KAYNAKLAR	95
EKLER	98
ÖZGEÇMİŞ	115

KISALTMALAR

YSA : Yapay Sinir Ağları
RTFA : Radyal Tabanlı Fonksiyon Ağları

TABLO LİSTESİ

	<u>Sayfa No</u>
Tablo 1.1. Sinir Sistemi ile Yapay Sinir Ağlarının Benzerlikleri	4
Tablo 4.1. Yaş Dağılımı	46
Tablo 4.2. Yaş İstatistikleri Özeti	46
Tablo 4.3. Cinsiyet Dağılımı	46
Tablo 4.4. Medeni Durum Dağılımı	47
Tablo 4.5. Çocuk Sahipliği Dağılımı	47
Tablo 4.6. Baba Hayatta Dağılımı	47
Tablo 4.7. Doğum Bölgesi Dağılımı	48
Tablo 4.8. En Uzun Süre Yaşanılan Yer/Bölge Dağılımı	48
Tablo 4.9. Lise Öğrenimi Dağılımı	49
Tablo 4.10. Anne Eğitim Düzeyi Dağılımı	49
Tablo 4.11. Baba Eğitim Düzeyi Dağılımı	50
Tablo 4.12. Büyük Erkek Kardeş Sayısı Dağılımı	50
Tablo 4.13. Büyük Erkek Kardeş Sahipliği Dağılımı	50
Tablo 4.14. Büyük Kız Kardeş Sayısı Dağılımı	50
Tablo 4.15. Büyük Kız Kardeş Sahipliği Dağılımı	51
Tablo 4.16. Küçük Erkek Kardeş Sayısı Dağılımı	51
Tablo 4.17. Küçük Erkek Kardeş Sahipliği Dağılımı	51
Tablo 4.18. Küçük Kız Kardeş Sayısı Dağılımı	51
Tablo 4.19. Küçük Kız Kardeş Sahipliği Dağılımı	52
Tablo 4.20. Toplam Erkek Kardeş Sayısı Dağılımı	52
Tablo 4.21. Erkek Kardeş Sahipliği Dağılımı	52
Tablo 4.22. Toplam Kız Kardeş Sayısı Dağılımı	52
Tablo 4.23. Kız Kardeş Sahipliği Dağılımı	52
Tablo 4.24. Kardeş Sahibi Olma Dağılımı	53
Tablo 4.25. Değiştirilmiş Toplam Kardeş Sayısı Dağılımı	53
Tablo 4.26. Türk Olma/Olmama Dağılımı	53
Tablo 4.27. Çalışan Anne Dağılımı	54
Tablo 4.28. Çalışan Anne Dağılımı -2	54
Tablo 4.29. Aile Durumu Dağılımı	54
Tablo 4.30. Lise Tipi Dağılımı	55
Tablo 4.31. Lise Not İstatistikleri	55
Tablo 4.32. Üniversiteye İlk Sınavda Giriş Dağılımı	55
Tablo 4.33. Bölüm Tercih Sırası Dağılımı	56
Tablo 4.34. Bölüm Tercih Sırası Dağılımı – 2	56
Tablo 4.35. Kütüphane Kullanımı Dağılımı	57
Tablo 4.36. Akademik Olmayan Yayın Okuma Dağılımı	57
Tablo 4.37. Akademik Yayın Okuma Dağılımı	58
Tablo 4.38. Konferans Katılımı Dağılımı	58
Tablo 4.39. Yüksek Lisans Kararı Dağılımı	58

Tablo 4.40.	Not Ortalaması Dağılımı	59
Tablo 4.41.	Okul Projeleri Katkısı Dağılımı	59
Tablo 4.42.	Devamlılık Yüzdesi Dağılımı	59
Tablo 4.43.	Yarı Zamanlı Çalışma Dağılımı	60
Tablo 4.44.	Sinemaya Gitme Sıklığı Dağılımı	60
Tablo 4.45.	Spor Yapma Dağılımı	60
Tablo 4.46.	Hobi/kişisel Gelişim Kursu Dağılımı	61
Tablo 4.47.	Günlük İnternet Kullanım Saati Dağılımı	61
Tablo 4.48.	Ortalama Televizyon İzleme Süresi Dağılımı	61
Tablo 4.49.	Ortalama Çalışma Süresi Dağılımı	62
Tablo 4.50.	Üniversitedeki Arkadaşlarla Görüşme Dağılımı	62
Tablo 4.51.	Kulüplere Katılım Dağılımı	62
Tablo 4.52.	İstanbul'da İkametgah Yeri Dağılımı	63
Tablo 4.53.	Okula Ulaşım Şekli Dağılımı	63
Tablo 4.54.	Ortalama Ulaşım Süresi Dağılımı	63
Tablo 4.55.	Çalışma Çevresi Memnuniyet Dağılımı	63
Tablo 4.56.	Kişisel Bilgisayar Dağılımı	64
Tablo 4.57.	Aylık Ortalama Gelir Dağılımı	64
Tablo 4.58.	Bursluluk Dağılımı	64
Tablo 4.59.	Öğrenim Kredisi Kullanma Dağılımı	65
Tablo 4.60.	İş Bulabilme Dağılımı	65
Tablo 4.61.	İş Bulma Süresi Dağılımı	65
Tablo 4.62.	İ.T.Ü. Sosyal Çevre Görüş Dağılımı	65
Tablo 4.63.	En Çok Sevilen Ders Grubu Dağılımı	66
Tablo 4.64.	Memnuniyet Dağılımı	66
Tablo 4.65.	Çalışma Verileri	67
Tablo 4.66.	Başarıyı Etkileyen Değişkenler ve Önem Dereceleri	68
Tablo 6.1.	Uygulama Değerleri	86
Tablo 6.2.	Başarıyı Etkileyen Değişkenler ve Önem Dereceleri	87
Tablo 7.1.	Başarıyı Etkileyen Değişkenler ve Önem Dereceleri (Karar Ağaçları)	89
Tablo 7.2.	Başarıyı Etkileyen Değişkenler ve Önem Dereceleri (Yapay Sinir Ağları)	89
Tablo 7.3.	Ortak Başarı Değişkenleri	90

ŞEKİL LİSTESİ

	<u>Sayfa No</u>
Şekil 1.1 : Biyolojik Sinir Sisteminin Blok Gösterimi.....	2
Şekil 1.2 : Beyin ve Sinir Sisteminin Genel Görünüşü.....	2
Şekil 1.3 : Nöron.....	3
Şekil 1.4 : Genel Yapı.....	7
Şekil 2.1 : Statik Hücre Modeli.....	12
Şekil 2.2 : Doğrusal Hareket Fonksiyonu.....	13
Şekil 2.3 : Parçalı Doğrusal Hareket Fonksiyonu	14
Şekil 2.4 : Sigmoid (tanh) Hareket Fonksiyonu.....	15
Şekil 2.5 : Eşik Hareket Fonksiyonu.....	15
Şekil 2.6 : Hiperbolik Tanjant Fonksiyonu.....	16
Şekil 2.7 : FIR Filtre Ağırlıklı Dinamik Hücre Modeli.....	16
Şekil 2.8 : FIR Filtre Olarak Tasarlanan Ağırlıklar.....	17
Şekil 2.9 : Birinci Dereceden Bir Filtreli Dinamik Bir Hücre Modeli.....	17
Şekil 2.10 : Cebrik Hücre Modeli.....	18
Şekil 2.11 : Geri Beslemeli İki Katmanlı Yapay Sinir Ağı.....	20
Şekil 2.12 : Radyal Tabalı Fonksiyon Ağı.....	20
Şekil 2.13 : Çağrışımli Bellek Ağları.....	22
Şekil 2.14 : Vektör Kuantalama Ağı.....	23
Şekil 2.15 : Hopfield Ağı.....	25
Şekil 2.16 : Elman ve Jordan Ağı.....	26
Şekil 2.17 : Kohonen Ağı.....	27
Şekil 3.1 : Gözetimli Öğrenme Yöntemi.....	29
Şekil 3.2 : Gözetimsiz Öğrenme Yöntemi.....	30
Şekil 3.3 : Karma Öğrenme Yapısı.....	31
Şekil 3.4 : Yerel Minimum Değerlere Saplanma Olasılığı Yüksek Bir Fonksiyonun Kesit Görünümü.....	36
Şekil 3.5 : Çok Katmanlı Bir Algılayıcı Geri Yayılım Akış Şeması.....	39
Şekil 5.1 : MATLAB Ana Sayfası.....	72
Şekil 5.2 : MATLAB nntool Komutu.....	72
Şekil 5.3 : MATLAB Grafikselsel Arayüz.....	73
Şekil 5.4 : MATLAB Veri Giriş Penceresi.....	74
Şekil 5.5 : MATLAB Ağ Oluşturma.....	75
Şekil 5.6 : Oluşturulan Ağın Yapısı.....	75
Şekil 5.7 : Model Üzerinde Çalışma Penceresi.....	76
Şekil 5.8 : Modelin Eğitilmesi.....	77
Şekil 5.9 : Eğitim Performansı.....	77
Şekil 5.10 : Ağırlıkların Belirlenmesi.....	78
Şekil 5.11 : Simülasyon Verileri.....	78
Şekil 5.12 : Verilerin MATLAB Ana Sayfasına Aktarılması.....	79
Şekil 5.13 : MATLAB Ana Sayfası - Veri Aktarımı.....	80
Şekil 5.14 : NeuroBuilder.....	80
Şekil 5.15 : NeuroSolutions Veri Aktarımı.....	81
Şekil 5.16 : NeuroSolutions Değişken Tipleri.....	81

Şekil 5.17	: NeuroSolutions Verilerin Ayrıştırılması.....	82
Şekil 5.18	: Gizli Hücre Sayısının Belirlenmesi.....	82
Şekil 5.19	: NeuroSolutions Eğitim Parametrelerinin Seçilmesi.....	83
Şekil 5.20	: NeuroSolutions Eğitim Parametrelerinin Seçilmesi-2.....	83
Şekil 5.21	: NeuroSolutions Sonuçların Gösterilmesi İçin Düzenleme.....	84
Şekil 5.22	: NeuroSolutions Ağ Yapısı.....	85

SEMBOL LİSTESİ

W	: Hücrenin ağırlıklar matrisi
x	: Hücrenin giriş vektörü
v	: Hücrenin net girişi
y	: Hücrenin çıkışı
$\varphi(\cdot)$	: Hücrenin aktivasyon fonksiyonu
M	: Bellek Matrisi
x_i^c	: c sınıfı için eğitime giriş deseninin i'inci elemanı
P	: Sınıf sayısı
N	: İşlemci eleman sayısı
x_i	: j hücresinin i hücresinden gelen girdisi
y_j	: j hücresinin çıktısı
w_{ij}	: i ile j hücreleri arasındaki bağlantı ağırlık değeri
$E(k)$	: k anındaki hata değeri
$w(k)$	: Bağlantı ağırlığı
$d(k)$	: Ağırlık değişiminin eğim bileşeni
k	: Öğrenme katsayısı artma faktörü
j	: Öğrenme katsayısı azaltma faktörü
$a(k)$	: k anındaki öğrenme oranı
κ_α	: Sabit öğrenme hızı skala faktörü
φ_α	: Sabit öğrenme hızı azaltma faktörü
γ_α	: Sabit öğrenme hızı üstel faktörü.
κ_μ	: Sabit momentum hızı skala faktörü
φ_μ	: Sabit momentum hızı azaltma faktörü
γ_μ	: Sabit momentum hızı üstel faktörü
a_{max}	: Öğrenme katsayısının üst sınırı
μ_{max}	: Momentum katsayısının üst sınırı

SINIFLANDIRMADA YAPAY SINIR AĞLARI VE KARAR AĞAÇLARI KARŞILAŞTIRMASI: ÖĞRENCİ BAŞARILARI ÜZERİNE BİR UYGULAMA

ÖZET

Yapay sinir ağları, insan beyninin özelliklerinden olan öğrenme yolu ile yeni bilgiler türetebilme, yeni bilgiler oluşturabilme ve keşfedebilme gibi yetenekleri herhangi bir yardım almadan otomatik olarak gerçekleştirmek amacı ile geliştirilen bilgisayar sistemleridir. Bu özellikleri ile birlikte yapay sinir ağları; sınıflandırma, örnek tamamlama, optimizasyon, üretim, kontrol, tıp, bilim ve mühendislik gibi alanlarda yerini almıştır.

Bu çalışma içerisinde yapay sinir ağlarının sınıflandırma özelliği kullanılarak yapay sinir ağlarının bu konudaki başarısı gözlemlenmiştir. Mayıs 2005 döneminde Defining Critical Factors Affecting Student Success: A Data Mining Approach adlı tez çalışmasında üniversitesi öğrencilerinin başarılı olup olmadıkları ele alınarak sınıflandırma çalışması yapılmış ve bu çalışmada etken olan değişkenler belirlenmiştir. Yapay sinir ağları ile yapılan bu uygulama ile ise aynı verileri temel alarak başarılı ve başarısız öğrenciler belirlenmeye çalışılmış ve bu sınıflandırmada etken olan değişkenler belirlenmiştir.

Çalışmanın temel amacı her iki metodun (Karar Ağaçları-Yapay Sinir Ağları) birbirleri ile karşılaştırılması ve ortaya konulan sonuçların ele alınması ile birlikte üniversite öğrencilerinin başarılarını etkileyen faktörlerin gözlemlenmesidir. Bu kapsamda yapılan uygulama sonucunda yapay sinir ağları %91,77'lik başarı oranı ile, %86 başarı oranına sahip karar ağaçları metodundan daha iyi sonuçlar ortaya koymuştur.

Çalışma içerisinde yapılan uygulama ve daha önceki karar ağaçları uygulaması sonuçları toplu olarak ele alındığında öğrencilerin derse devam yüzdesi, sınıfı, çalışma saati, babasının hayatta olup olmaması, kaldığı yer, babasının eğitim durumu gibi değişkenler başarı üzerinde etkili olarak gözlemlenmiştir.

ARTIFICIAL NEURAL NETWORKS – DECISION TREE COMPARISON IN CLASSIFICATION ANALYSIS: AN APPLICATION ON STUDENTS' SUCCESS

SUMMARY

Artificial neural networks are computer systems developed to perform automatically the capabilities such as new information invention and formation with one of the feature of human brain, learning. With these characteristics, artificial neural networks has taken place in the fields such as classification, optimization, production, control, medicine, science and engineering.

The success of artificial neural networks in the field of classification is observed in this study. In the May 2005 term, an application on classification analysis has been done with defining the students's success as successful and unsuccessful and the critical factors effecting succes are determined, in the thesis named as Defining Critical Factors Affecting Student Success: A Data Mining Approach. The same data is used to define students such as successful or unsuccessful and the critical factors are determined with the study done with artificial neural networks.

The aim of study is to compare each method (decision tree and artificial neural networks) success on classification and to observe the critical factros effecting students' success with the result held from the applications. In this concept, it has been found that the application done using artifical neural networks with the success ratio of %91,77 is better than the one done using decision tree with the success ratio of %86.

After examining both application results, the factors; attendance perception, class, studying hours, father alive, scholarship, residence and father education have been determined to be critical while classifying studenst as successful and unsuccessful.

1. YAPAY SİNİR AĞLARI

1.1 Yapay Sinir Ağları

İnsanların beynin çalışmasını sayısal bilgisayarlar üzerinde taklit etmek istemesi sonucunda yapay sinir ağları ortaya çıkmıştır. Temelde beynin çalışma yapısı, insanlar gibi düşünen ve öğrenen sistemler elde etme fikrinin olması, çalışmaları insan beynini oluşturan hücrelerin incelenmesi üzerine yoğunlaştırmıştır. Bu çalışmalar esnasında her bir nöronun diğer nöronlar ile ilişkili olduğu ve aldığı bazı girdileri çıktıya dönüştürdüğü gözlemlenmiştir. Günümüzde yapay sinir ağları olarak adlandırılan bu alan, birçok nöronun belirli kurallar çerçevesinde bir araya gelerek bir işlevin gerçekleştirmesi esnasındaki yapısal, matematiksel ve felsefi sorunlara cevap bulmaya çalışan bir bilim dalı olmuştur (Şahin, 2001).

İnsan beyninin taklit edilmesi ile ortaya çıkan bu yapılar her ne kadar nanosaniyeler içerisinde işlemleri yerine getirebilme yetisine sahip olsa da insan beyninin milisaniyeler mertebesindeki işlevsel hızına ulaşamamıştır. Buna rağmen karmaşık sistemlerdeki eşleştirmelerdeki hassaslığı ve yapısal esnekliği ile bu alan artan bir ilgi görmektedir (Efe ve Kaynak, 2000). Her ne kadar insan beyni kadar hızlı bir işlevsel yapıya sahip olmasa da sağlamış oldukları hız ile bilgisayar bilimleri, mühendislik, ekonomi ve psikoloji gibi bir çok alanda farklı kullanımları bulunmaktadır (Kuo ve diğ., 2001).

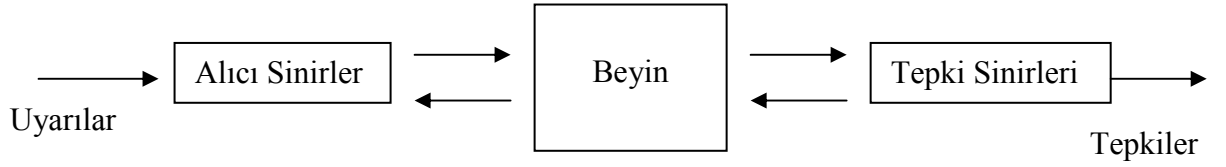
Yapay sinir ağı, beynin belirli bir işi veya fonksiyonu yerine getirme yöntemini modelleyen bir makinedir. Ağın yapısı, elektronik sistemler veya bilgisayar yazılımları ile tanımlanmaktadır. Tanımlanan bu model, sinir hücresi veya işlem birimi adı verilen hücreler arasındaki bağlantıyı kullanmakta ve bu işlemler esnasında öğrenme adı verilen bir süreç ile performansını artırabilmektedir. Bu noktada yapay sinir ağı kavramını deneysel bilgi saklama ve kullanıma hazır hale getirme yeteneğine sahip basit işleme birimlerinden oluşan, çok yoğun, paralel ve dağıtılmış düzende çalışan bir işlemci olarak tanımlamak mümkündür. İnsan beyni ile benzerliği ise bilgiyi öğrenme yoluyla elde etmesi ve bilginin depolanması için sinir hücreleri arası bağı kullanmasıdır.

Öğrenme işlemleri öğrenme algoritmaları olarak adlandırılan ve ağıın istenen modele ulaşmasını sağlayan fonksiyonlardan oluşmaktadır (Haykın, 1999).

Bilgisayarlardaki doğrusal programlamalardan farklı olarak yapay sinir ağıları sinir hücreleri arasındaki bağlantıları, eşik değerleri ve ağ yapısı gibi parametreleri kullanarak farklı ayarlar yapıp eğitim gerçekleştirir. Yapay sinir ağlarında bellek ve işlemci paralel olarak çalışır. Bilgi sinir hücreleri arasındaki bağlarda da saklanabilir veya değiştirilebilir (Bolat, 2003).

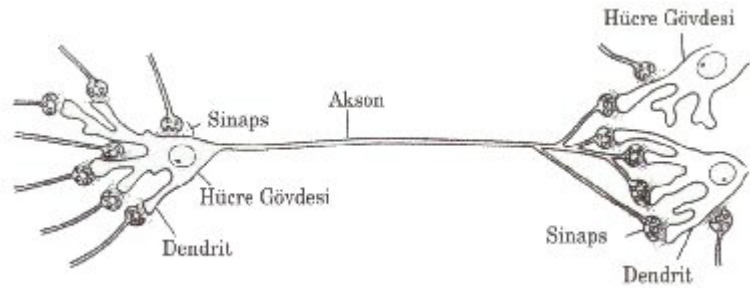
1.2. Biyolojik Sinir Sistemleri

Biyolojik sinir sistemi verinin alınması, yorumlanması ve karar üretilmesi gibi işlevlerin yürütüldüğü beyni merkezde bulunduran 3 katmanlı bir yapıdır. Uyarılar alıcı sinirler ile birlikte elektriksel sinyallere dönüştürülerek beyne iletilir. Beynin oluşturduğu çıktılar ise, tepki sinirleri tarafından belirli tepkilere dönüştürülür.



Şekil 1.1: Biyolojik Sinir Sisteminin Blok Gösterimi

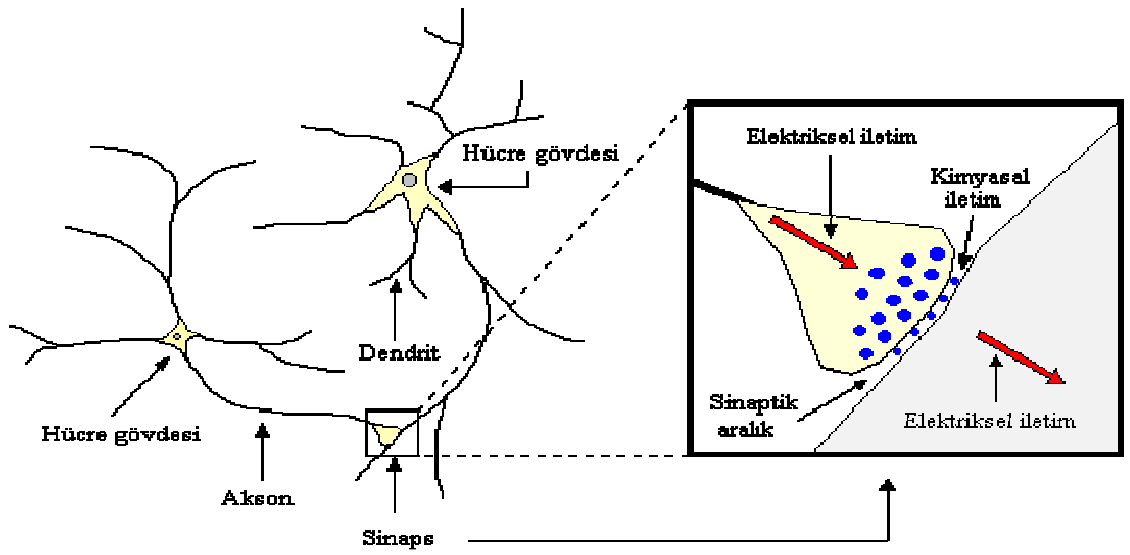
Sinir ağı yapısında bilgilerin alıcı ve tepki sinirleri arasında ileri ve geri beslemeli olarak değerlendirilmesi ve sonucunda tepkilerin oluşması, kapalı bir çevrim sürecine benzemektedir. Temel işlem elemanı sinir hücreleridir. İnsan beyinde yaklaşık 10 milyar sinir hücresi olduğu tahmin edilmektedir. Sinir hücreleri hücre gövdesi, gövdeye giren alıcı lifler (dendrit) ve gövdeden çıkan sinyal iletici lifler (akson) olmak üzere 3 temel bileşenden meydana gelir.



Şekil 1.2. : Beyin ve Sinir Sisteminin Genel Görünüşü

Dentritler aracılığı ile bilgiler diğer hücrelerden hücre gövdesine iletilir. Hücrelerde oluşan çıktılar ise akson yardımı ile bir diğer hücreye aktarılır. Aktarımın gerçekleştiği bu noktada aksonlarda ince yollara ayrılabilmekte ve diğer hücrenin dentritlerini oluşturmaktadırlar. Akson-dentrit bağıntısını oluşturduğu bu noktalara sinaps adı verilir.

Sinapsa ulaşan ve dentritler tarafından alınan bilgiler genellikle elektriksel darbelerdir, fakat bu bilgiler sinapstaki kimyasal ileticilerden etkilenirler. Hücrenin tepki oluşturması için bu tepkilerin belirli bir sürede belirli seviyeye ulaşması gerekmektedir. Bu değer eşik değeri olarak adlandırılır.



Şekil 1.3: Nöron

Tipik bir nöron, hücre gövdesi ve dentritleri üzerine dış kaynaklardan gelen elektrik darbelerinden üç şekilde etkilenir. Gelen darbelerden bazıısı nöronu uyarır, bazıısı bastırır, geri kalanı da davranışında değişikliğe yol açar. Nöron yeterince uyarıldığında çıkış kablosundan (aksonundan) aşağı bir elektriksel işaret göndererek tepkisini gösterir. Genellikle bu tek akson üzerinde çok sayıda dallar olur. Aksondan inmekte olan elektrik işareti dallara ve alt dallara ve sonunda başka nöronlara ulaşarak onların davranışını etkiler. Nöron, çok sayıda başka nöronlardan genellikle elektrik darbesi biçiminde gelen verileri alır. Yaptığı iş bu girdilerin karmaşık ve dinamik bir toplamını yapmak ve bu bilgiyi aksonundan aşağı göndererek bir dizi elektrik darbesi biçiminde çok sayıda başka nörona iletmektir. Nöron, bu etkinlikleri sürdürmek ve molekül sentezlemek için de enerji kullanır fakat başlıca işlevi işaret

alıp işaret göndermek, yani bilgi alışverişidir (<http://bagem.tripod.com/bagem/yz2.html>).

Bir sinir hücresinin tepki hızı günümüz bilgisayarlarına göre oldukça yavaş olmasına rağmen duyuşal bilgilerin değerdendirilmesi son derece hızlıdır. Bu insan beyninin öğrenme, birleştirme, uyarlama ve genelleştirme yeteneđi nedeniyle son derece karmaşık, doğrusal olmayan ve paralel dağılmış bir bilgi işleme sistemi olduğunu göstermektedir.

Yapay sinir ağlarının oluşturulmasında da sinir sistemi model alınmış ve benzer tanımlamalar yapılmıştır. Bu tanımlamalar aşağıdaki tabloda görülebilir (<http://batitrakya.dostweb.com/yapaysiniraglari/yapaysiniraglari.htm>).

Tablo 1.1: Sinir Sistemi ile Yapay Sinir Ağlarının Benzerlikleri

Sinir Sistemi	Yapay Sinir Ađı
Nöron	İşlem Elemanı
Dentrit	Transfer Fonksiyonu
Hücre Gövdesi	Hareket Fonksiyonu
Akson	Eleman Çıkışı
Sinaps	Ağırlıklar

1.3. Yapay Sinir Ağlarının Tarihsel Gelişimi

Yapay sinir ağlarının insan beynini modellemesi, dijital bilgisayarların modellenmesi gerçeğinde de var olan insan beyni gibi işleyen makineler oluşturma isteđi ile ortaya çıkmıştır.

1943 yılında bir nörobiyolojist olan Warren McCulloch ve bir istatistikçi olan Walter Pitts, “Sinir Aktivitesinde Düşüncelere Ait Bir Mantıksal Hesap” başlıklı bir makale ile ilk dijital bilgisayarlara ışık tutmuştur.

1949 yılında, Hebb, daha sonra kendi adı ile anılan öğrenme kuramını, yapay sinir ađına uygulamıştır. Bu kuram “A hücresinin bir aksonu, B hücresini uyaracak ve tekrarlı veya sürekli olarak tetikleyecek kadar yakında ise, hücrelerde B’yi tetikleyen A’nın etkinliğini artıracak bir büyüme işlemi veya metabolik değışiklik olur” şeklindedir.

Marvin Minsky, Hebb kuramı ile birlikte makroskobik zeka kavramını ortaya atmış ve uzman sistemlerin doğmasını sağlamıştır. Frank Rosenblatt, Bronx Yüksek Bilim Okulu’nda gözün hesaplamaları ile ilgilenmiştir. Bu bilim adamları, öğrenme

ve zeka konusunu bilgisayar simülasyonlarında nasıl kullanabileceklerini 1956'da düzenlenen ilk yapay zeka konferansında tartışmışlardır.

1959'da Stanford üniversitesinden Bernard Widrow, "ADALINE" (Adaptive Linear Neuron) olarak adlandırılan bir adaptif lineer elemanı geliştirmiştir. Adaline ve iki tabakalı biçimi olan "MADALINE" (Multiple Adaline) ses tanıma, hava tahmini, karakter tanıma ve adaptif kontrol gibi çok çeşitli uygulamalar için kullanılmıştır. Widrow, telefon hatları üzerinde yansımaları elemine etmeye yarayan adaptif filtreleri geliştirmede, adaptif lineer eleman algoritmasını kullanmıştır. Bu çalışma ile birlikte yapay sinir ağları ilk defa gerçek bir probleme uygulanmıştır.

1970'lerin ilk başlarında Helsinki Teknik Üniversitesi'nden Teuvo Kohonen, öğrenme ve birleşik hafızalar üzerine çalışmalar yapmış bu çalışmalar danışmansız öğrenme metotlarının gelişmesine ışık tutmuştur.

Yapay sinir ağları ile ilgili 1969-1982 yılları karanlık bir dönem olarak görülmektedir. 1969 yılında Minsky ve Papert'in yapmış olduğu çalışma ile yapının XOR lojik problemlerine çözüm getiremediği ispatlanmıştır. Bu gelişme yapay sinir ağlarına olan ilgiyi azaltmış ve karanlık dönemin başlangıcı olmuştur.

1974 yılında Werbos, Geri Yayılmalı Algoritmayı tanıtmıştır. Willshaw ve Malsburg 1976 yılında Özdüzenlemeli Harita adında kümeleme algoritmasını kurmuştur. 1982 yılında Kohonen, özdüzenlemeli harita kuramını beyindeki oluşumların karşılaştırmalı haritasını çıkarabilmek için ortaya koymuştur.

Hopfield , 1982 yılında moleküler biyolojiden beyin kuramcılığına geçiş yapan bir model geliştirmiştir. Günümüzde kendi adıyla anılan bir ağ yapısı mevcuttur ve bu yapı bir çok alanda uygulanmıştır.

1986 yılında, Rumelhart, 1974'te Werbos tarafından bulunan çok katmanlı algılayıcı tipli geri yayılmalı algoritma denen bir eğitim algoritması geliştirmiştir (Güzeliş, 1993).

1987 yılında ilk yapay sinir ağları sempozyumu yapılmış ve sonrasında uygulamalar artmaya başlamıştır.

Günümüzde bu alanda bir çok araştırma grubu bulunmakta ve farklı alanlardan farklı uygulamacılar bu konuda çalışmaktadır (Bolat, 2003 Haykırı, 1999).

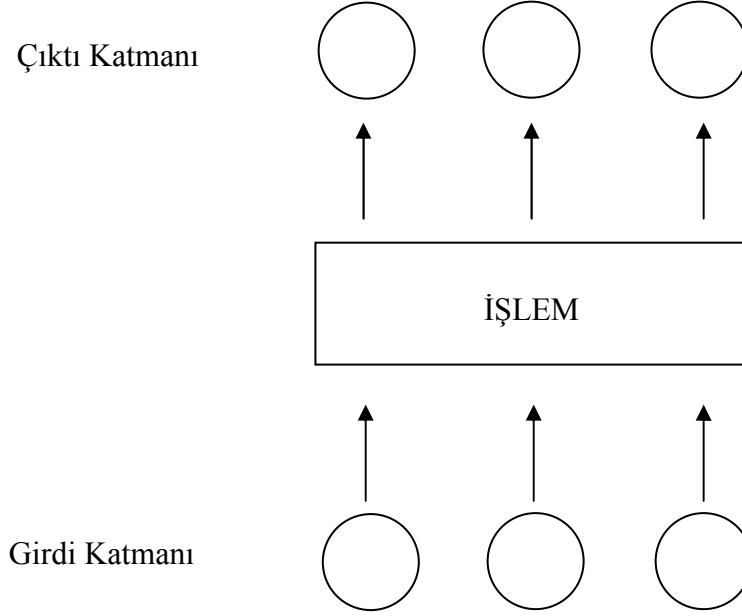
1.4. Genel Olarak Yapay Sinir Ağları Tanımı

Günümüze kadar uygulanmış yapay sinir ağları modellerinin ortak özelliklerini yansıtan tanımlar aşağıdaki şekilde ele alınabilir (Güzeliş, 1993):

- Ağ, hücre denilen yapı taşlarının birbirine belirli bir geometri ile bağlanmasından oluşan bir sistemdir.
- Hücreler, genelde çok girişli ve tek çıkışlı, yüksek dereceden lineer olmayan dinamik alt devrelerdir.
- Tek olan çıkış işareti, çoğullanarak dışarıya çıkış olarak alınabilir veya diğer hücrelere giriş olarak kullanılabilir.
- Bir hücre dış girişleri, komşu hücre çıkışlarını ve bir eşik değerini giriş olarak kabul eder.
- Hücrenin lineer dinamik birimi herhangi bir dereceden olabilir.
- Hücre çıkışındaki lineer olmayan cebrik fonksiyon tek giriş-çıkışlıdır.
- Hücre parametreleri hücreler arasında farklılık gösterebilir.
- Ağ, genelde çok boyutlu katmanların ileri veya geri beslemeli olacak şekilde bağlanmasından oluşur.
- Ağdaki her katman, hücrelerin çok boyutlu bir dizisidir.
- Katman içinde, her bir hücre komşularına aynı şekilde bağlıdır. Böylece katman içi bağlantı geometrisi düzgündür.
- Katmanlar arası bağlantı, bir metrik ile tanımlı düzgün bir geometriye sahiptir.
- Ağda yalnızca bağlantı ağırlıkları değişebilir.
- Bağlantı ağırlıkları, ya eğitici yada eğitici olmayan bir öğrenme kuralı ile değiştirilir yada önceden belirlenen değerlerde sabit tutulur.

1.5. Yapay Sinir Ağlarında Genel Yapı

Literatürde çok farklı ağ yapıları yer almasına rağmen genel bir yapay sinir ağı yapısı Şekil 1.4’te gibi gösterilebilir.



Şekil 1.4: Genel Yapı

Girdi Katmanı: Girdi katmanı en az bir girdi elemanının bulunduğu bölümdür. Girdi katmanında veriler herhangi bir işleme tabi tutulmadan girdileri ile aynı değerde çıktı üretirler (Hawley ve Johnson, 1994) .

Çıktı Katmanı: Çıktı katmanı en az bir çıktıdan oluşur ve çıktı ağ yapısında bulunan fonksiyona bağlıdır. Bu birimlerde girdi katmanında olanın aksine işlem gerçekleştirilir ve birimler kendi çıktılarını üretirler.

İşlem Katmanı: Genellikle “Kara Kutu” olarak adlandırılır. Girdi birimlerinin belirli işlemlere tabi tutulduğu bölgedir. Seçilen ağ yapısına göre işlem katmanının yapısı ve fonksiyonu da değişebilir. Tek bir katmandan oluşabileceği gibi birden fazla katmandan da oluşabilir (Chaudhury ve diğ., 1994) .

Tüm bu katmanlar ele alındığında bir yapay sinir ağı modeli

- Hücreler arası bağlantılar ile değerlerin iletildiği
- Bağlantıların belirli ağırlıklara sahip olduğu ve değerlerin bu ağırlıklarla işleme alındığı
- Hücrelerden çıktı fonksiyonu elde edilmesinde bir hareket fonksiyonunun kullanıldığı

bir model olarak ele alınabilir (Lee ve Park, 2001) .

1.6. Yapay Sinir Ağlarının Özellikleri

Yapay sinir ağlarının günümüzde kullanılan diğer bilgi işlem yöntemlerinden farklılıkları aşağıdaki şekilde sıralanabilir(Seven, 1993 Simpson, 1990):

Paralellik: Alışlagelmiş bilgi işlem yöntemlerinin çoğunda işlemler seri bir düzen içerisinde. Bu düzen özellikle hız sorununa yol açmaktadır. Daha öncede vurgulandığı gibi bilgisayarla beyne göre çok hızlı çalışmasına rağmen beynin toplam hızı bilgisayara göre çok yüksektir. Yapay sinir ağlarında işlemler doğrusal değildir ve bu bütün ağa yayılmış durumdadır. Aynı katmanlar arasında zaman bağımlılığı da bulunmamaktadır. Bu, tüm sistemin eş zamanlı çalışabilmesine olanak vermekte ve hızı çok artırmaktadır. Bu sayede doğrusal olmayan karmaşık problemlerin de çözümlenmesi mümkündür.

Öğrenebilirlik: Mevcut bilgi işlem sistemleri belirli bir algoritma kapsamında çalışmakta ve kendisi ağırlık ya da verileri yenileyememektedir. Bu durumda tam tanımlı olmayan problemler de çözülememektedir. Yapay sinir ağları ile sisteme önceden girilen örnekler kullanılarak ağırlıkların saptanmasını sağlamakta ve her yeni çalışmada bu öğrenme işlemi yenilenebilmektedir. Amaca uygun olacak bir çalışmada önceden ağırlıkların ve bağlantıların verilememesi bir sorun oluşturmakta iken yapay sinir ağlarının örneklerle kendini eğitmesi ve gerekli verileri oluşturması bu sorunu ortadan kaldırmaktadır.

Hata Toleransı: Geleneksel yöntemlerde herhangi bir elemanı yerinden almak o sistemin çalışmasını imkansız kılabilir. Yapay sinir ağlarındaki paralel yapı, ağın sahip olduğu bilginin tüm bağlantılara yayılmasını sağlamaktadır. Bu sayede bazı bağlantıların veya hücrelerin etkisiz hale gelmesi ağın doğru bilgiyi üretmesini önemli derecede etkilemez ve bununla birlikte ağların geleneksel yöntemlere göre hatayı tolere etme yetenekleri yüksektir.

Uyarlanabilirlik: Yapay sinir ağlarında ağırlıkların yeniden yapılandırılabilir olması belirli bir problemi çözmek için eğitilen yapay sinir ağının, problemdeki değişikliklere göre yeniden eğitilebilmesi ve farklı koşullara uyarlanabilmesini sağlamaktadır. Bu özellik yapay sinir ağlarının örnek tanıma, sinyal işleme, sistem tanılama ve denetim gibi alanlarda etkin olarak kullanılmasını sağlamıştır.

Genelleme: Yapay sinir ağı eğitimi sonrasında, eğitim sırasında karşılaşılmayan test örnekleri de değerlendirip, arzu edilen tepkiler üretebilir. Örneğin, karakter tanımlamada bozuk karakterlerin girişlerinde doğru karakterlerin elde edilmesi mümkündür.

Yerel Bilgi İşleme: Yapay sinir ağlarında problemin tümü ile ilgilenmek yerine parçaları ile ilgilenilebilmesi ve görev paylaşımı sayesinde çok karmaşık ve zor problemler çözülebilmektedir.

Gerçekleme Kolaylığı: Karışık fonksiyonların yerine basit işlemlerin kullanılması gerçekleştirme kolaylığı sağlamaktadır.

Donanım ve Hız: Yapay sinir ağı, paralel yapısı sayesinde entegre devre teknolojisi ile gerçekleştirilebilir. Bu özellik, hızlı bilgi işleme yeteneğini artırmaktadır.

1.7. Yapay Sinir Ağlarının Uygulama Alanları

Yapay sinir ağı karmaşık ve ekonomik olmayan çok farklı alanlardaki problemlerin çözümünde başarı ile uygulanmıştır(May,1994). Genel olarak uygulama alanları aşağıdaki şekildedir (Haykin, 1999):

Sınıflandırma: Sinir ağından bir örnek geçirilir. Buna karşılık çıkış olarak bir örnek sınıfı belirtilir.

Tahmin: Sinir ağına verilen değerlerden bir sonuç elde edilmesi için kullanılır. Gelecek tahminleri, talep tahminleri gibi alanlarda kullanılabilir.

Örnek Tamamlama: Eksik bir örnek ağı girildiğinde örneğin eksik kalan kısımları ağı tarafından tamamlanır ve tam örnek çıktı olarak alınır.

Optimizasyon: Ağı bir optimizasyon problemi girilir ve ilk değerleri ifade eden örnek kümesinin girilmesinden sonra çözüme ilişkin değişkenler kümesi çıktı olarak alınır.

Örnek Eşleştirme: Ağı girilen örneğin tipine en uygun örnek tipi belirlenir.

Yapay Zeka: Ses ve görüntü tanımlama gibi alanlarda kullanılabilir.

Gürültü Uzaklaştırma: Gürültülü bir veri ağı girildiğinde gereksiz veriler elenir ve yeni bir veri kümesi elde edilir.

Finansman ve Yatırım: Kredi analizi, sigorta riskleri, opsiyon ve gelecek trend, stok yatırım analizleri gibi olayların modellenmesinde kullanılır.

Üretim: Kalite kontrol ve analizler ile model kurma ve geliştirme ve mevcut otomasyon sistemini geliştirme mümkündür.

Kontrol: Girdiye göre üretilen sonuçlar ile mevcut duruma ait sonuçlar arasında karşılaştırma yapılır.

Tıp: Hastalık nedenlerinin belirlenmesi ve sınıflandırılması, genetik ve kan örnekleme haritalarının oluşturulması gibi işlemlerde kullanılmaktadır.

Bilim ve Mühendislik: Karmaşık modellerin kurulması, doğrusal olmayan modeller, çok değişkenli eğri uyumları, iklim modellemesi gibi konularda uygulamalar yer almaktadır. Geleneksel istatistik ve regresyon testlerinin yetersiz kaldığı bir çok konuda kullanılabilir.

2. HÜCRE VE AĞ YAPISI

Hücreler, yapay sinir ağlarının çalışmasında en küçük bilgi işleme birimleridir. Genel olarak bir hücre girdiler, ağırlıklar, transfer fonksiyonu, hareket fonksiyonu ve çıktılar olmak üzere beş bileşenden meydana gelir.

Girdiler diğer hücrelerden veya dış ortamdan hücreye giren bilgilerdir. Bilgilerin hücreye girişi bağlantılar üzerindeki ağırlıklar vasıtasıyla gerçekleşmekte ve ağırlıklar girdinin hücre üzerindeki etkisini belirlemektedir.

Transfer fonksiyonu, bir hücreye giren net girdiyi hesaplar ve genellikle net girdi girişlerin ilgili ağırlıklar ile çarpımlarının toplamıdır. Transfer fonksiyonu ağ yapısına göre değişerek maksimum alan, minimum alan yada çarpım fonksiyonu şeklinde olabilir.

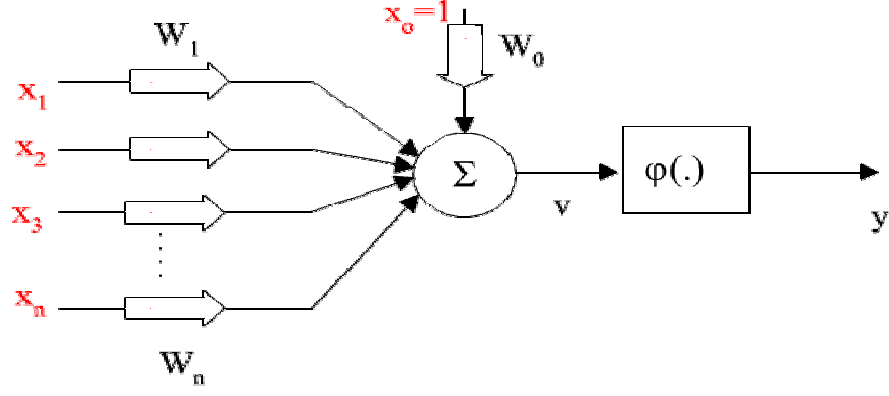
Hareket fonksiyonu, transfer fonksiyonundan elde edilen net girdiyi bir işlemde geçirerek hücre çıktısını belirler ve genellikle doğrusal olmayan bir fonksiyondur.

Hücre modellerinde, net girdiyi artıran +1 değerli polarma girişi yada azaltan -1 değerli eşik girişi bulunabilir ve bu giriş sabit değerli bir giriş ise girdi vektörü (x_0), katsayısı ise (genellikle b ile gösterilir) ağırlık vektörü (W_0) içerisine alınabilir.

Genel olarak hücre modelleri işlevlerine göre statik veya dinamik davranış gösterirler.

2.1. Statik Hücre Modeli

Aşağıdaki şekilde ağırlıkları sabit olarak alıp, geri besleme veya geciktirilmiş sinyallerin kullanılmadığı varsayılırsa hücre statik bir işlevi yerine getirecektir ve bu tür hücreler statik hücre modeli olarak isimlendirilir.



Şekil 2.1: Statik Hücre Modeli

Sıfır indiler polarma girişini ve ağırlığını göstermek üzere statik hücre modelinin denklemi aşağıdaki gibidir:

$$v = \sum W_i x_i \text{ yada } v = \sum W_i x_i + b, y = \phi(v) \quad (2.1)$$

W- hücrenin ağırlıklar matrisi

x- hücrenin giriş vektörü

v- hücrenin net girişi

y- hücre çıkışı

φ(.)- hücrenin hareket fonksiyonu

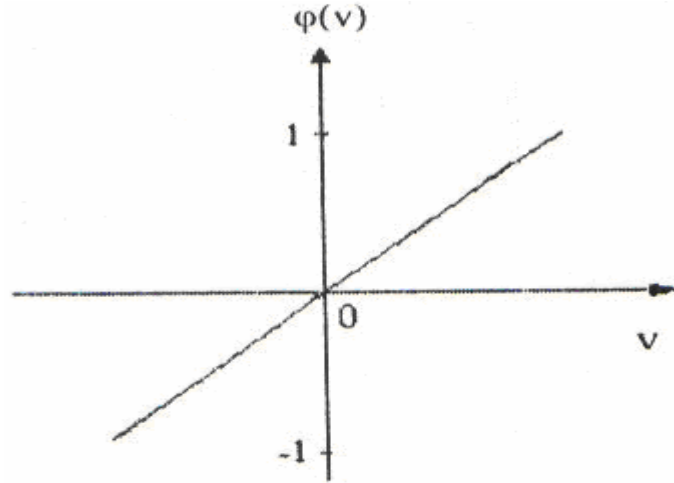
Denklem 2.1'den, **x** giriş vektörünün bileşenlerinin dış (geri beslemesiz) girişler olması durumunda hücrenin doğrusal olmayan statik bir işlevi gerçekleştireceği görülmektedir.

2.2 Hareket Fonksiyonları

Hareket fonksiyonları hücre modellerinde hücrenin işlevine göre çeşitlenmektedir. Hareket fonksiyonları seçime bağlı olmak üzere sabit veya uyarlanabilir parametrelere sahiptir. Hücre modellerinde yaygın olarak kullanılan hareket fonksiyonları aşağıda verilmiştir (Bolat, 2003).

2.2.1. Doğrusal Hareket Fonksiyonu

Doğrusal hareket fonksiyonları doğrusal bir problemi çözmek amacıyla veya katmanlı yapay sinir ağlarının çıkış katmanında kullanılır. Hücrenin net girdisini doğrudan hücre çıkışı olarak verilir. Matematiksel ifadesi $y=v$ şeklinde tanımlanabilir.

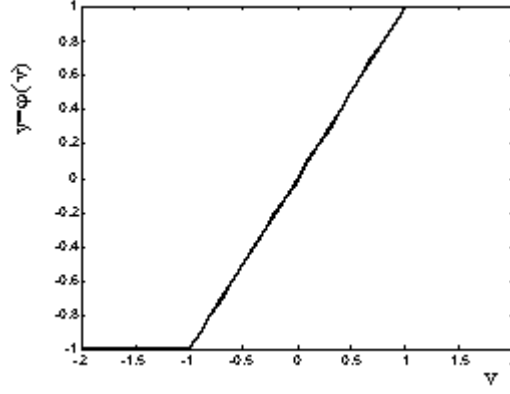


Şekil 2.2: Doğrusal Hareket Fonksiyonu

2.2.2. Parçalı Doğrusal Fonksiyon

Parçalı doğrusal hareket fonksiyonu belirli bir değerden sonra girdiyi hücre çıkışına götürür. Doğrusal olunan bölge aktif çalışma bölgesidir. Parçalı doğrusal fonksiyonun denklemi ve grafiği aşağıda verilmiştir.

$$y = \begin{cases} 1 & v > 1 \\ v & -1 < v < 1 \\ -1 & v < -1 \end{cases} \text{ ise} \quad (2.2)$$



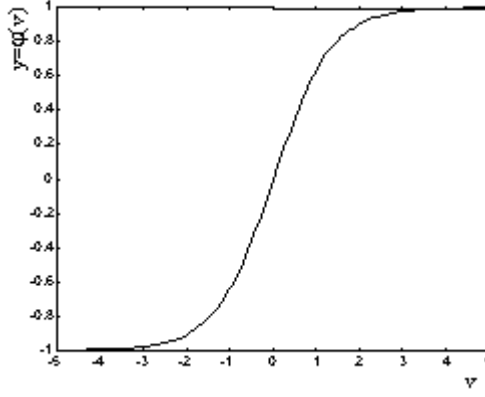
Şekil 2.3 Parçalı Doğrusal Hareket Fonksiyonu

2.2.3. Sigmoid Hareket Fonksiyonu

Doğrusal olmayan problemlerin çözümünde kullanılan Sigmoid fonksiyonu (tanh) türevi alınabilir bir fonksiyondur. Matematiksel ifadesi aşağıdaki gibidir:

$$\varphi(v) = \frac{1}{1 + e^{(-av)}} \quad (2.3)$$

Fonksiyonun matematiksel ifadesindeki a katsayısı, eğim parametresidir. Bu katsayının değerinin artması fonksiyonu eşik fonksiyonuna yaklaştırır. Sigmoid fonksiyonunun değerleri 0-1 arasındadır ve bu aralık içerisinde sürekli türevi alınabilir. Fonksiyonun türevinin alınabilir olması dönüşüm işleminin analitik kontrolünü kolaylaştırmaktadır. Özellikle en küçük kareler metodunda öğrenme algoritmaları için fonksiyonun türevine ihtiyaç duyulur. Bu sebeple doğrusal ve eşik fonksiyonlarının kullanılmadığı bu tür işlemler için sigmoid fonksiyonu tercih edilir.

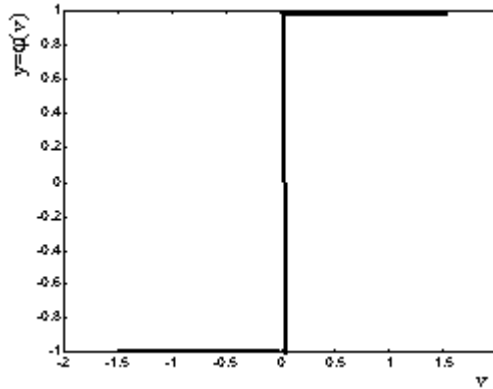


Şekil 2.4 Sigmoid (tanh) Hareket Fonksiyonu.

2.2.4.Eşik Hareket Fonksiyonu

McCulloch Pitts modeli olarak bilinen eşik hareket fonksiyonlu hücreler, mantıksal çıkış verir. “Ya hep ya hiç” temel mantığı ile çalışan bu fonksiyon, sınıflandırıcı ağlarda tercih edilir.

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

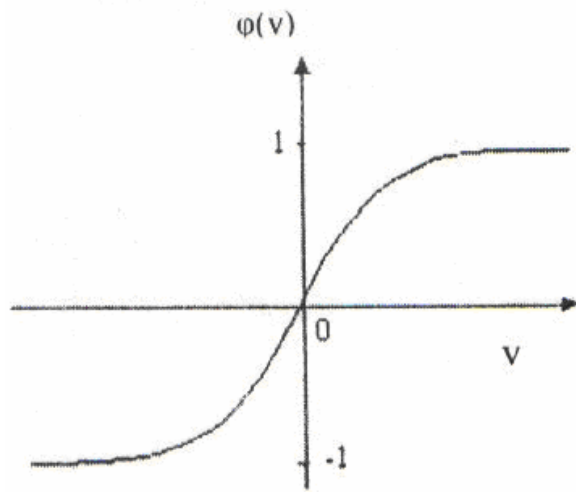


Şekil 2.5 Eşik hareket fonksiyonu

2.2.5. Hiperbolik Tanjant Fonksiyonu

Hiperbolik tanjant fonksiyonu, sigmoid fonksiyonu gibi türevi alınabilen bir fonksiyondur. Sigmoid fonksiyonundan farkı fonksiyon çıkış değerlerinin -1 ile +1 arasında olmasıdır. Fonksiyonun matematiksel ifadesi aşağıda verilmiştir:

$$\varphi(v) = \tanh(v) \quad (2.5)$$



Şekil 2.6: Hiperbolik Tanjant Fonksiyonu

2.3. Dinamik Hücre Modelleri

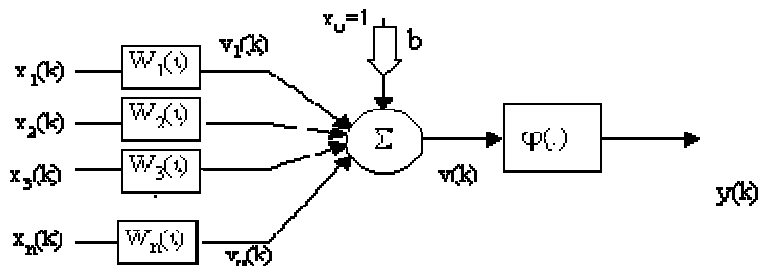
Örüntü tanıma veya sınıflandırma uygulamalarında yapay sinir ağlarında statik hücre modelleri uygun olmakla birlikte sistem modelleme ve denetimi gibi konularda dinamik hücre modellerinin kullanılması gerekmektedir.

Dinamik hücre modelleri genel olarak iki şekilde oluşturulabilir.

- Hücresinin ağırlıkları dinamik bir model (bir filtre) olarak seçilebilir.
- Hücresinin net girdisi dinamik bir modelden (bir filtre) geçirilebilir.

2.3.1 FIR Filtre Ağırlıklı Dinamik Hücre Modeli

Hücre ağırlıklarının sabit seçilmesi yerine bir filtre kullanılmaktadır. Bu sayede hücrenin dinamik davranması sağlanabilmektedir. Aşağıdaki şekilde her bir hücre ağırlığının FIR filtre olarak modellendiği ayrık zamanlı hücre yapısı görülmektedir.

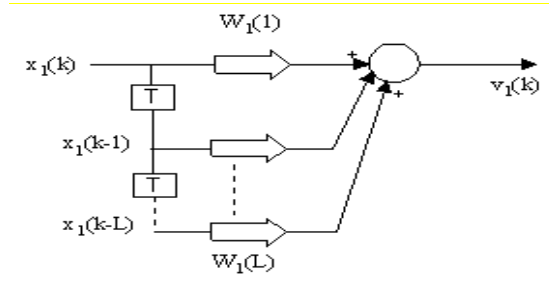


Şekil 2.7: FIR filtre ağırlıklı dinamik hücre modeli

Şekil 2.7’de verilen dinamik hücrenin matematiksel modeli aşağıdaki gibi yazılabilir.

$$v(k) = \sum \sum W_i(l).x_i(k-l) + b \quad (2.6)$$

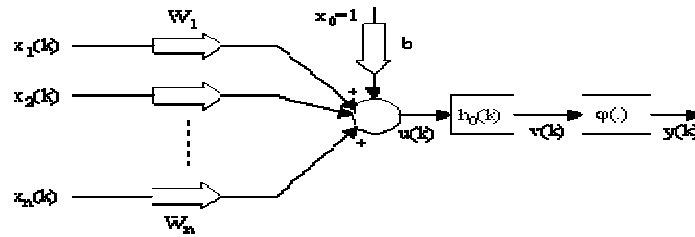
Denklemden L- filtrenin sonlu bellek (gecikme) sayısı olarak tanımlanır. Ağırlıkların gerçekleştirdiği işlevler, sadece birinci hücre girişi için ayrı zamanda şekil 2.8’de verilen blok şema ile gösterilebilir.



Şekil 2.8 FIR filtre olarak tasarlanan ağırlıklar

2.3.2. Birinci Dereceden Bir Filtreli Dinamik Hücre Modeli

Birinci Dereceden Bir Filtreli Dinamik hücre modelinde ağırlıkların dinamik bir modelden seçilmesi yerine hücre net girdisi doğrusal dinamik bir modelden geçirilmektedir. Farklı filtreler seçilebilmesine rağmen genelde birinci dereceden bir filtre kullanılır. Bu durumda filtrenin ani darbe cevabı $h_0(k)$ ya göre hücre modeli aşağıdaki gibi gösterilebilir.



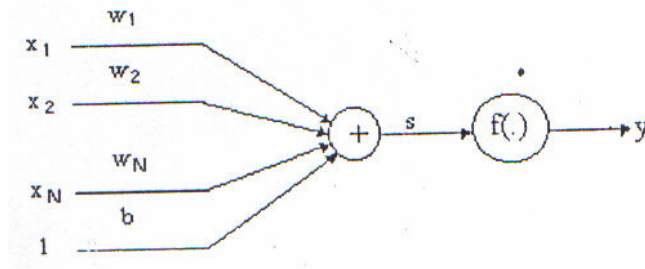
Şekil 2.9 Birinci Dereceden Bir Filtreli Dinamik Bir Hücre Modeli

2.4. Yapay Sinir Ağ Modelleri

Yapay sinir ağları, farklı hücrelerin birbirleri ile farklı şekillerde bağlanmalarından oluşmaktadır. Hücrelerin bağlantı şekilleri, öğrenme kuralları ve aktivasyon fonksiyonlarına göre yapay sinir ağları farklılaşmaktadır (Narendra and Parthasarathy, 1990).

2.4.1. İleri Beslemeli Yapay Sinir Ağları

İleri Beslemeli yapay sinir ağlarında bilgi akışı tek yönlüdür. Bir hücre girdi olarak ancak kendisinden bir önceki hücrenin çıktısını kullanabilir. Katmanlar olarak yapılandırılan bu ağ modelinde giriş katmanı ortamdaki alınan bilgileri aynen orta katmana iletir. Bilginin orta ve çıkış katmanında işlenmesi ile birlikte çıkış değeri belirlenir. Katmanlarda lineer dinamik sistem yer almamaktadır.



Şekil 2.10: Cebrik Hücre Modeli

Yukarıda şekil incelendiğinde toplam ağırlıklı bir y fonksiyonu ele alınırsa w ifadeleri x girişlerindeki ağırlıkları belirtmektedir. B ise sabit girişin ağırlığıdır ve eşik olarak görev almaktadır. Matematiksel olarak aşağıdaki yapı ele alınabilir:

$$s = \sum_{i=1}^N w_i x_i + b \quad (2.7)$$

İfade için ek olarak aşağıdaki tanımlar yapılırsa

$$w = [w_1 w_2 \dots w_N]^T \quad (2.8)$$

$$x = [x_1 x_2 \dots x_N]^T \quad (2.9)$$

ağırlıklı toplam ifadesi şu şekilde olacaktır:

$$s = w^T x + b \quad y = f(s) \quad (2.10)$$

$f(.)$ fonksiyonu olarak değişik ifadeler kullanılmaktadır ve hücreler bu fonksiyona göre adlandırılmaktadır. Burada en çok kullanılan fonksiyonlar sign fonksiyonu, doymalı ve doymasız lineer fonksiyonlar ve logaritmik ve tanjant sigmoid fonksiyonlarıdır.

Bu fonksiyonlara göre hücre, çıkış fonksiyonu olarak lineer fonksiyon alınırsa “Adaline”, sgn fonksiyonu alınırsa “algılayıcı”, sigmoid fonksiyonu alınırsa “genlikte sürekli algılayıcı” olarak adlandırılır.

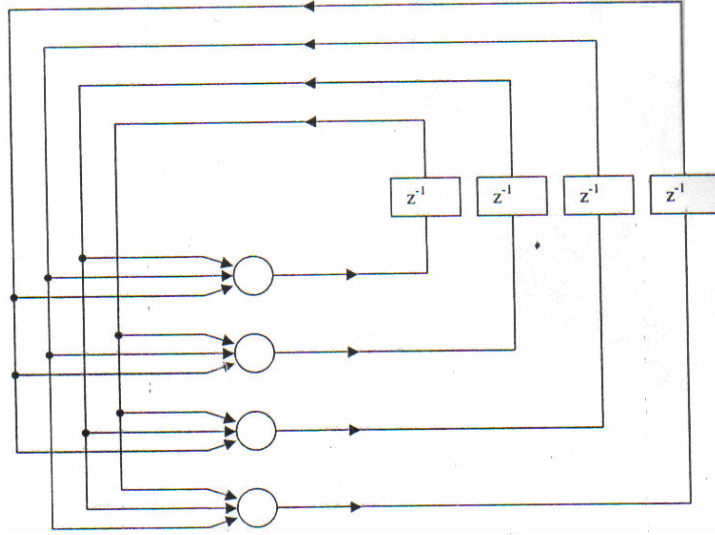
2.4.2. Geri Beslemeli Yapay Sinir Ağları

Geri beslemeli yapay sinir ağlarında bir katmandaki hücre girdi olarak sadece kendinden önceki hücrenin çıktısını değil aynı zamanda o katmanın veya daha sonraki katmanların çıktılarını da alabilir.

Bir geri beslemeli sinir ağında temel olarak çıkış ve ara katmanlardaki çıkışlar, giriş birimlerine veya önceki ara katmanlara geri beslenebilir. Böylece, girişler hem ileri yönlü hem de geri yönlü aktarılabilir. Bu çeşit sinir ağlarında dinamik hafıza özelliği bulunmaktadır ve bir andaki çıkış hem o anın hem de önceki girişlerin sonucudur. Bu özelliği ile geri beslemeli yapay sinir ağları tahmin modellerinde sık kullanılır.

En yaygın geri beslemeli yapay sinir ağları Hopfield, Özdüzenlemeli Harita, Elman ve Jordan ağlarıdır.

Geri besleme, bir katmandaki hücreler arasında olduğu gibi katmanlar arasındaki hücreler arasında da olabilir. Bu yapısı ile geri beslemeli yapay sinir ağları, doğrusal olmayan dinamik bir davranış gösterir. Dolayısıyla, geri beslemenin yapılış şekline göre farklı yapıda ve davranışta geri beslemeli yapay sinir ağları yapıları elde edilebilir. Bu nedenle, bu bölümde bazı geri beslemeli yapay sinir ağları yapılarında örnekler verilecektir. Şekil 2.11’de iki katmanlı ve çıkışlarından giriş katmanına geri beslemeli yapay sinir ağları yapısı görülmektedir (Bolat, 2003 ve Şeker, 2001).



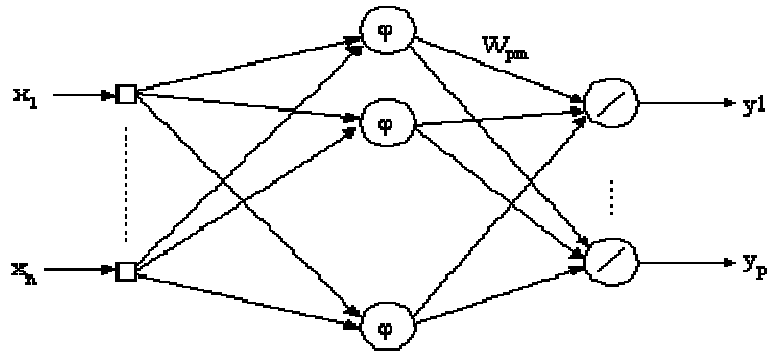
Şekil 2.11 Geri Beslemeli İki Katmanlı Yapay Sinir Ağı .

2.4.3. Radyal Tabanlı Fonksiyon Ağları (RTFA)

Radyal tabanlı fonksiyon ağları katmanlı yapay sinir ağlarında en iyileme için eğiticili geriye yayılmalı algoritmanın kullanılması ile oluşturulur. Bu ağ tasarımında çok boyutlu uzayda eğri uydurma yaklaşımı izlenir. Bu sebeple RTFA'nın eğitimi aslında çok boyutlu uzayda eğitim verilerine en uygun yüzeyi bulma problemidir.

Radyal tabanlı fonksiyonlar, sayısal analizde çok değişkenli interpolasyon problemlerinin çözümünde kullanılmış ve yapay sinir ağlarının gelişmesi ile birlikte bu fonksiyonlardan yapay sinir ağı tasarımında yararlanılmıştır.

RTFA giriş, orta ve çıkış katmanları ile ileri beslemeli yapay sinir ağlarına benzerdir. Fakat giriş katmanından orta katmana dönüşüm radyal tabanlı hareket fonksiyonları ile doğrusal olmayan sabit bir dönüşümdür. Orta katman ile çıkış katmanı arasındaki dönüşüm ise uyarlamalı ve doğrusaldır.



Şekil. 2.12 Radyal tabanlı fonksiyon ağı

Herhangi bir çok deęişkenli ve sürekli $f(x)$ fonksiyonunu yaklaştıırma ya da interpolate etme problemi Eğri Uydurma Teorisi olarak adlandırılır.

İnterpolasyon problemi

$k=1,2,\dots,N$ için

x_k – veri noktası ve d_k -gerçek deęerler olmak üzere

$F(x_k)=d_k$ interpolasyon koşulunu saęlayan $F(.)$ fonksiyonunun bulunması olarak tanımlanır.

2.4.4. Çaęrıřımlı Bellek Aęları

Çaęrıřımlı sistemler, belirli giriş vektörlerini belirli çıkış vektörlerine dönüřtüren yada ilişkilendiren sistemlerdir. Çaęrıřımlı bellek aęları, eğitim sürecinde aęa verilen örneklerin aęrılıkları birlikte saklayarak sonraki süreçlerde aęa verilen örnekler için doęru sonuçları ortaya koyar. Bu özellikleri ile çaęrıřımlı bellek aęları örüntü tanıma ve eldeki eksik verilerden doęru verileri ortaya çıkarma gibi uygulamalarda kullanılır.

Çaęrıřımlı bellek aęlarına ait matematiksel denklemler ařaęıda verilmektedir. Denklemlerde görüleceęi gibi çaęrıřımlı bellek aęları doęrusal veya doęrusal olmayan bir vektör ile ilişkilendirilebilir.

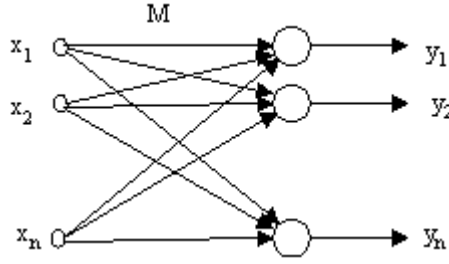
$$Y=M.x, y=\varphi(M,x).x \quad (2.11)$$

x -giriş vektörü

y - çıkış vektörü

M -bellek matrisi

Şekil 2.13’de çağrışımlı bir bellek ağının yapısı gösterilmiştir.



Şekil 2.13 Çağrışımlı Bellek Ağları

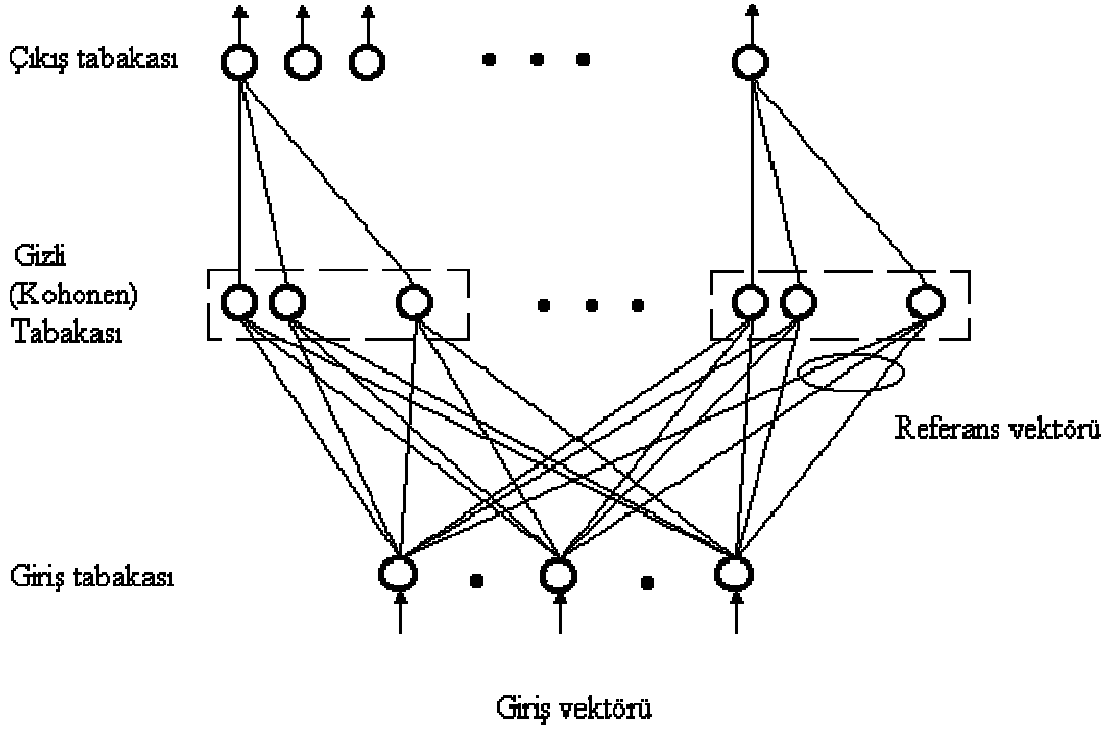
Çağrışımlı bellek ağlarında orta katmandan çıkış katmanına doğrusal bir dönüşüm, giriş katmanından orta katmana ise doğrusal olmayan bir dönüşüm vardır. Bu sebeple, çağrışımlı bellek ağlarının tasarımında ağ giriş uzayını normalize eden n boyutlu bir bölümlendirme yapısı belirlenmelidir. Ağın modelleme yeteneği, bellek gereksinimi ve öğrenme hızı bu bölümlendirme yapısının tasarımı ile ilişkilidir.

2.5. Diğer Yapay Sinir Ağları

Bu kısımda en çok kullanılan ve bilinen yapay sinir ağları tanıtılmıştır.

2.5.1. Vektör Kuantalama

Aşağıdaki şekilde, üç kattan oluşan vektör kuantalama ağı görülmektedir. Katlar giriş, gizli tabaka ve çıkış tabakasıdır. Giriş tabakası ile gizli tabaka arasındaki tüm hücreler bağlıdır, gizli tabaka ile çıkış tabakası arasındaki bağ ise kısımdır. Her çıkış işlemci elemanı farklı bir gizli işlemci elemanına bağlıdır. Gizli tabaka ile çıkış tabakası elemanları arasındaki ağırlıklar ise 1’e sabitlenmiştir. Her gizli işlemci elemanına ait bir referans vektör yer almaktadır. Referans vektörünün elemanları gizli tabaka ile giriş tabakası arasındaki işlemcilerin ağırlıklarından oluşur. Ağ öğrenirken bu değerler değişir. Gizli işlemci elemanları ve çıkış işlemci elemanları ikili (binary) çıkışa sahiptir. Ağa verilen giriş desenine en yakın olan gizli işlemci eleman kümesi “1”, diğerleri “0” değeri üretir. “1” üreten çıkış işlemci elemanı giriş işaretini sınıflar ve her işlemci eleman ayrı bir sınıfa atanır.



Şekil 2.14: Vektör Kuantalama Ağı

Vektör kuantalama öğretim prosedürü en basit şekliyle

- i. Referans vektörlerinin ağırlıklarının başlangıç değerlerini belirle,
- ii. Ağa giriş deseni uygula,
- iii. Giriş deseni ile her referans vektörü arasındaki “Öklit Uzaklığı”nı hesapla,
- iv. Giriş desenine en yakın referans vektörünün (-ki bu vektör kazanan gizli işlemci elemanının referans vektörüdür) ağırlıklarını yeniden düzenle.
- v. Eğer bu gizli işlemci eleman, o çıkış işlemci elemanın bağlı olduğu gizli işlemci eleman kümesine bağlı ise referans vektörünü giriş desenine daha yakın hale getir. Aksi takdirde referans vektörünü uzaklaştır,
- vi. (ii) numaralı adıma yeni bir giriş deseni ile dön ve işlemlere bütün giriş desenleri sınıflandırılınca kadar devam et.

olarak belirtilebilir.

2.5.2. Hopfield Ağı

Şekil 2.15’de Hopfield ağıının bir şekli gösterilmiştir. Bu ağ genellikle ikili (0 veya 1) ve bipolar (+1 veya -1) girişler kabul eder. Daha önce belirtilen geri beslemeli yapı şeklinde tek tabaka işlemci elemanları vardır ve her işlemci eleman bir diğerine bağlanmıştır.

Ağın eğitilmesi tek bir adımdan oluşur. w_{ij} ağırlıkları aşağıdaki gibi hesaplanır:

$$w_{ij} = \begin{cases} \frac{1}{N} \sum_{c=1}^P x_i^c x_j^c, & i \neq j \\ 0, & i = j \end{cases} \quad (2.12)$$

w_{ij} : i işlemci elemanından j işlemci elemanına olan bağlantının ağırlığı

x_i^d : c sınıfı için eğitime giriş deseninin i ’inci elemanı

P : sınıf sayısı

N : işlemci eleman sayısı

Denklem (2.12)’de $w_{ij} = w_{ji}$ ve $w_{ii} = 0$ durumları ağın kararlılığını garantiler. Böylece bilinmeyen bir bilgi ağı girildiğinde, ağın çıkışları bilinmeyen bir desen elemanlarına eşitlenir, yani

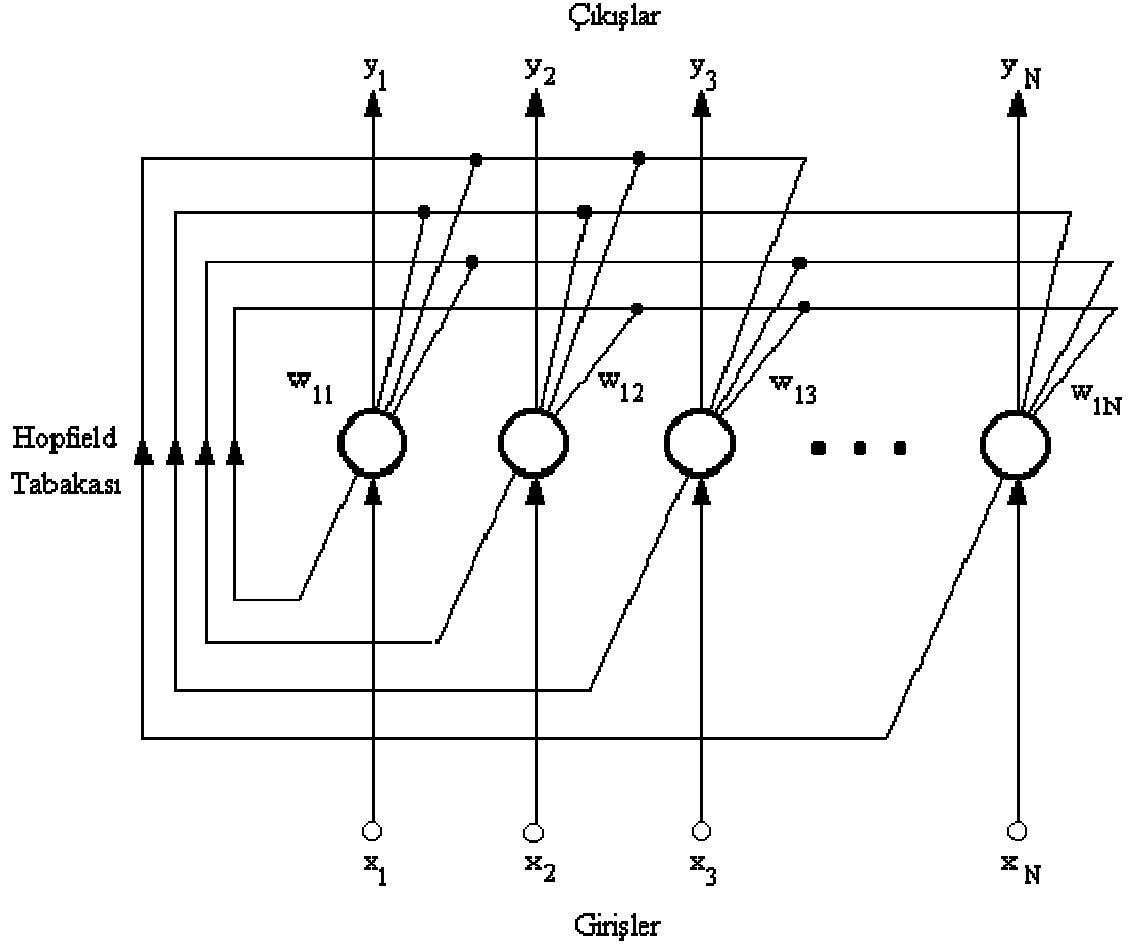
$$y_i(0) = x_i \quad 1 \leq i \leq N \quad (2.13)$$

Bu başlangıç değerleri ile başlayarak Hopfield ağı bir sonraki denklemi kullanarak minimum enerji durumuna geçmek için döngüye girer.

$$y_i(k+1) = f \left[\sum_{j=1}^N w_{ij} y_j(k) \right] \quad 1 \leq i \leq N \quad (2.14)$$

burada f aşağıdaki gibi tanımlanan keskin sınırlama (hard limiting) fonksiyonudur.

$$f(x) = \begin{cases} -1 & x < 0 \\ 1 & x > 0 \end{cases} \quad (2.15)$$

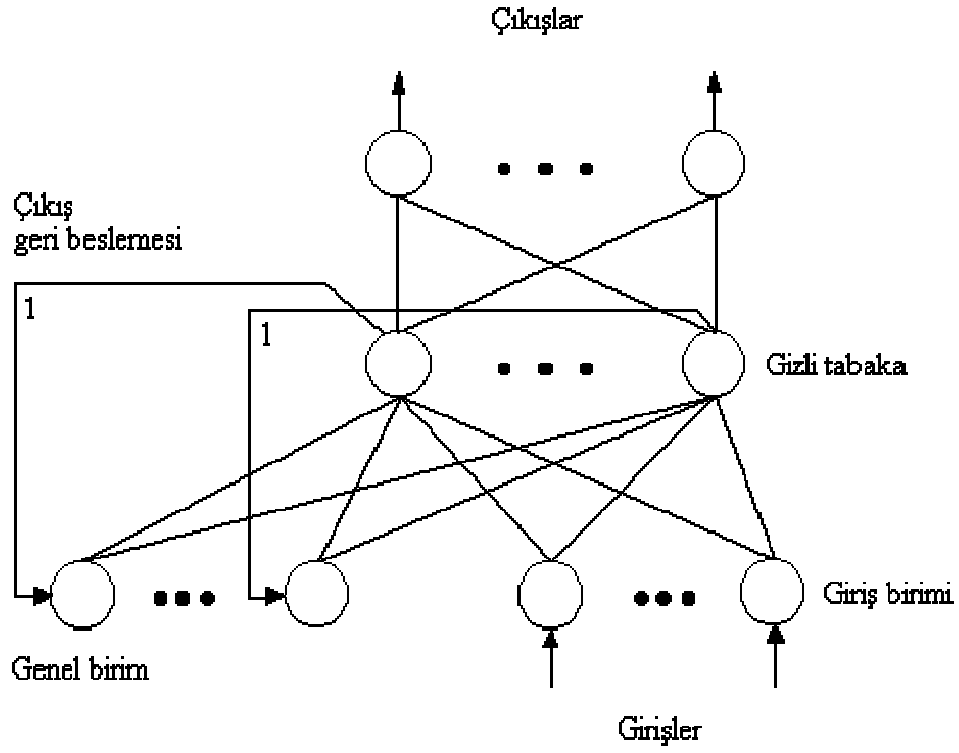


Şekil 2.15: Hopfield Ağı

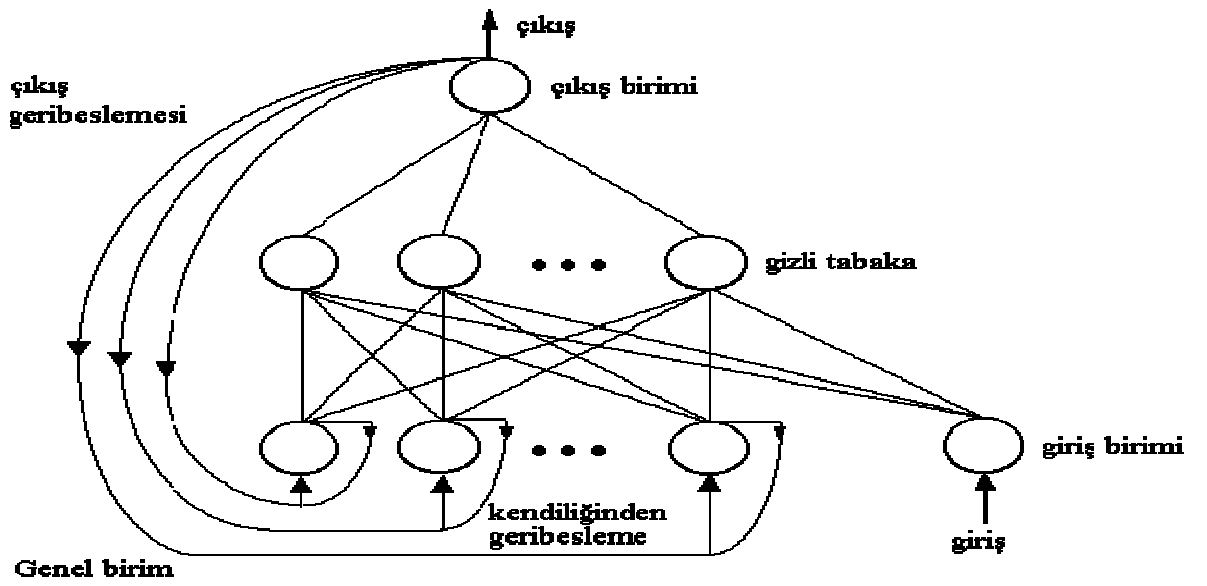
2.5.3. Elman ve Jordan Ağları

Şekil 2.16a ve b Elman ve Jordan ağlarını göstermektedir. Her iki ağda da gizli tabaka haricinde özel bir gizli tabaka yer almaktadır. Bu tabaka “durum” tabakası olarak adlandırılır ve bu tabakada gizli tabakadan veya çıkış tabakasından geri besleme işaretleri alınır.

Jordan ağında Elman ağından farklı olarak durum tabakasındaki her işlemci elemandan kendisine bağlantılar vardır. Her iki ağda da durum tabakasındaki işlemci elemanların çıkışları ileriye doğru gizli tabakaya verilmektedir. Bu ağlar, sadece ileri doğru bağlantılar göz önüne alındığında ve geri besleme bağlantılarına sabit değerler verildiğinde sıradan ileri beslemeli ağlar haline gelirler.



(a)



(b)

Şekil 2.16: a) Elman ağı, b) Jordan ağı

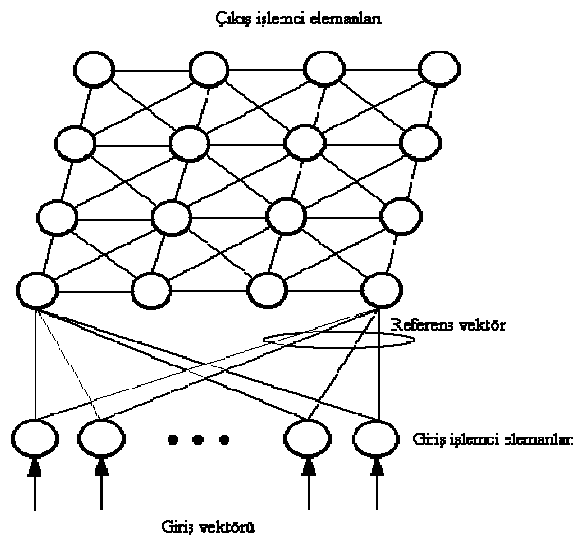
2.5.4. Kohonen Ağı

Şekil 2.17’de Kohonen ağı gösterilmiştir. Kohonen ağı, giriş tabakası ve çıkış tabakası olmak üzere iki tabakadan oluşur.

Çıkış tabakasındaki işlemci elemanlar düzenli iki boyutlu aralıklarda düzenlenmiştir. Çıkıştaki tüm işlemci elemanlar giriş tabakasındaki işlemci elemanlarına bağlıdır. Bağlantıların ağırlıkları kullanılarak, çıkış işlemci elemanının referans vektörünün elemanlarını oluşturulur.

Kohonen ağıнын öğrenme adımları aşağıdaki gibidir:

- i. Çıkış işlemci elemanlarının bütün referans vektörlerini küçük rasgele değerlere çek,
- ii. Bir giriş desenini al,
- iii. Kazanan çıkış işlemci elemanını belirle (Bu giriş desenine en yakın referans vektörüne sahip işlemci elemandır. Referans vektörü ile giriş vektörü arasındaki öklit uzaklığı genellikle uzaklık ölçüsü olarak alınır.),
- iv. Kazanan işlemci elemanın ve onun komşularının referans vektörünü güncelleştir. Bu referans vektörlerini giriş vektörüne yaklaştırılır. Bu yaklaştırma (ayarlama) kazanan işlemci eleman için en fazla ve bu işlemci elemandan uzaklaştıkça daha azdır. Öğrenme ilerledikçe komşuların sayısı azalmakta ve öğrenme sonunda sadece kazanan işlemci elemanın referans vektörü ayarlanmaktadır.



Şekil 2.17: Kohonen Ağı

3. ÖĞRENME VE ÖĞRENME KURALLARI

3.1 Yapay Sinir Ağlarında Öğrenme

Sinir ağları zaman için öğrenebilmektedirler. Bu bakımdan adaptif özellik taşırlar. Yani, sinir ağları geçmiş deneyimlerine dayanarak problem çözme yeteneklerini geliştirebilir. Yapay sinir ağlarındaki bu olay “öğrenme” olarak adlandırılır. Öğrenme işlemi istenilen çıktıları elde edebilmek amacıyla bağlantıların ağırlıklarının belirlenmesidir.

Literatürdeki çoğu öğrenme kuralı Hebb adlı bilim adamının hipotezi ile ilişkilidir. Hebb, sinir ağlarındaki öğrenme kavramını şu hipotez ile açıklamıştır:

“Bir i sinir hücresi bir j sinir hücresi tarafından sürekli bir biçimde uyarılır ise, bu iki sinir hücresi arasında, i hücresinin j hücresi tarafından uyarılma etkinliğini artıracak şekilde, bazı metabolik değişimler ve büyüme süreçleri gözlenir.” (Hebb,1949) .

Bu hipotezi matematiksel olarak aşağıdaki gibi ifade etmek mümkündür:

x_i : j hücresinin i hücresinden gelen girdisi

y_j : j hücresinin çıktısı

w_{ij} : i ile j hücreleri arasındaki bağlantı ağırlık değeri

$\alpha > 0$: öğrenme hızı parametresi, olmak üzere

$$w_{ij}^{yeni} = w_{ij}^{eski} + \alpha x_i y_j \quad (3.1)$$

Eğitimin amacı istenilen yapay sinir ağı yapısını elde etmek için ağırlıkların kontrol altında tutulmasıdır.

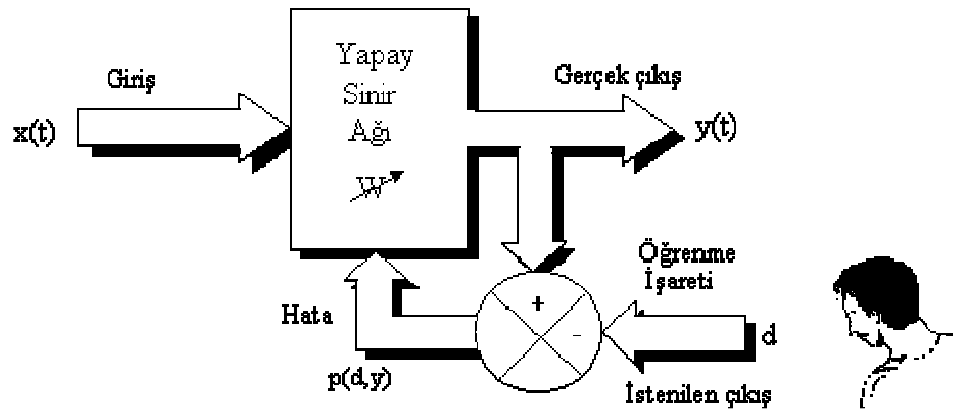
Yapay sinir ağlarının eğitiminde gözetimli ve gözetimsiz olmak üzere iki farklı yöntem kullanılır.

3.1.1.Gözetimli Öğrenme Yöntemi

Eğitim ve sonuçların test edilmesi amacıyla yapay sinir ağı için örnek bir çok veri toplanır. Bu verilerden seçilen bir alt küme “eğitim grubu” olarak kullanılır. Eğitim grubundaki örnekler ağı verilir ve ağın ürettiği çıktılar ile üretilmesi istenilen çıktılar karşılaştırılır. Doğru değerlere ulaşabilmek amacıyla ağıdaki ağırlıklar güncellenir. Bu sayede ağın çıktılarındaki hatalar azaltılacaktır. Ağırlıkların güncellenmesi ile birlikte tek bir örnek grubunun ağı verilmesi bir devir olarak adlandırılır. Eğitim devirleri sonuçların etkin bir hal alması işlemine kadar sürdürülür.

Ağın eğitilmesi sonrasında direk uygulamaya konulması mümkün değildir. Öncelikle yapay sinir ağı test edilmeli ve eğitim süreci sonucunda elde edilen başarı değerlerinin test sonucunda da ediliş edilmediğı kontrol edilmelidir. Eğitim aşamasında genelde küçük hata değeri alınabilir. Burada önemli olan test aşamasında bu değeri yaklaşabilmektir (Şahin, 2001).

Bu alanda bilinen ilk yapay sinir ağı 1958 yılında Rosenblatt tarafından ortaya konulmuştur. Algılayıcılar olarak adlandırılan bu yapay sinir ağı sınıflama görevini doğrusal ayrılabilir olmayan durumlarda kullanamaması sebebi ile zayıf bulunmuştur. Algılayıcıların bu yetersizlikleri yapay sinir ağları alanındaki çalışmaları uzun süre duraklatmıştır. 1986 yılında Rumelhart, Hinton ve Williams tarafından yapılan araştırmalar sonucunda çok katmanlı ileri beslemeli yapay sinir ağlarının ortaya konulması ile birlikte yapay sinir ağları alanında doğrusal ayrılabilirlikteki kısıt kalkmış ve çalışmalar tekrar önem kazanmaya başlamıştır (Şahin, 2001).



Şekil 3.1: Gözetimli Öğrenme Yöntemi

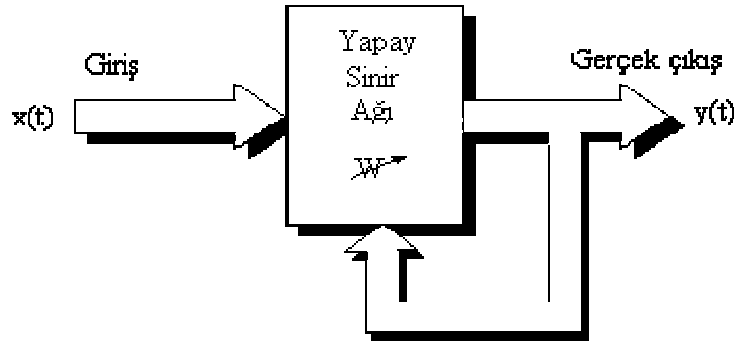
3.1.2. Gözetimsiz Öğrenme Yöntemi

Gözetimsiz eğitim ağlarında da bir dizi girdi ağı verilir. Fakat ağın verileri işlenmesi sonucunda oluşan çıktılar kıyaslama yapmak için kullanılmaz. Ağın öğrenme sürecinde eğitim grubundaki verilerin özelliklerini keşfetmesi beklenir. Bu özellikler keşfedilerek girdiler farklı özelliklerine göre gruplanırlar (Şahin, 2001).

Gözetimli öğrenme ile gözetimsiz öğrenme arasında temelde iki farklılık bulunmaktadır:

- Eğitim setindeki girdiler için çıktı değerlerinin bilinmesine gerek yoktur.
- Bu algoritma ile eğitilen ağ yapılarının kullandığı yöntem gözetimli öğrenme yöntemine göre daha anlaşılabilir ve ulaşılabilir.

En yaygın olarak bilinen gözetimsiz sinir ağı Teuvo Kohonen tarafından bulunan Kohonen ağı yapısıdır. Kohonen ağı yapısında rekabete dayalı eğitim adı verilen bir eğitim süreci kullanılmıştır. Rekabete dayalı eğitim sürecinde, ağı yapısındaki birimlerin ağırlıklarının belirlenmesinde bir yarış söz konusudur. Bu yarış sonucunda sadece bir yada birbirine komşu birkaç birim ağırlık değerlerini değiştirebilir. Diğer eğitim süreçlerinde ise tüm birimler için ağırlıklar güncellenmektedir.

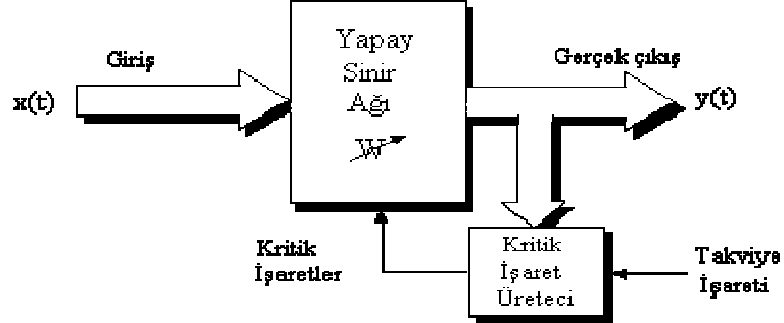


Şekil 3.2: Gözetimsiz Öğrenme Yöntemi

3.1.3. Karma öğrenme

Gözetimli öğrenme kuralına yakın bir metottur. Ağ çıkışından alınan değerler karşılaştırma için kullanılmaz fakat bu değerler iyi yada kötü olarak değerlendirilerek ağı geri verdikleri için gözetimli öğrenmeye yakındırlar. Şekil 3.3’de takviyeli öğrenme yapısı gösterilmiştir. Optimizasyon problemlerini çözmek için Hinton ve

Sejnowski'nin geliřtirdiđi Boltzmann kuralı karma öğrenmeye örnek olarak verilebilirler (Masters, 1993) .



Şekil 3.3: Karma Öğrenme Yapısı.

3.2. Öğrenme Parametreleri

Yapay sinir ağlarında beklenen eğitim seviyesine ulaşamamasının sebebi yetersiz eğitim yapılması veya yetersiz gizli sinir hücresi kullanılmasıdır. Gizli sinir hücrelerinin az sayıda kullanılması yetersiz öğrenmeye sebep olur. Fakat çok sayıda gizli sinir hücresi kullanılması da istenilen durum değildir. Bu durumda öğrenme aşamasında istenilen sonuçlar elde edilse bile aslında öğrenme gerçekleşmemiştir. Böyle bir durumda sinir ağı sadece verileri ezberleyerek işlem yapmış olacaktır. Geri yayılmalı öğrenme algoritmasında en çok eleştirilen noktalardan birisi de performansın kullanılan gizli hücre sayısına bağılı olması, eğitim seti üzerinde doğru ağırlıklara ulaşmak için çok zaman gerekmesi yani devir sayısının fazla olmasıdır (Gately,1996).

Yapay sinir ağı modellemesinde optimum gizli hücre sayısını elde etmek için “geometrik piramit kuralı” olarak adlandırılan metot kullanılır. Piramat kuralı hücre sayısının bir piramite benzer şekilde girdilerden çıktılarına doğru azalması gerektiđi varsayımına dayanır.

Girdi hücre sayısının n ve çıktı hücre sayısının m olduđu bir yapay sinir ağı için geometrik piramit kuralına göre gizli hücre sayısı

$$\sqrt{m*n} \quad (3.2)$$

denklemleri ile elde edilir.

Gizli hücre sayısının bulunmasında kullanılan farklı yöntemlerde bulunmaktadır. Bunlardan birisi gizli hücre sayısının girdi hücre sayısının %75'i olarak belirlenmesi gerektiğini ileri sürer. Gruca ve Klenz tarafından önerilen gizli hücre sayısı ise n girdi hücresi ve m çıktı hücresi sayısı olmak üzere $(2n + m)^{1/2}$ ile $(2n+1)$ arası bir değer olmalıdır.

Gizli hücre sayısının bulunmasındaki en sağlıklı yol küçük sayıda gizli hücre sayısı ile başlayıp hücre sayısını artırarak her seferinde belirlenen performans kriterine göre yapay sinir ağının etkinliğini hesaplayarak ideal olan sayısı belirlemektir. Metot ideal sonuçları verse de metot gereğince her bir gizli hücre sayısı için eğitimin tekrar edilmesi ve test edilmesi gerektiği için metot çok fazla zaman almaktadır. Bu süreç boyunca dikkat edilmesi gereken bir diğer nokta ise yapay sinir ağının eğitim grubunu ezberlemesini engellemektir. Eğitim sayısının artırılması hatayı azaltır gibi görülse de test işlemine geçildiği durumda performansı azaldığı görülecektir (Yoon ve diğ.,1993).

Eğitim sürecinde gizli hücre sayısı ve eğitim sürecinin uzunluğu haricinde bir diğer önemli nokta da başlangıç olarak seçilen ağırlık değerleridir. Bu değerler rassal oldukları için her bir gizli hücre sayısı için ortalama bir deneme rakamı belirlenmelidir ve her bir gizli hücre sayısı farklılaştırılmasında için bu sayı kadar deneme yapılmalıdır. Bu sayede rassallıktan kaynaklanan yanılgılar ortadan kaldırılmış olacaktır. Fakat tekrarlı öğrenmelerin sonucunda en ideal gizli hücre sayısına ulaşıp ulaşılmadığının anlaşılması tam olarak mümkün değildir. Bu sebeple eğitim sürecinde belirlenen hata payı toleransı da önem arz etmektedir. Hiçbir teknik ile %100 bir doğrulukta tahmin yada sınıflama gerçekleştirilemeyeceği dikkate alınmalıdır (Şahin, 2001).

3.3. Öğrenme Kuralları

Değişik ağ modelleri için çok farklı öğrenme kuralları yer almaktadır. Bu bölümde farklı öğrenme kuralları ele alınacaktır. Tek katmanlı ileri beslemeli ağlar için eğitici öğrenme kuralları Algılayıcı Öğrenme Kuralı, Delta Öğrenme Kuralı, Widrow-Hoff Öğrenme Kuralı ve eğitici öğrenme kuralları Hebb Öğrenme Kuralı ve Kazanan-Herşeyi Alır Öğrenme Kuralı'dır. Çok katmanlı ileri beslemeli ağlar için eğitici öğrenme kuralı olarak Geri Yayılmalı Algoritma örnek olarak verilebilir.

3.3.1. Genel Eğitici Öğrenme Kuralı

Amari tarafından kullanılan bir kuraldır. Bu kurala göre j . Hücrenin ağırlık vektörü w_j , olmak üzere $w_j(0)$ başlangıç değerinden başlanarak ardışık olarak değiştirilir. Her bir adım bu şekilde hesaplanır. $(k+1)$. Adım için w_j ağırlık vektörü aşağıdaki şekilde hesaplanır:

$$w_j(k+1) = w_j(k) + \Delta w_j(k) \quad (3.3)$$

w_j , ağırlık değişimini belirtmektedir. Değer olarak x giriş vektörü ile öğrenme işareti r ile orantılıdır. Öğrenme işareti r , ağırlık, istenen çıkış değeri ve giriş vektörünün bir fonksiyonudur. Buna göre k . adım için

$$w_j(k) = \alpha r [w_j(k), x(k), d_j(k)] x(k) \quad (3.4)$$

ile ifade edilir.

Denklem içerisindeki α değeri öğrenme oranı olarak adlandırılır. Öğrenme oranı öğrenme hızını belirleyen pozitif bir sayıdır.

Eşik değeri de sabit girişin bir ağırlığı olduğu için

$$b_j(k+1) = b_j(k) + \Delta b_j(k) \quad (3.5)$$

$$b_j(k) = \alpha r_b [b_j(k), d_j(k)] \quad (3.6)$$

olarak yazılabilir (Zurada, 1992).

3.3.2. Hebb Öğrenme Kuralı

Hebb öğrenme kuralına göre öğrenme sinyali hücre çıkışına eşittir. Öğrenme sinyalini r ile simgelersek $r = f(w_i^t x)$ ile belirtilir. Bu durumda ağırlık vektöründeki artış ise

$$w_i = c f(w_i^t x) x \quad (3.7)$$

ile veya

$$w_{ij} = c o_i x_j \quad (3.8)$$

ile belirtilir.

Bu öğrenme kuralında başlangıç olarak sıfıra yakın değerler ile başlanılmalıdır Hebb öğrenme kuralına göre giriş ve çıkış değerlerinin çarpımı pozitif ise w_{ij} ağırlığında bir artış, negatif ise bir azalış gerçekleşir (Moumin, 1995).

3.3.3. Delta Öğrenme Kuralı

Delta öğrenme kuralı eğitici öğrenme durumunda sürekli hareket fonksiyonlarında kullanılır. Bu kural içerisinde öğrenme sinyali delta olarak adlandırılmaktadır. (Zurada, 1992)

$$r = [d_i - f(w_i^t x)] f'(w_i^t x) \quad (3.9)$$

bu denklemde $f'(w_i^t x)$ hareket fonksiyonu $f(\text{net})$ 'in bir türevidir. Ağırlık uygulamaları için

$$w_{ij} = \eta(d_i - o_i) f'(\text{net}_i) x_j, j = 1, 2, \dots, n \quad (3.10)$$

denklemini kullanılmaktadır. Denklem içerisindeki η değeri rassal pozitif bir sayıdır ve öğrenme sabiti olarak adlandırılır.

Bu kural istenilen çıkış değeri ile elde edilen çıkış değeri arasındaki hatanın karesinin minimum olması için çalışmaktadır (Moumin, 1995).

3.3.4. Widrow-Hoff Öğrenme Kuralı

Delta Öğrenme kuralına benzer olarak eğitici bir öğrenme kuralıdır. Fakat bu öğrenme kuralında hareket fonksiyonunun bir önemi bulunmamaktadır. Bu kuralda amaç, istenilen çıkış değeri ile gerçek çıkış değeri arasındaki hatayı minimize etmektir. Bu kural için öğrenme sinyali denklemi aşağıdaki gibi verilebilir.

$$r = d_i - w_i^t x \quad (3.11)$$

Bu denkleme karşılık gelen ağırlık uygulaması ise

$$\Delta w_{ij} = c(d_i - w_i^t x) x_j, j = 1, 2, \dots, n \quad (3.12)$$

olarak belirtilidir. Temel olarak Widrow-Hoff öğrenme kuralı Delta öğrenme kuralının özel bir halidir (Moumin, 1995).

3.3.5. Algılayıcılı Öğrenme Kuralı

Algılayıcılı öğrenme kuralında öğrenme sinyali istenilen çıkış değeri ile gerçek çıkış değeri arasındaki fark olarak alınır. Bu ise bu kural ile uygulanan metodun eğiticili öğrenme olması gerekliliğini getirir. Öğrenme sinyali aşağıdaki şekilde ifade edilebilir:

$$r = d_i - o_i \quad (3.13)$$

Sinir ağının çıkış değeri olan $o_i = \text{sgn}(w_{it}x)$ 'dir. d_i ise sinir ağından istenilen çıkış değeridir. Bu kural için kullanılan ağırlık değerleri ise

$$w_{ij} = c(d_i - \text{sgn}(w_{it}x))x_j, j=1,2,\dots,n \quad (3.14)$$

ile belirtilir.

Ağırlık uyarlamalarında da görüleceği gibi bu kural sadece bipolar ikili çıkış veren sinir ağları için kullanılır. Ağırlık uyarlaması ise sadece çıkış değeri hatalı olduğu zaman yapılır (Moumin, 1995).

3.4. Geriye Yayılmalı Öğrenme Algoritması

Geriye Yayılmalı Öğrenme Algoritması, hataları geriye doğru çıkıştan girişe azaltmaya çalıştığı için bu ismi almıştır. Temelde, çıktı katmanındaki hata değerinin eğim hesabı ile işlem yönüne ters şekilde geriye doğru yayılma metot olarak kullanılmaktadır. Geriye yayılmalı algoritma çok katmanlı ve ileri sürümlü ağ yapılarının eğitimi amacıyla geliştirilmiş ilk etkin yöntemdir (Masters, 1993).

Geriye yayılmalı algoritma günümüzde en çok kullanılan öğrenme algoritmasıdır. İlk olarak 1974'te Werbos tarafından geliştirilmiş fakat bilim dünyasına duyurulamamıştır. 1986 yılında Rumelhart tarafından tekrar geliştirilerek duyurulmuştur (Hecht, 1990).

Geri yayılmalı algoritma, temelde bir eğim iniş algoritmasıdır. Çok değişkenli fonksiyonlarda eğim, fonksiyonun en dik olduğu doğrultuyu gösterir. Eğim üzerindeki küçük değişiklikler fonksiyonda maksimum değişikliklere sebep olacaktır. Eğimin zıt yönünde atılacak olan bir adım fonksiyon değerinde maksimum azalış

sağlayacaktır. Geriye yayımlı algoritmada fonksiyon olarak eğitim setinin ağ yapısına tanıtılması sonrasında elde edilecek olan hata fonksiyonu kullanılmaktadır.

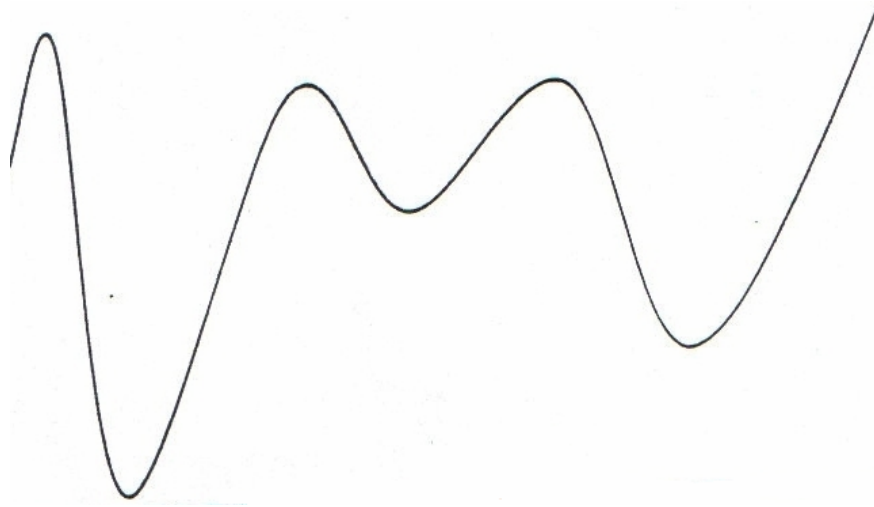
Geriye yayımlı algoritma, ağ yapısının hata fonksiyonu için eğim hesabına bakmakta ve sonrasında elde edilen doğrultuya ters yönde adım atarak bu işlemi tekrarlamaktadır. Bu durumda hata miktarı her seferinde biraz daha azalacaktır. Hata miktarını azaltacak doğrultuda hareket edildiği için kısa zamanda minimum hata düzeyine inilebilmektedir.

Algoritmada her bir seferde atılan mesafe “öğrenme katsayısı” olarak adlandırılmaktadır. Bu katsayının küçük seçilmesi durumunda yakınsama değeri dolayısı ile öğrenme son derece yavaşlayacaktır. Katsayının büyük seçilmesi durumunda ise büyük sıçramalar sonucunda öğrenme asla gerçekleşmeyecektir.

Geri yayımlı öğrenme algoritması temelde iki kusur barındırmaktadır:

- Hata fonksiyonunda en iyi değişim için kullanılan eğim terimi yerel bir göstergedir. Bu gösterge çok küçük uzaklık farklarında bile çok farklı doğrultuları gösterebilmektedir, Meydana gelen bu dalgalanmalar ise minimum hata seviyesine ulaşma yolunu uzatmaktadır,
- İkinci kusur, atılması gereken adım mesafesini hesaplamamanın son derece güç olmasıdır (Masters,1993).

Bu noktada yaşanacak problem yerel minimum değerlerinin global minimum değeri ile karıştırılmasıdır veya hatalı minimum değer saptamaktır. Gerçek hayattaki bir çok problem aşağıdaki şekilde verilen gibi bir kesit görüntüye sahiptir.



Şekil 3.4: Yerel Minimum Değerlere Saplanma Olasılığı Yüksek Bir Fonksiyonun Kesit Görünümü

Kesit üzerinde bir çok minimum değer olması sebebiyle global minimum değere ulaşma şansı başlangıç noktası seçimi ile ilgilidir. Yerel minimum noktalarından kaçınmak için kullanılabilecek ilk metot başlangıç noktası seçiminde yerel minimum değerlerden uzak bir noktadan başlamaktır. İkinci metot ise bulunan noktanın yerel minimum olup olmadığını anlayan bir süreç işletmektir. Bu amaçla sinir ağları için kullanılan Benzetimli Tavlama yöntemi bir seçenek olabilir. Diğer bir seçenek ise genetik algoritmalarıdır. Yapay sinir ağlarında kullanılan bu hibrit yöntemler genelde çözüm kalitesini artırmak amacıyla da kullanılırlar (Smith ve diğerleri, 1996).

Algoritma sekiz adımda özetlenebilir (Zurada,1992):

X , $(N \times 1)$, d ve y , $(M \times 1)$ boyutlu vektörler olmak üzere P örnekten oluşan $\{(x(1),d(1)), (x(2),d(2)), \dots, (x(P),d(P))\}$ eğitim kümesi verilir. c çevrim sayısını ve p de çevrimdeki adım sayısını göstermektedir.

Birinci adım: Öğrenme oranı α (>0) ve kabul edilebilir en büyük hata $E_{\max}$ seçilir. $W^1_{L \times N}$ ve $W^2_{M \times L}$ ağırlıklarına, $b^1_{L \times 1}$ ve $b^2_{M \times 1}$ eşiklerine rastgele küçük değerler atanır.

$$C \leftarrow 1, p \leftarrow 1, E \leftarrow 0$$

İkinci adım: Eğitim kısmı burada çalıştırılır ve girişler ağı uygulanır sonrasında tüm hücre çıkışları hesaplanır.

$$x \leftarrow x(p), d \leftarrow d(k) \quad (3.15)$$

$$z_j \leftarrow f^1(w_j^{1T} x + b_j^1) \quad (3.16)$$

$$y_j \leftarrow f^2(w_j^{2T} z + b_j^2) \quad (3.17)$$

Üçüncü adım: Hata hesaplanır.

$$E \leftarrow \frac{1}{2} \|d - y\|^2 + E \quad (3.18)$$

Dördüncü adım: Her iki katman için de hata işaret vektörleri $\delta^1_{L \times 1}$ ve $\delta^2_{M \times 1}$ hesaplanır.

$$\delta_j^2 = (d_j - y_j) f'^2(s_j^2) \quad (3.19)$$

$$\delta_q^1 = f'^1(s_q^1) \sum_{j=1}^M \delta_j^2 w_{jq}^2 \quad (3.20)$$

Beşinci adım: Çıkış katmanını için ağırlıklar ve eşikler ayarlanır.

$$w_{jq}^2 \leftarrow w_{jq}^2 + \alpha \delta_j^2 z_q \quad (3.21)$$

$$b_j^2 \leftarrow b_j^2 + \alpha \delta_j^2 \quad (3.22)$$

Altıncı adım: Gizli katmanın ağırlıkları ve eşikleri ayarlanır.

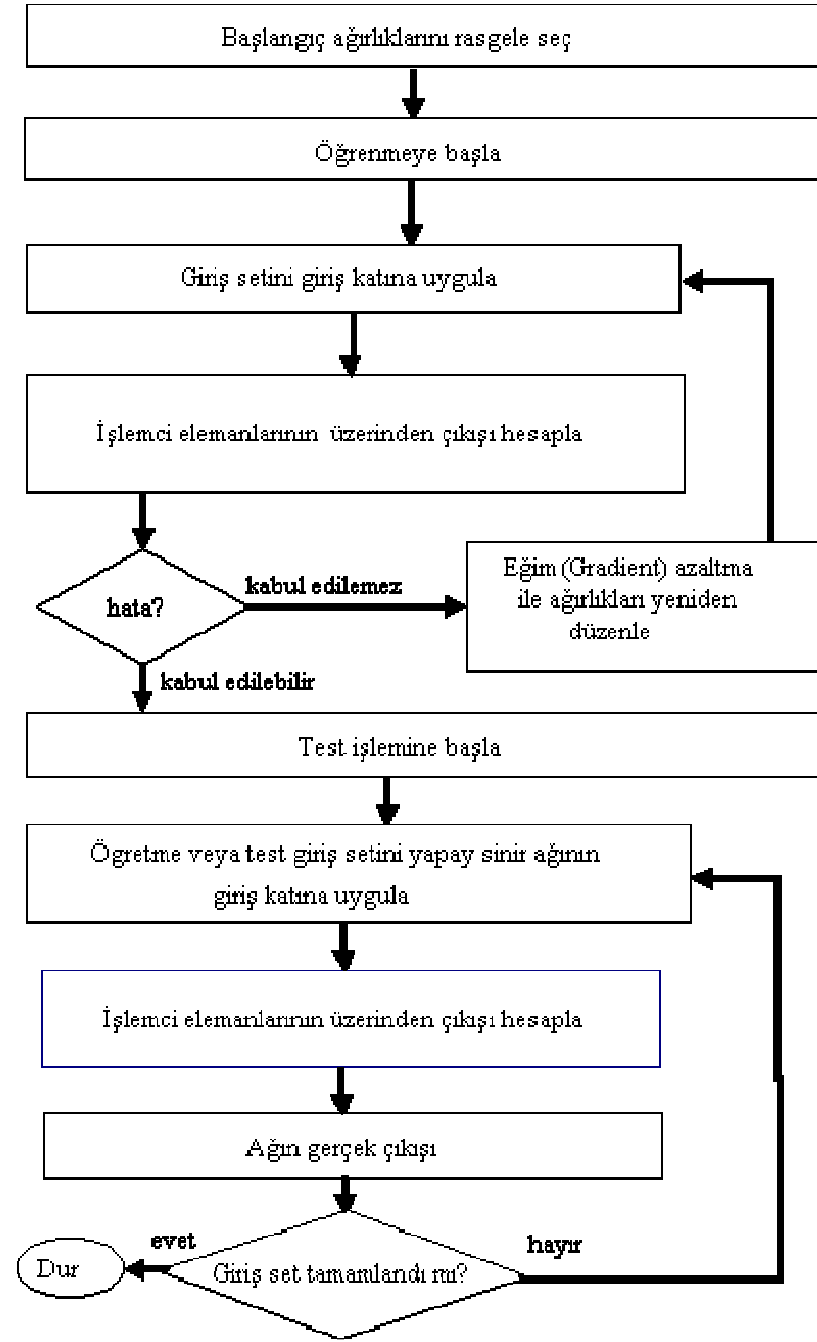
$$w_{qi}^1 \leftarrow w_{qi}^1 + \alpha \delta_q^1 x_i \quad (3.23)$$

$$b_q^1 \leftarrow b_q^1 + \alpha \delta_q^1 \quad (3.24)$$

Yedinci adım: $p < P$ ise $p \leftarrow p+1$, $c \leftarrow c+1$ yapılır ve ikinci adıma gidilir değilse sekizinci adıma geçilir.

Sekizinci adım: Eğitim çevrimi tamamlanmıştır. $E < E_{\max}$ ise eğitim bitmiştir.

Geri yayılım algoritması akış şeması Şekil 3.5’de verilmiştir.



Şekil 3.5: Çok katmanlı bir algılayıcıli geri yayılım akış şeması

Geri yayımlı algoritmada geliştirilmiş delta öğrenme kuralı kullanılmaktadır.

3.5. Delta Öğrenme Kuralı

Tek katmanlı genlikte sürekli algılayıcı tipli ağlar için kullanılmaktadır. Standart geri yayımlı algoritmada eğim bileşeni aşağıdaki gibidir.

$$\delta(k) = \frac{\partial E(k)}{\partial w(k)} \quad (3.25)$$

$E(k)$, k anındaki hata değeri

$w(k)$ bağlantı ağırlığı

$d(k)$ ağırlık değişiminin eğim bileşeni

Standart geri yayımlı algoritmada bağlantı ağırlığı,

$$w(k+1) = w(k) + \alpha \delta(k) \quad (3.26)$$

olarak güncelleştirilir. α sabit bir öğrenme oranıdır. Her bağlantı için değişken öğrenme oranı $\alpha(k)$ atanır ve bağlantı ağırlığı

$$w(k+1) = w(k) + \alpha(k) \delta(k) \quad (3.27)$$

şeklinde güncellenir. Jacobs, her bağlantıda öğrenme hızının azalması ve artması için sezgiselliğin gerçekleştirilmesinde, $d(k)$ eğim bileşeninin ağırlıklı ortalamasını kullanılmıştır. Bu ağırlıklı ortalama $\bar{\delta}(k)$ 'dir. Ağırlıklı ortalama aşağıdaki şekilde gösterilmektedir.

$$\bar{\delta}(k) = (1 - \theta) \delta(k) + \theta \bar{\delta}(k-1) \quad (3.28)$$

θ konveks ağırlık faktörüdür. Öğrenme k sabiti sezgisel açıdan, önceki eğim bileşeninin üstel artması ve şu anki eğim bileşeni aynı işaretli olması durumunda artan ağırlıklar ile birleştirilir. Öğrenme oranı mevcut eğim bileşeni üstel ortalamadan farklı işaretli ise, mevcut değeri ile orantılı olarak azalır. Öğrenme oranının güncelleştirilmesi aşağıda gösterilmiştir.

$$\Delta \alpha(k) = \begin{cases} \kappa & \bar{\delta}(k-1) \delta(k) > 0 \\ -\varphi \alpha(k) & \bar{\delta}(k-1) \delta(k) < 0 \\ 0 & \text{diğer} \end{cases} \quad (3.29)$$

k öğrenme katsayısı artma faktörü

j öğrenme katsayısı azaltma faktörü

a(k) k anındaki öğrenme oranı

Algoritma, öğrenme katsayılarını lineer olarak artırmakta, fakat geometrik olarak azaltmaktadır.

3.6. Genelleştirilmiş Delta Öğrenme Kuralı

Genelleştirilmiş delta öğrenme kuralı ile eğitim esnasında belirli bir anda “epoch” adı verilen hata hesaplanır. Bu kural ile hata düzeltilebilmektedir.

Genelleştirilmiş delta öğrenme kuralında belirli bir anda ölçülen hata bir önceki anda ölçülen hatadan büyük ise ağırlıklar bir önceki ağırlıklara geri döner. Hata düzeltme özelliğine eklenen dayanıklılık faktörü ile daha iyi sonuçlar elde edilebilmektedir. Bu ise her periyotta hata testi yapılması ihtiyacını ortadan kaldırarak n periyotta ölçümleme yapmanın yeterli olduğu bir yapı ortaya koymaktadır.

Genelleştirilmiş delta öğrenme kuralı ile delta öğrenme kuralları arasındaki farklar aşağıdaki gibi özetlenebilir.

- Sezgisel momentum kullanılır.
- Ağırlık uzayında büyük atlamalar ortadan kaldırılmıştır.
- Geometrik azaltma büyük (amacı aşan) atlamaları engellemede yeterince hızlıdır.
- Genelleştirilmiş delta öğrenme kuralında, her bir bağlantıdaki momentum hızı $m(k)$ ve öğrenme hızı $a(k)$ zamanla değişmektedir.
- Genelleştirilmiş delta öğrenme kuralı için öğrenme hızı değişim formülü aşağıdaki gibidir:

$$\Delta\alpha(k) = \begin{cases} \kappa_{\alpha} \exp\left\{-\gamma_{\alpha}|\bar{\delta}(k)|\right\} & \text{eğer } \bar{\delta}(k-1) \delta(k) > 0 \\ -\varphi_{\alpha} \alpha(k) & \text{eğer } \bar{\delta}(k-1) \delta(k) < 0 \\ 0 & \text{diğer durumda} \end{cases} \quad (3.30)$$

exp, üstel fonksiyonu

κ_{α} , sabit öğrenme hızı skala faktörü

φ_a , sabit öğrenme hızı azaltma faktörü

γ_a , sabit öğrenme hızı üstel faktörü.

k zamanındaki momentum hızı değişimi aşağıdaki ifade ile bulunabilir,

$$\Delta\mu(k) = \begin{cases} \kappa_\mu \exp\left[-\gamma_\mu|\bar{\delta}(k)|\right] & \text{eğer } \bar{\delta}(k-1)\delta(k) > 0 \\ -\varphi_\mu\mu(k) & \text{eğer } \bar{\delta}(k-1)\delta(k) < 0 \\ 0 & \text{diğer durumda} \end{cases} \quad (3.31)$$

κ_μ , sabit momentum hızı skala faktörü

φ_μ , sabit momentum hızı azaltma faktörü

γ_μ , sabit momentum hızı üstel faktörü

Öğrenme ve momentum hızları, azalmalarını ve artmalarını kontrol eden farklı sabitlere sahiptir. $\bar{\delta}(k)\delta(k)$ 'nin işareti bu amaçla kullanılmaktadır. Ağırlık uzayında meydana gelebilecek aşırı atlamaların engellenmesi için üst sınırlar her bir bağlantı hızı ve momentum hızı üzerinde belirlenir. Matematiksel olarak bu ifade aşağıdaki gibi verilebilir.

$$\alpha(k) \leq \alpha_{\max} \quad (3.32)$$

$$\mu(k) \leq \mu_{\max} \quad (3.33)$$

$\alpha_{\max}$ öğrenme katsayısının üst sınırı

$\mu_{\max}$ momentum katsayısının üst sınırı

3.7. Eğitim Zamanını Azaltan Yöntemler

Geriye yayılmalı algoritmanın eğitim sürecinin uzun süre alması sebebiyle pek çok çalışma yapılmış ve eğitim sürecini azaltan bazı yöntemler geliştirilmiştir. Bu yöntemlerden bazıları aşağıda açıklanmıştır.

3.7.1 Adaptif Öğrenme Oranı

k. çevrim için hata $E(k)$ bir önceki çevrimdeki hata $E(k-1)$ olarak belirtildiğinde, $E(k) < E(k-1)$ ise α belirli bir sabitle çarpılarak artırılır. $E(k)$, $E(k-1)$ 'in

daha önce belirlenen katından büyükse, yeni hesaplanan ağırlıklar ve hata iptal edilir ve α değeri belirli bir sabitle çarpılarak azaltılır.

Bu işlem ile birlikte ağıın eğitileceği α değeri ağ için uygun bir değer olur ve eğitim zamanı azaltılmış olur.

3.7.2. Daha İyi İlk Ağırlık Seçme Yöntemi

Rastgele değerlerden daha iyi bir ilk ağırlım değeri seçilmesi ile ilgili Nguyen ve Widrow çalışmışlardır. İki katmanlı bir ağıın bir fonksiyonu nasıl yaklaştırdığını analiz eden Nguyen ve Widrow bu amaçla bir yöntem geliştirmişlerdir. Yöntem, saklı katmandaki hücrelerin çıkış fonksiyonu sigmoid ve çıkış katmanındaki hücrelerin çıkış fonksiyonu lineer olan bir ağıın sadece ilk katman ağırlıkları için uygulanmaktadır. Çıkış katmanı ağırlıkları ise yine rastgele seçilmektedir.

Yöntem saklı katmanlarından sigmoid fonksiyon kullanılan iki katlı saklı katmanlı ağıın birinci saklı katmanı için de uygulandığında elde edilen sonuçlar başarılı olmaktadır.

Yöntemde öncelikle saklı katmanın ağırlık matrisi ve eşik vektörü elemanları $[-1,+1]$ aralığında rastgele olarak seçilir. Sonra ağırlık matrisinin satırları normalize edilir ve ağırlık matrisi ile eşik vektörü belli bir katsayı ile çarpılır. Bu katsayı, tanjant sigmoid fonksiyonu ($\lambda=2$) için 0,7, logaritmik sigmoid fonksiyonu ($\lambda=1$) için ise 2,8 olarak verilmektedir. $\lambda=1$ olan tanjant sigmoid fonksiyonu için ise bu katsayı 1,4'tür. Bu yöntem ile birlikte eğitim hızı 10 kata kadar artırılabilir.

3.7.3. Momentum Metodu

Momentum metodu ile birlikte eğitim zamanının azaltılmasının yanı sıra sık yerel minimumlardan da kurtulmak mümkündür (Demuth ve Beale, 1993) .

Momentum metodu uygulamasında bir andaki ağırlık değişimine daha önceki andaki ağırlık değişiminin belli bir oranı eklenir. Uygulamaya göre seçilen andaki ağırlık değişimi aşağıdaki şekilde yazılabilir.

$$\text{Veri uyarlamalı } \Delta w(k) = - \nabla E(k) + \beta \Delta w(k-1) \quad (3.34)$$

$$\text{Grup uyarlamalı } \Delta w(c) = - \nabla E(c) + \beta \Delta w(c-1) \quad (3.35)$$

β katsayısı genellikle 0,1 ile 0,9 arasında seçilen ve momentum oranı olarak adlandırılan bir sabittir. İfadelerdeki son ağırlık değişiminin oranını gösteren kısım ikinci kısımdır ve momentum terimi olarak adlandırılır.

4. KARAR AĞAÇLARI UYGULAMASI SONUÇLARI

4.1. Üniversite Öğrencilerinin Başarılarını Etkileyen Değişkenlerin Belirlenmesi ve Öğrencilerin Başarılarına Göre Sınıflandırılması

Yapay sinir ağları modeli üniversite öğrencilerinin başarılarını etkileyen değişkenlerin belirlenmesinde ve öğrencilerin başarılarına göre sınıflandırılmasında kullanılmıştır. Uygulama öncelikle karar ağaçları metodu kullanılarak Mayıs 2005 Döneminde Defining Critical Factors Affecting Student Success: A Data Mining Approach adlı tez çalışmasında gerçekleştirilmiştir. Bu çalışmada aynı veriler ile sonuçların yapay sinir ağları modellemesi sonucunda bulunması ve sonuçların karşılaştırılması gerçekleştirilecektir. Bu sayede karar ağaçları metodu ve yapay sinir ağları metodu aynı çalışmaya uygulanmış olacak ve üstünlükleri noktasında bir öngörü oluşturulmaya çalışılacaktır. Öncelikle ilk çalışmada kullanılan anket soruları ve ilk çalışma sonuçları kısaca aktarılacak ve sonrasında yapay sinir ağları uygulaması sonrası sonuçlar belirtilecektir. Son bölümde ise bu sonuçlar karşılaştırılacaktır.

İlk çalışma için İ.T.Ü İşletme Fakültesi öğrencilerine 2004-2005 öğretim yılı ilk dönem final sınavları esnasında anket uygulanmıştır. Ankete Endüstri Mühendisliği ve İşletme Mühendisliği öğrencileri dahil edilmiştir. 438 öğrencinin katılımı ile gerçekleştirilen anket sonuçları ayrı formlarda toplanmıştır. İlgili anket Ek-1’de yer almaktadır (Bölükbaşı, 2005).

Anket

Anket içerisinde dört kategoride sorular yer almaktadır. Demografik Özellikler, Eğitim, Akademik ve Kişisel Bilgiler kategorileri ile birlikte sırası ile

Demografik Özellikler: Öğrencilerin demografik özellikleri araştırılmıştır.

Eğitim: Üniversite öncesi başarı ölçümlenmiştir.

Akademik: Üniversiteye adaptasyon ve üniversite eğitimi sırasındaki başarı ölçümlenmiştir.

Kişisel Bilgiler: Üniversite eğitimi esnasında öğrencilerin yaşam tarzları araştırılmıştır (Bölükbaşı, 2005).

4.1.1. Demografik Özellikler:

Demografik özellikler çoğu kara ağaçları çalışmasında kullanılan ve önemli bir yer olan bölümdür. Burada öğrencilerin aileleri ve evdeki yaşam biçimleri önemli değişkenlerdir. Bunun yanında öğrencilerin başarısının çoğunlukla aile davranışları ile ilişkili olduğuna inanılır (Bölükbaşı, 2005).

Demografik özelliklerle ilgili sorular ve sonuçlar aşağıdaki gibidir.

Doğum tarihi

Öğrencilerin yaşlarını hesaplamak için sorulmuştur.

Tablo 4.1: Yaş Dağılımı

Yaş	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
19	2	0,46	2	0,46
20	55	12,56	57	13,01
21	120	27,40	177	40,41
22	140	31,96	317	72,37
23	100	22,83	417	95,21
24	15	3,42	432	98,63
25	5	1,14	437	99,77
26	1	0,23	438	100,00

Tablo 4.2: Yaş İstatistikleri Özeti

Konu	Maksimum	Ortalama	Minimum	N
Yaş	26	21,80	19	438

Cinsiyet

Cinsiyetin başarı üzerinde bir etkisinin olup olmadığının anlaşılması için sorulmuştur. İ.T.Ü. teknik bir üniversite olması sebebi ile ve çoğunlukla mühendislik ağırlıklı olduğu için öğrencilerin çoğu erkektir.

Tablo 4.3: Cinsiyet Dağılımı

Cinsiyet	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
Bayan	147	33,56	147	33,56
Bayan	291	66,44	438	100

Medeni Durum

Öğrencilerin çoğunun bekar olması beklenmektedir. Anket sonucunda alınan cevaplarda bu yöndedir.

Tablo 4.4:Medeni Durum Dağılımı

Evlilik Durumu	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
Evli	2	0,46	2	0,46
Bekar	436	99,54	438	100

Bu değişken çalışmada kullanılmamıştır. Anlamli bir dağılım belirtmediği için çalışma dışında tutularak aktif bir değişken olarak dikkate alınmamıştır.

Çocuk Sahipliği

Öğrencilerin çocuk sahibi olmadıkları bekleniyordu. Çalışma sonrası verilerde sadece bir öğrencinin çocuk sahibi olduğu belirlendi.

Tablo 4.5: Çocuk Sahipliği Dağılımı

Çocuk Sahipliği	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
Hayır	437	99,77	437	99,77
Evet	1	0,23	438	100

Anne Hayatta

Tüm öğrencilerin annelerinin hayatta olması sebebi ile bu değişken de çalışmaya dahil edilmemiştir.

Baba Hayatta

Öğrencilerin %8,68 'inin babasının hayatta olmaması sebebi ile bu değişkenin başarı için anlamlı olabileceği düşünülmüştür. Çalışma esnasında başarı üzerinde etkili olup olmayacağı belirlenecektir.

Tablo 4.6.Baba Hayatta Dağılımı

Baba Hayatta	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
Hayır	38	8,68	38	8,68
Evet	400	91,32	438	100

Doğum Yeri/Bölge

Ülkemizde bulunan yedi farklı bölgenin öğrencileri başarıları üzerinde etkili olup olmadığını belirlemek amacı ile dahil edilmiştir. Trafik kodları kullanılarak bölgeler tespit edilmiştir.

Tablo 4.7:Doğum Bölgesi Dağılımı

Bölge	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
Akdeniz	46	10,5	46	10,5
Doğu Anadolu	24	5,48	70	15,98
Ege	65	14,84	135	30,82
Güney Doğu Anadolu	12	2,74	147	33,56
Karadeniz	50	11,42	197	44,98
Marmara	148	33,79	345	78,77
İç Anadolu	66	15,07	411	93,84
Ülke Dışı	27	6,16	438	100

En Uzun Süre Yaşanılan Yer/Bölge

Öğrencilerin en uzun süre yaşadıkları bölgenin başarı üzerindeki etkisi olacağı düşünülmüştür. Trafik kodları yardımı ile cevaplar alınmıştır.

Tablo 4.8: En Uzun Süre Yaşanılan Yer/Bölge Dağılımı

En Uzun Süre Yaşanılan Bölge	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
Akdeniz	53	12,1	53	12,1
Doğu Anadolu	20	4,57	73	16,67
Ege	70	15,98	143	32,65
Güney Doğu Anadolu	7	1,6	150	34,25
Karadeniz	42	9,59	192	43,84
Marmara	187	42,69	379	86,53
İç Anadolu	52	11,87	431	98,4
Ülke Dışı	7	1,6	438	100

Lise Öğrenimi

Üniversite başarısı için lise öğreniminin önemli olduğu bilinmektedir. Bu sebeple lise eğitiminin alındığı bölge de önemli olarak ele alınmıştır.

Tablo 4.9:Lise Öğrenimi Dağılımı

Lise Öğrenimi	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
Akdeniz	56	12,79	56	12,79
Doğu Anadolu	16	3,65	72	16,44
Ege	73	16,67	145	33,11
Güney Doğu Anadolu	8	1,83	153	34,93
Karadeniz	40	9,13	193	44,06
Marmara	187	42,69	380	86,76
İç Anadolu	48	10,96	428	97,72
Ülke Dışı	10	2,28	438	100

Anne ve Baba Eğitim Düzeyi

Ebeveynlerin eğitim seviyesi ile çocuğun yetiştirilmesi arasında bir ilişki olduğuna inanılmaktadır. Özellikle geleneksel Türk ailelerinde çocuklar zamanının çoğunu anneleri ile geçirmektedirler. Bu yüzden anne eğitim düzeyinin baba eğitim düzeyinden daha etkili olduğu düşünülmektedir. Verilerin analiz edilmesi ile sonuçlar görülecektir.

Tablo 4.10: Anne Eğitim Düzeyi Dağılımı

Anne Eğitim Düzeyi	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
Üniversite	103	23,52	103	23,52
Doktora	1	0,23	104	23,74
Lise	126	28,77	230	52,51
Master	6	1,37	236	53,88
Tahsili Yok	21	4,79	257	58,68
İlkokul	110	25,11	367	83,79
Orta Okul	36	8,22	403	92,01
Ön Lisans	35	7,99	438	100,00

Tablo 4.11: Baba Eğitim Düzeyi Dağılımı

Baba Eğitim Düzeyi	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
Üniversite	191	43,61	191	43,61
Doktora	7	1,6	198	45,21
Lise	98	22,37	296	67,58
Master	17	3,88	313	71,46
Tahsili Yok	3	0,68	316	72,15
İlkokul	65	14,84	381	86,99
Orta Okul	26	5,94	407	92,92
Ön Lisans	31	7,08	438	100,00

Yaşça Büyük Erkek Kardeş veya Kız Kardeş Sayısı

Tüm yaşça büyük kardeş sayıları alındıktan sonra bu değerler ikili değerlere çevrilmek üzere evet veya hayır şeklinde düzenlenmiştir.

Tablo 4.12: Büyük Erkek Kardeş Sayısı Dağılımı

Büyük Erkek Kardeş Sayısı	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
0	329	75,11	329	75,11
1	87	19,86	416	94,98
2	18	4,11	434	99,09
3	3	0,68	437	99,77
5	1	0,23	438	100,00

Tablo 4.13: Büyük Erkek Kardeş Sahipliği Dağılımı

Büyük Erkek Kardeş Sahipliği	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
Hayır	329	75,11	329	75,11
Evet	109	24,89	438	100,00

Korelasyon ilişkisi sebebi ile sadece değiştirilen ikinci değerler çalışmaya dahil edilmiştir.

Tablo 4.14: Büyük Kız Kardeş Sayısı Dağılımı

Büyük Kız Kardeş Sayısı	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
0	287	65,53	287	65,53
1	112	25,57	399	91,1
2	28	6,39	427	97,49
3	8	1,83	435	99,32
4	1	0,23	436	99,54
5	1	0,23	437	99,97
6	1	0,23	438	100,00

Tablo 4.15: Büyük Kız Kardeş Sahipliği Dağılımı

Büyük Kız Kardeş Sahipliği	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
Hayır	287	65,53	287	65,53
Evet	151	34,47	438	100,00

Yaşça Küçük Erkek Kardeş veya Kız Kardeş Sayısı

Kişilerin küçükler ile olan iletişimlerinin büyükler ile olan iletişimlerinden farklılaşması sebebi ile çalışmaya dahil edilmiştir. Tüm yaşça büyük kardeş sayıları alındıktan sonra bu değerler binary değerlere çevrilmek üzere evet veya hayır şeklinde düzenlenmiştir.

Tablo 4.16: Küçük Erkek Kardeş Sayısı Dağılımı

Küçük Erkek Kardeş Sayısı	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
0	280	63,93	280	63,93
1	135	30,82	415	94,75
2	21	4,79	436	99,54
3	1	0,23	437	99,77
7	1	0,23	438	100,00

Tablo 4.17: Küçük Erkek Kardeş Sahipliği Dağılımı

Küçük Erkek Kardeş Sahipliği	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
Hayır	280	63,93	280	63,93
Evet	158	36,07	438	100,00

Korelasyon ilişkisi sebebi ile sadece değiştirilen ikinci değerler çalışmaya dahil edilmiştir.

Tablo 4.18: Küçük Kız Kardeş Sayısı Dağılımı

Küçük Kız Kardeş Sayısı	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
0	324	73,97	324	73,97
1	95	21,69	419	95,66
2	15	3,42	434	99,09
3	3	0,68	437	99,77
4	1	0,23	438	100,00

Tablo 4.19: Küçük Kız Kardeş Sahipliği Dağılımı

Küçük Kız Kardeş Sahipliği	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
Hayır	324	73,97	324	73,97
Evet	114	26,03	438	100,00

Tablo 4.20: Toplam Erkek Kardeş Sayısı Dağılımı

Toplam Erkek Kardeş Sayısı	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
0	191	43,61	191	43,61
1	190	43,38	381	86,99
2	45	10,27	426	97,26
3	8	1,83	434	99,09
4	2	0,46	436	99,54
5	1	0,23	437	99,77
7	1	0,23	438	100,00

Tablo 4.21: Erkek Kardeş Sahipliği Dağılımı

Erkek Kardeş Sahipliği	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
Hayır	191	43,61	191	43,61
Evet	247	56,39	438	100,00

Korelasyon ilişkisi sebebi ile sadece değiştirilen ikinci değerler çalışmaya dahil edilmiştir.

Tablo 4.22: Toplam Kız Kardeş Sayısı Dağılımı

Toplam Kız Kardeş Sayısı	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
0	191	43,61	191	43,61
1	180	40,87	371	84,47
2	50	11,42	421	95,89
3	10	2,28	431	98,17
4	5	1,14	436	99,32
5	2	0,46	438	100,00

Tablo 4.23: Kız Kardeş Sahipliği Dağılımı

Kız Kardeş Sahipliği	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
Hayır	191	43,61	191	43,61
Evet	247	56,39	438	100,00

Korelasyon ilişkisi sebebi ile sadece değiştirilen ikinci değerler çalışmaya dahil edilmiştir.

Tablo 4.24: Kardeş Sahibi Olma Dağılımı

Toplam Kardeş Sayısı	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
0	32	7,31	32	7,31
1	247	56,39	279	63,7
2	107	24,43	386	88,13
3	26	5,94	412	94,06
4	13	2,97	425	97,03
5	6	1,37	431	98,40
6	4	0,91	435	99,32
8	3	0,68	438	100,00

Tablo 4.25: Değiştirilmiş Toplam Kardeş Sayısı Dağılımı

Toplam Kardeş Sayısı	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
> 1	159	36,3	159	36,3
1	247	56,39	406	92,69
0	32	7,31	438	100,00

Uygulama esnasında toplam kardeş sayısı ile ilgili olarak değiştirilmiş bölüm kullanılacaktır.

Vatandaşlık

Soru içerisinde öğrencilere Türk olup olmadıkları sorulmuştur.

Tablo 4.26: Türk Olma/Olmama Dağılımı

Türk Olma/Olmama	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
Hayır	8	1,83	8	1,83
Evet	430	98,17	438	100,00

Yaklaşık olarak tüm öğrencilerin Türk olması sebebi ile bu değişkenin çalışmada olmasının anlamlı olmayacağı görülmüş ve değişken çalışmadan çıkarılmıştır.

Anne ve Baba Çalışma Durumu

Günümüzde çoğu anne baba ailelerine daha iyi bir gelecek hazırlamak için çalışmaktadır. Fakat bundan otuz yıl öncesinde annelerin çalışması erkekler tarafından uygun görülen bir durum değildi. Anne ve babaların çalışma durumlarının öğrencilerin başarılarındaki etkisini ölçmek amacı ile bu soru dahil edilmiştir.

Tablo 4.27: Çalışan Anne Dağılımı

Anne Çalışma Durumu	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
Hayır	225	51,37	225	51,37
Evet	213	48,63	438	100,00

Cevaplara göre yaklaşık olarak annelerin yarısı çalışmaktadır. Bu değişken çalışmada kullanılacaktır.

Tablo 4.28: Çalışan Anne Dağılımı-2

Baba Çalışma Durumu	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
Hayır	17	3,88	17	3,88
Evet	421	96,12	438	100,00

Çalışan babalarla ilgili alınan hayır yanıtlarının babaların daha önce hiç çalışmadığı veya şu an çalışıp çalışmadıkları ile ilgili bir tereddüt oluşturduğu gözlemlenmiştir. Sonuçlar çalışma durumu ile ilgili hedefi göstermemektedir. Bazı deneklerin her iki cevabı da işaretledikleri belirlenmiştir. Bu sebeplerle sonuçların uygulamada kullanılmamasına karar verilmiştir.

Aile Durumu

Aile yaşamının öğrenciler üzerinde etkili olduğu düşünüldüğü için eklenmiş ve analiz edilmek istenmiştir.

Tablo 4.29: Aile Durumu Dağılımı

Aile Durumu	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
Boşanmış	9	2,24	9	2,24
Beraber	386	96,26	395	98,50
Ayrı	6	1,5	401	100,00

Ankete katılan 37 öğrenci bu soruyu cevaplamamıştır. Genellikle babası hayatta olmaya öğrencilerin bu soru için cevap verdiği gözlemlenmiştir. 37 kişi için farklı bir grup oluşturulsa dahi yine de bu sorunun çalışma kapsamında kullanılması uygun bulunmamıştır.

4.1.2. Eğitim Soruları (Üniversite Öncesi)

Öğrencilerin üniversite öncesi geçmişini ve başarısını araştırmak için oluşturulmuştur (Bölükbaşı, 2005).

Lise Tipi

Öğrencilere öğrenim gördükleri lise tipi sorulmuştur.

Tablo 4.30: Lise Tipi Dağılımı

Lise Tipi	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
Yurtdışı	1	0,23	1	0,23
Anadolu Lisesi	217	49,54	218	49,77
Özel	35	7,99	253	57,76
Ticaret Lisesi	2	0,46	255	58,22
Fen	112	25,57	367	83,79
Devlet	71	16,21	438	100,00

Genellikle başarılı okullar arasında Anadolu Liseleri ve Fen Liselerinin olduğu bilinmektedir. Cevaplar incelendiğinde de öğrencilerin yaklaşık %50 sinin Anadolu Lisesinden ve %25'inin de Fen Lisesinden mezun olduğu görülmektedir.

Lise Mezuniyet Notu

Lise mezuniyet notunun üniversite başarısında belirleyici bir değişken olduğu beklenmektedir.

Tablo 4.31: Lise Not İstatistikleri

	Maksimum	Ortalama	Minimum	N
Lise Notu	5	4,803	3,45	437

Üniversiteye Giriş

Bu noktada öğrencilere İ.T.Ü'ye ilk üniversite sınavları sonrasında girip girmedikleri sorulmuştur. Bu değerler uygulamada ele alınmıştır.

Tablo 4.32: Üniversiteye İlk Sınavda Giriş Dağılımı

Üniversiteye İlk Sınavda Giriş	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
Hayır	30	6,85	30	6,85
Evet	408	93,15	438	100,00

Bölüm Tercih Sırası

Öğrencilerin üniversite tercihlerinde kazandıkları bölümün kaçınıcı sırada olduđu sorulmuştur.

Tablo 4.33: Bölüm Tercih Sırası Dağılımı

Bölüm Tercih	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
1	35	8,31	35	8,31
2	47	11,16	82	19,48
3	71	16,86	153	36,34
4	72	17,1	225	53,44
5	55	13,06	280	66,51
6	47	11,16	327	77,67
7	31	7,36	358	85,04
8	16	3,8	374	88,84
9	19	4,51	393	93,35
10	9	2,14	402	95,49
11	10	2,38	412	97,86
12	3	0,71	415	98,57
13	3	0,71	418	99,29
14	3	0,71	421	100

Öğrencilerden 17 tanesi bu soruya cevap vermemiştir. Sadece %8,31lik bir kısım bölümü ilk tercihlerinde yazmışlardır. Bazı öğrenciler diğeri üniversitelerden yatay geçiş yaparak bölüme gelmiş, bazıları da bölümü ikinci bir alan olarak seçmişlerdir. Bu veriler aşağıdaki gibi şekillendirilerek çalışmaya dahil edilmiştir.

Tablo 4.34: Bölüm Tercih Sırası Dağılımı-2

Bölüm Tercih Sırası	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
01.Şub	82	18,72	82	18,72
03.Nis	143	32,65	225	51,37
05.Haz	102	23,29	327	74,66
>=7	94	21,46	421	96,12
Yatay	17	3,88	438	100,00

4.1.3. Akademik Sorular (Üniversite Öğrenimi Sırasında)

Öğrencilerin üniversite öğrenimi esnasındaki akademik bilgilerini almak için sorular sorulmuştur (Bükbaşı, 2005) .

Kütüphane Kullanımı

Üniversite eğitimi esnasında bir çok ödev ve projenin yapılabilmesi için internet ve kütüphane kullanımı ile araştırmalar gerçekleştirilmektedir. Kütüphanelerde internet bağlantısının olması da bu kullanımda daha etkili olmaktadır. Öğrencilerin bu araştırmalarının başarıları üzerinde olumlu etkiye sahip olduğu beklenmektedir. Öğrencilere kütüphaneleri kullanıp kullanmadıkları sorulmuş ve aşağıdaki cevaplar elde edilmiştir.

Tablo 4.35: Kütüphane Kullanımı Dağılımı

Kütüphane Kullanımı	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
Hayır	35	7,99	35	7,99
Evet	403	92,01	438	100,00

Akademik Olmayan Kitap/Dergi Okuma İlgisi

Öğrencilere akademik olmayan yayınlara karşı ilgileri sorulmuş ve sosyal hayat ile bağlantıları da test edilmiştir. Öğrencilerin dergiler, gazeteler vb. akademik olmayan yayınları da takip etmesi beklenmektedir.

Tablo 4.36: Akademik Olmayan Yayın Okuma Dağılımı

Kütüphane Kullanımı	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
Asla	3	0,68	3	0,68
Nadiren	41	9,36	44	10,05
Bazen	185	42,24	329	52,28
Genellikle	209	47,72	438	100,00

Akademik Yayın Okuma İlgisi

Öğrencilerin akademik yayınları okuyup okumadıkları sorulmuş ve aşağıda görüleceği gibi öğrencilerin yarı yarıya farklı davranışlar sergiledikleri görülmüştür. Bu değişkenin başarı üzerinde etkili olup olmadığı çalışma sonrasında görülecektir.

Tablo 4.37: Akademik Yayın Okuma Dağılımı

Akademik Yayın İlgisi	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
Hayır	230	52,51	230	52,51
Evet	208	47,49	438	100,00

Konferans Katılımı

Öğrencilere konferans veya sempozyumlara katılıp katılmadıkları sorulmuştur.

Tablo 4.38: Konferans Katılımı Dağılımı

Konferans Katılımı	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
Hayır	92	21,00	92	21,00
Evet	346	79,00	438	100,00

Öğrencilerin %79'unun konferanslara katıldıkları görülmüştür.

Yüksek Lisans Kararı

Öğrencilere dört yıllık eğitim sonrasında yüksek lisans eğitimine katılmayı düşünüp düşünmedikleri sorulmuştur. Öğrencileri %45'i katılacakları yönünde kararlarını belirtmişlerdir.

Tablo 4.39: Yüksek Lisans Kararı Dağılımı

Yüksek Lisans Kararı	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
Hayır	74	16,89	74	16,89
Karar vermemiş	164	37,44	238	54,34
Evet	200	45,66	438	100,00

Not Ortalaması

Çalışmanın ana amacı öğrencilerin başarılarını etkileyen değişkenlerin belirlenmesidir. Bu sebeple başarılı ve başarısız öğrenciler şeklinde bir ayırım yapılmış ve bunun için not ortalaması 3,00'ın üzerindeki öğrenciler başarılı ve diğerleri de başarısız olarak kabul edilmiştir. Bu sayede başarıyı etkileyen değişkenler için çizilecek olan yol belirlenmiş olacak ve bu değerler çalışmada ana değişken olarak kullanılacaktır.

Öğrencilerden alınan not ortalamalarına göre belirlenen sınıflara ait veriler aşağıdaki gibidir.

Tablo 4.40: Not Ortalaması Dağılımı

Not Ortalaması	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
Başarılı	224	52,83	224	52,83
Başarısız	200	46,93	424	100,00

Ankette bu soruya cevap vermeyen 14 öğrenci tespit edilmiş ve bu öğrenciler çalışma dışında tutulmuştur.

Tahmini Mezuniyet Not Ortalaması

Öğrencilere mezuniyet zamanında beklemiş oldukları not ortalaması tahminleri sorulmuştur. Alınan sonuçlarda öğrencilerin mevcut ortalamaları ile yaklaşık değerler bekledikleri gözlemlenmiş ve bu sebeple bu değişken çalışma dışında tutulmuştur.

Okul Projeleri

Öğrencilere okul projeleri veya ödevlerinin akademik hayat ve profesyonel hayat içi katkısı olup olmadığı sorulmuştur. Öğrencilerden %77,85'i proje veya ödevlerin katkısının olduğunu belirtmiştir.

Tablo 4.41: Okul Projeleri Katkısı Dağılımı

Proje Katkısı	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
Yok	59	13,47	59	13,47
Kararsız	38	8,68	97	22,15
Var	341	77,85	438	100,00

Devamlılık Yüzdesi

İ.T.Ü.'nde derslerden geçebilmek için en az %70 derse katılım gereklidir. Öğrencilere bu kapsamda derslere katılım yüzdeleri sorulmuştur ve başarı zerinde katılım yüzdesinin etkisi gözlemlenmek istenmiştir.

Tablo 4.42: Devamlılık Yüzdesi Dağılımı

Devamlılık Yüzdesi	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
70	165	37,67	165	37,67
80	94	21,46	259	59,13
90	138	31,51	397	90,64
100	41	9,36	438	100,00

4.1.4 Kişisel Sorular

Anketin kişisel sorular bölümünde öğrencilerin hayat tarzları, hobileri, finansal durumları ve günlük yaşamları ile ilgili bilgiler yer almaktadır.

Yarı Zamanlı Çalışma (Part Time İş)

Öğrencilerin çalışmalarının başarıları üzerindeki etkisini gözlemlemek amacıyla sorulmuştur.

Tablo 4.43: Yarı Zamanlı Çalışma Dağılımı

Yarı Zamanlı Çalışma	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
Hayır	337	76,94	337	76,94
Evet	101	23,06	438	100,00

Sinemaya Gitme Sıklığı

Sinemaya gitmek üniversite öğrencilerinin en popüler hobilerinden birisidir. Öğrencilerin hobileri kapsamında sinema alışkanlıkları da incelenmiştir.

Tablo 4.44: Sinemaya Gitme Sıklığı Dağılımı

Sinemaya gitme	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
Haftada Birden Fazla	9	2,05	9	2,05
Asla	49	11,19	58	13,24
Ayda Bir	312	71,23	370	84,47
Haftada Bir	68	15,53	438	100,00

Spor Yapma Alışkanlığı

Öğrencilerden bu soru sonrasında alınan yanıtlara göre öğrencilerin %56,39'unun spor yaptıkları belirlenmiştir.

Tablo 4.45: Spor Yapma Dağılımı

Spor Yapma	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
Hayır	191	43,61	191	43,61
Evet	247	56,39	438	100,00

Hobi/Kişisel Gelişim Kursu

Öğrencilere herhangi bir hobi veya kişisel gelişim amacıyla kursa katılımlarının olup olmadığı sorulmuştur.

Tablo 4.46: Hobi/kişisel Gelişim Kursu Dağılımı

Kursa Katılım	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
Hayır	293	66,89	293	66,89
Evet	145	33,11	438	100,00

Ortalama İnternet Kullanımı

Öğrenciler için öğrenimleri boyunca internet bir bilgi kaynağıdır. Neredeyse tüm üniversitelerin laboratuvarlarında bilgisayarların internet bağlantısı vardır. Fakat internet kullanımı her zaman bilgi amaçlı olmayabilir.

Anket içerisinde öğrencilerin günde ne kadar süre internet kullandıkları sorulmuştur.

Tablo 4.47: Günlük İnternet Kullanım Saati Dağılımı

Günlük İnternet Kullanım Saati	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
2-4	139	31,74	139	31,74
<2	216	49,32	355	81,05
>4	83	18,95	438	100,00

Ortalama Televizyon İzleme Süresi

Ülkemizde televizyon izlemek insanların hobileri arasındadır. Fakat öğrenim süresince bu hobi olumsuz etkiye sahip olabilir. Bu sebeple öğrencilere günde ortalama kaç saat televizyon izledikleri sorulmuştur.

Tablo 4.48: Ortalama Televizyon İzleme Süresi Dağılımı

Ortalama Televizyon İzleme Süresi	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
2-4	64	14,61	64	14,61
<2	368	84,02	432	98,63
>4	6	1,37	438	100,00

Ortalama Çalışma Süresi

Öğrencilere günlük ortalama kaç saat ders çalıştıkları sorulmuştur. Öğrencilerden sadece %2,5'i günde 4 saatten daha fazla ders çalışmaktadır.

Tablo 4.49: Ortalama Çalışma Süresi Dağılımı

Ortalama Çalışma Süresi	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
2-4	113	25,8	113	25,8
<2	314	71,69	427	97,49
>4	11	2,51	438	100,00

Üniversitedeki Arkadaşlarla Görüşme

Bu soru ile öğrencilerin sosyalleşme seviyeleri ve bunun başarı üzerindeki etkisi ölçümlenmeye çalışılmıştır.

Çoğu öğrencinin üniversiteden arkadaşları ile görüştükleri sadece %5,25'lik bir dilimin görüşmediği belirlenmiştir.

Tablo 4.50: Üniversitedeki Arkadaşlarla Görüşme Dağılımı

Üniversitedeki Arkadaşlarla Görüşme	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
Hayır	23	5,25	23	5,25
Evet	415	94,75	438	100,00

Üniversite Kulüplerine Katılım

Öğrencilerin üniversitede yer alan kulüplere üye olup olmadıkları sorulmuştur. %31,74'lük dilimin kulüp faaliyetlerine katıldıkları tespit edilmiştir.

Tablo 4.51: Kulüplere Katılım Dağılımı

Kulüplere Katılım	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
Hayır	299	68,26	299	68,26
Evet	139	31,74	438	100,00

İstanbul'da İkametgah Yeri

Öğrencilerin yaşadıkları ortam başarıları üzerinde etkili olabilir. Bu sebeple öğrencilere sorulan soruya %41,78 oranında öğrenci bir oda arkadaşı ile yaşadığı şeklinde cevap vermiştir.

Tablo 4.52: İstanbul’da İkametgah Yeri Dağılımı

İstanbul’da İkametgah Yeri	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
Evde (Tek Başına)	12	2,74	12	2,74
Yurtta	121	27,63	133	30,37
Kendi Ailesi İle	92	21,00	225	51,37
Evde (Arkadaşı İle)	183	41,78	408	93,15
Aile Yanında	30	6,85	438	100,00

Okula Ulaşım Şekli

Öğrencilere okula gelmek için kullanmış oldukları araç sorulmuş ve %79 oranında öğrencinin okula toplu taşıma araçları kullanarak geldiği görülmüştür.

Tablo 4.53: Okula Ulaşım Şekli Dağılımı

Okula Ulaşım Şekli	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
Toplu Taşıma	346	79	346	79
Özel Araç	24	5,48	370	84,47
Yürüme	68	15,53	438	100,00

Ortalama Ulaşım Süresi

Öğrencilerin okula gelmek için yolda geçirdikleri süre sorulmuştur. Öğrencilerin %24,66’sının yolda bir saatten fazla süre harcadıkları belirlenmiştir.

Tablo 4.54: Ortalama Ulaşım Süresi Dağılımı

Ortalama Ulaşım Süresi	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
15-30 dk.	79	18,04	79	18,04
30-60 dk.	203	46,35	282	64,38
<15 dk.	48	10,96	330	75,34
>60 dk.	108	24,66	438	100,00

Çalışma Çevresi

Çoğu öğrenci için ders çalışırken konsantre olabilmek ve bu anlamda çevre koşulları çok önemlidir. Öğrencilere bu kapsamda çalışma çevreleri ile ilgili memnuniyet dereceleri sorulmuş ve aşağıdaki cevaplar alınmıştır.

Tablo 4.55: Çalışma Çevresi Memnuniyet Dağılımı

Çalışma Çevresi Memnuniyeti	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
Memnun Değil	61	13,93	61	13,93
Memnun	277	63,24	338	77,17
Çok Memnun	100	22,83	438	100,00

Bilgisayar Sahipliği

Öğrenim süresinde bir çok proje ve ödevin yapılması için bilgisayar kullanılması gerekmektedir. Ayrıca internet erişimi içinde bilgisayar ihtiyacı vardır. Okullardaki laboratuvarlarda bilgisayarlar olmasına rağmen kişisel bir bilgisayara sahip olmak daha fonksiyoneldir.

Tablo 4.56: Kişisel Bilgisayar Dağılımı

Kişisel Bilgisayar	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
Hayır	57	13,01	57	13,01
Evet	381	86,99	438	100,00

Aylık Ortalama Harcama

Öğrencilerin aylık ortalama gelirleri harcama düzeyleri ile ilgilidir. Bu sebeple öğrencilere aylık ortalama gelirleri sorulmuş ve öğrencilerin çoğunun 200-400 YTL arasında gelir elde ettikleri gözlemlenmiştir.

Tablo 4.57: Aylık Ortalama Gelir Dağılımı

Aylık Ortalama Gelir	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
200-400	194	44,29	194	44,29
400-600	140	31,96	334	76,26
<200	46	10,50	380	86,76
>600	58	13,24	438	100,00

Bursluluk

Üniversitede öğrencileri maddi anlamda da desteklemek amacıyla farklı burslar yer almaktadır. Öğrencilere herhangi bir burs alıp almadıkları sorulmuştur.

Tablo 4.58: Bursluluk Dağılımı

Bursluluk	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
Hayır	144	32,88	144	32,88
Evet	294	67,12	438	100,00

Öğrenim Kredisi

Ülkemizde devlet okullarında da öğrenim için katkı ödenmektedir. Öğrencilerin bir kısmı bunu öğrenimler sırasında kredi olarak almakta ve öğrenimleri sonrasında ödemelerini gerçekleştirmektedirler.

Tablo 4.59: Öğrenim Kredisi Kullanma Dağılımı

Öğrenim Kredisi Kullanma	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
Hayır	141	32,19	141	32,19
Evet	297	67,81	438	100,00

İş Bulma

İşsizlik oranının yüksek olması ve öğrencilerin bu anlamda beklentileri gelecekleri adına bazı şekillenmelerde etkili olmaktadır. Öğrencilere mezuniyetleri sonrasında istedikleri zamanda iş bulup bulamayacakları sorulmuştur.

Tablo 4.60: İş Bulabilme Dağılımı

İş Bulabilme	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
Hayır	112	25,57	112	25,57
Evet	326	74,43	438	100,00

İş Bulma Süresi

Öğrencilere mezuniyetleri sonrasında iş bulmak için yeterli olacak süre sorulmuştur. Öğrencilerin %67'sinin altı aylık bir sürede iş bulmayı bekledikleri gözlemlenmiştir.

Tablo 4.61: İş Bulma Süresi Dağılımı

İş Bulma Süresi	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
12 ay	70	15,98	70	15,98
6 ay	216	49,32	286	65,30
Hemen	77	17,58	363	82,88
Fikri Yok	75	17,12	438	100,00

İ.T.Ü. Sosyal Çevresi

Öğrencilere üniversitenin sosyal çevresi ile ilgili sorular sorulmuş ve öğrencilerin %57,76'sı sosyal çevre ile ilgili görüşlerinin kötü olduğunu belirtmiştir.

Tablo 4.62: İ.T.Ü. Sosyal Çevre Görüş Dağılımı

İ.T.Ü. Sosyal Çevre	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
Kötü	253	57,76	253	57,76
İyi	15	3,42	268	61,19
Orta	170	38,81	438	100,00

En Çok Sevilen Ders Grubu

Öğrencilere en çok sevdikleri ders grubu sorulmuştur. Alınan cevaplarda % 42,66'lık çoğunluk Mesleki Tasarım grubunu sevdiğini belirtmiştir.

Tablo 4.63: En Çok Sevilen Ders Grubu Dağılımı

En Çok Sevilen Ders Grubu	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
Mesleki Tasarım	183	42,66	183	42,66
Temel Mühendislik	98	22,84	281	65,50
İnsan ve Toplum Bilimleri	29	6,76	310	72,26
Temel Bilimler	119	27,74	429	100,00

İ.T.Ü. 'de Olmaktan Duyulan Memnuniyet

Öğrencilere İ.T.Ü'lü olmaktan memnun olup olmadıkları sorulmuş ve %70,55'lik bir çoğunluğun memnun olduğu belirlenmiştir.

Tablo 4.64: Memnuniyet Dağılımı

Memnuniyet	Sıklık	Yüzde	Kümülatif Sıklık	Kümülatif Yüzde
Memnun Değil	58	13,24	58	13,24
Memnun	309	70,55	367	83,79
Çok Memnun	71	16,21	438	100,00

4.2. Karar Ağaçları Uygulaması

Karar ağaçları uygulaması için kümelemesi analizine imkan veren SAS Enterprise Miner programı kullanılmıştır.

Öğrencilerin 2004-2005 öğretim yılı sonucundaki ortalama notları ile başarılı ve başarısız olarak veriler şekillendirilmiştir. Öğrencilerin notları 3'ün üzerinde ise başarılı olarak kabul edilmişler ve diğerleri de başarısız olarak sınıflandırılmışlardır. Öğrencinin başarılı olması durumu "S" ile başarısız olması durumu ise "US" ile belirtilmiştir. Öğrencilerin başarılı olmaları tahmin edilmeye çalışıldığı için hedef olay olarak "S" belirlenmiştir.

Çalışma için oluşturulan ankette 52 adet soru yer almaktadır ve bu sorular için yaklaşık 15 adet yeni cevap şekli oluşturulmuştur. Bazı sorular çalışmaya yeni cevap şekilleri ile alınmış bazıları da çalışma için gereksiz görülerek çalışmaya dahil edilmemiştir. (Bölükbaşı, 2005)

Uygulamaya dahil edilen değişkenler aşağıdaki gibidir. Değişkenlerin Türkçe isim karşılıkları ekler bölümünde yer almaktadır.

Tablo 4.65: Çalışma Verileri

Department	X_Smaller_Sister	Conferance_Involvement	School_Clubs
Class	X_Total_Number_of_Brothers	Master_Degree	Residence
Gender	X_Total_Number_of_Sisters	Project_Aid	School_Arrival
Father_Alive	X_Total_Number_of_Sibling	Attendance_Perc	Arrival_Duration
Birth_Region	Mother_Working	Parttime_Working	Studying_Environment
Live_Most_Region	High_School_Type	Cinema_Frequency	Special_Computer_Usage
High_School_Region	High_School_GPA2	Sports	Average_Monthly_Expenditure
Mother_Edu	Uni_Entrance_Fisrt	Hobby_Course	Scholarship
Father_Edu	X_Order_of_Department_Selection	Internet_Hours	Fee_Credit
X_Bigger_Brother	Library_Usage	TV_Hours	Finding_Job
X_Bigger_Sister	Nonacedemic_Interest	Studying_Hours	Find_Job_Duration
X_Smaller_Brother	Acedemic_Interest	Meeting_Uni_Friends	ITU_Social_Environment
Most_Liked_Course_Group	General_Pleasure_of_ITU		

Uygulamada ankette not ortalamasını belirtmiş olan 424 öğrenci incelenmiştir. 14 öğrenci not ortalamasını belirtmemiştir. 424 öğrencinin 224'ü başarısız ve 200'ü başarılıdır. Başarı oranı ise %47'dir.

Model sonucunda 216 öğrenci başarısız bulunmuştur. Bu öğrencilerden 189'u başlangıç verilerinde de başarısız olarak görülmektedir. Fakat başlangıçta başarılı olan 27 öğrenci model sonucunda başarısız olarak belirlenmiştir.

Model sonucunda 208 öğrenci ise başarılı olarak belirlenmiştir. Bunlardan 173 tanesi ilk baştaki başarılı öğrenciler olarak belirlene grup içerisinde. Fakat 35 öğrenci başarısız oldukları halde başarılı olarak belirlenmişlerdir.

Toplam model hatası $(27+35)/(424)= 0,14$ 'tür. Model %86 oranında başarıya ulaşmıştır. Model sonuçlarına göre başarıyı etkileyen değişkenler aşağıdaki gibidir (Bölükbaşı, 2005).

Tablo 4.66: Başarıyı Etkileyen Değişkenler ve Önem Dereceleri

Değişken İsmi	Önem
Devamlılık Yüzdesi	1,0000
Cinsiyet	0,6192
İstanbul'da İkametgah Yeri	0,5022
En Çok Sevilen Ders Grubu	0,4518
İş Bulma Süresi	0,3755
Doğum Yeri/Bölge	0,3745
Sınıf	0,3264
Bursluluk	0,3056
Baba Eğitim Düzeyi	0,2792
Ortalama Çalışma Süresi	0,2687
İ.T.Ü. 'de Olmaktan Duyulan Memnuniyet	0,2546
Yüksek Lisans Kararı	0,2403
Baba Hayatta	0,2362
Sinemaya Gitme Sıklığı	0,2204
Aylık Ortalama Harcama	0,1976

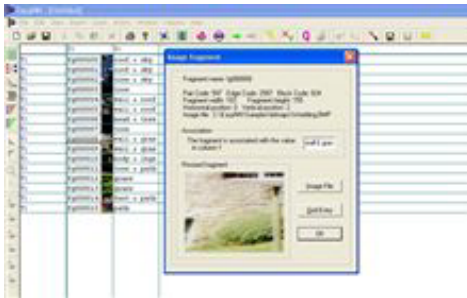
5. YAPAY SİNİR AĞLARI UYGULAMASINDA KULLANILAN YAZILIMLAR

Yapay sinir ağlarındaki tüm gelişmeler bilgisayar bilimindeki gelişmelere paralel olarak yürümektedir. Tek bir hücrenin yapısının ve hareketinin basit olmasına rağmen genel yapı ve hesaplamalar oldukça karmaşıktır. Geri yayımlı öğrenme algoritması veya gözetimsiz öğrenme algoritmalarının bilgisayar desteği olmadan yapılması mümkün değildir. Bu sebeple yapay sinir ağları uygulamalarında bilgisayar desteği gereklidir (Gately,1996).

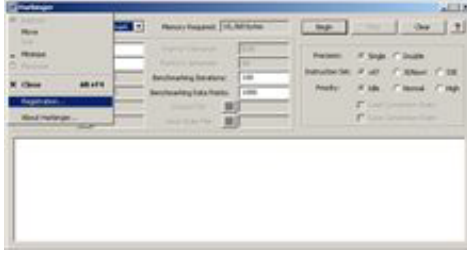
Yapay sinir ağları uygulamaları için kullanılan yazılım programlarının bazıları aşağıda yer almaktadır: (<http://www.filesland.com/software/neural-network-1.html>)



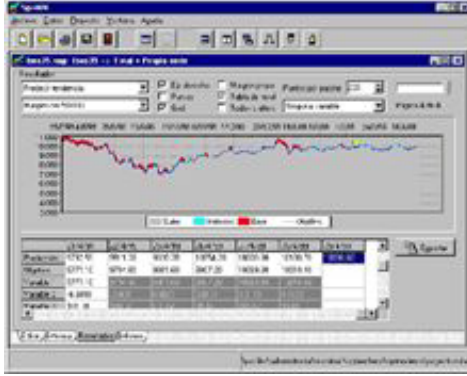
GoldenGem: Genel amaçlı bir Yapay Sinir Ağı programıdır.



EasyNN Plus: Microsoft Windows sistemleri için hazırlanmış bir yazılımdır. Nesnelerden, text dosyalarından veya tablolardan çok katmanlı hücre oluşturulabilmektedir. Eğitim, doğrulama ve test dosyaları program aracılığı ile de hazırlanabilir.



de kullanılmaktadır.



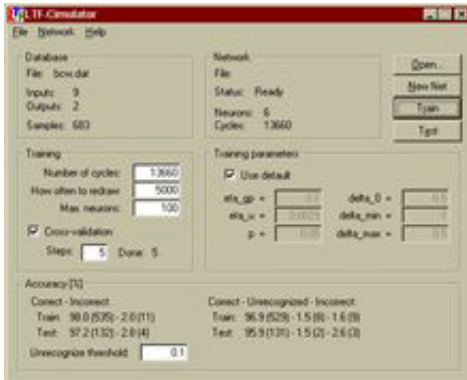
görmek mümkündür.



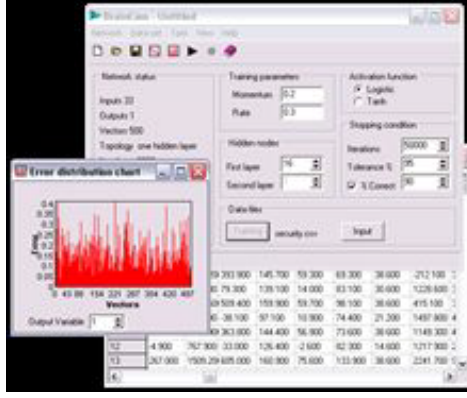
Harbinger: Windows tabanlı zaman serileri ve karşılaştırma uygulamalarında kullanılmaktadır. Model davranışları için sinir ağları yapısı ile çalışmaktadır. Ayrıca farklı bilgisayar donanım platformlarının performansını hesaplamak için

SprinN Lite eng: Pazar tahminler için kullanılan bir yazılımdır. Tahmin araçlarında yapay sinir ağları metotları ile hesaplamalar yapılmaktadır. Yapılan işlemlerin risklerini, komisyonları, teknik analiz değişkenlerinin etkilerini detaylı raporlarla ve grafiklerle

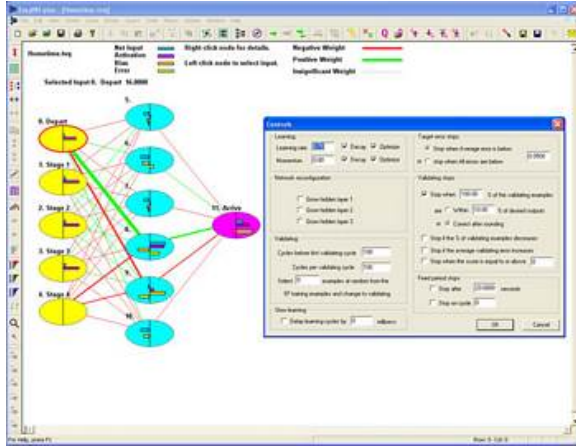
TradingSolutions: Yatırım ve finansman konularındaki geleneksel finansal analiz metotlarını yapay sinir ağları teknikleri ile gerçekleştirmektedir. Kendi içerisinde daha önceden oluşturulmuş 10 adet sinir ağı modeli yer almaktadır.



LTF- Cimulator: Simülasyon oluşturmak için kullanılmaktadır. Hazırlanacak modellerin eğitim ve doğrulama işlemleri gerçekleştirilebilmektedir. Eğitim parametreleri otomatik olarak seçilebilmektedir. Sonuçlar text formatında kaydedilebilir.



BrainCom: Geriye yayımlı algoritmayı kullanan bir yapay sinir ağı modelidir. Tahmin yöntemleri, nesne tanımlama, sınıflandırma, karar alma, problem optimizasyonu, zaman serisi analizleri gibi bir çok alanda kullanılabilmektedir.



EasyNN: Microsoft Windows için hazırlanmış bir yapay sinir ağı yazılımıdır. Text dosyaları ile veri aktarımı yapılabilmekte ve bu veriler kullanılarak sinir ağı modelleri oluşturulabilmektedir. Eğitim, doğrulama ve sorgulama fonksiyonları bulunmaktadır.

Bu programlar haricinde SPSS ve MATLAB uygulamaları içerisinde de yapay sinir ağları ile ilgili özellikler yer almaktadır. MATLAB içerisinde kodlar yardımı ile veya Grafiksel Arayüz kullanımı ile işlem yapılabilir. SPSS için ise Clemetine isimli program yardımı ile işlemler gerçekleştirilebilir.

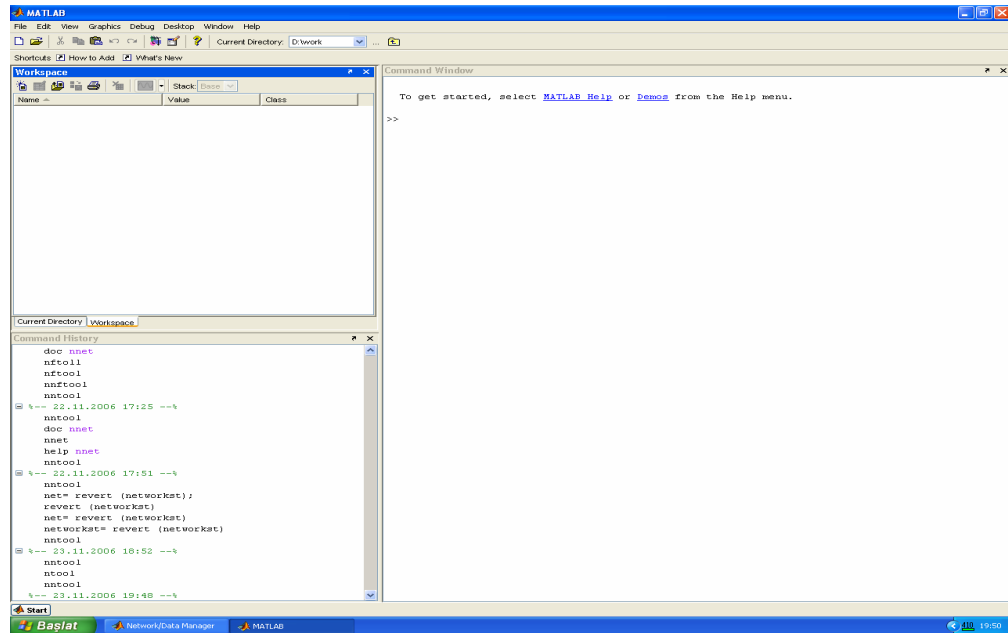
Uygulama için NeuroSolutions kullanılmıştır. Aşağıdaki kısımda NeuroSolutions ve MATLAB Grafiksel Arayüz kullanımı anlatılmaktadır. Ayrıca ekler bölümünde Sınıflandırma Analizi için kullanılabilecek C kodları yer almaktadır. (http://www-ee.uta.edu/eeweb/ip/Software/OldSW/Linear_class.c)

5.1 Yapay Sinir Ağları Uygulamasında Grafiksel Arayüz Kullanımı

Grafiksel Arayüz MATLAB içerisinde kullanıcılar için oluşturulmuş kullanımı basit bir araçtır. Grafiksel Arayüz ile birlikte yapay sinir ağları için model kurulması, eğitilmesi ve test edilmesi mümkündür.

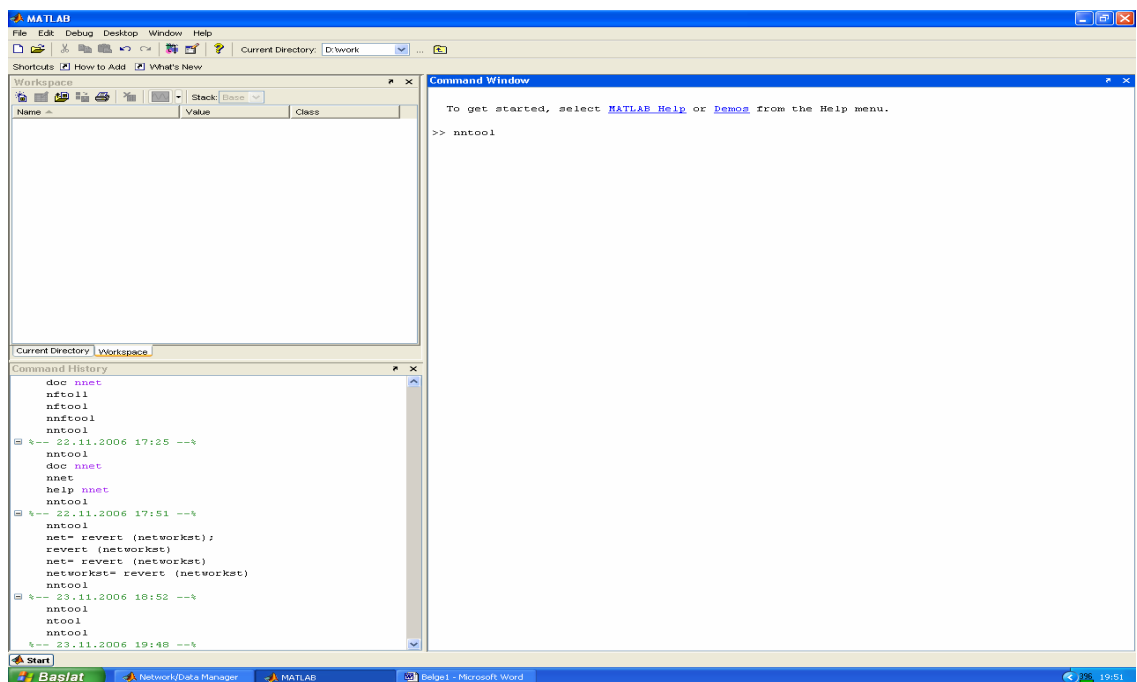
Grafiksel Arayüz kullanımı için yapılması gereken MATLAB komut sayfasında

komutunu girmektedir. Aşağıdaki şekilde MATLAB ana sayfası görülmektedir.



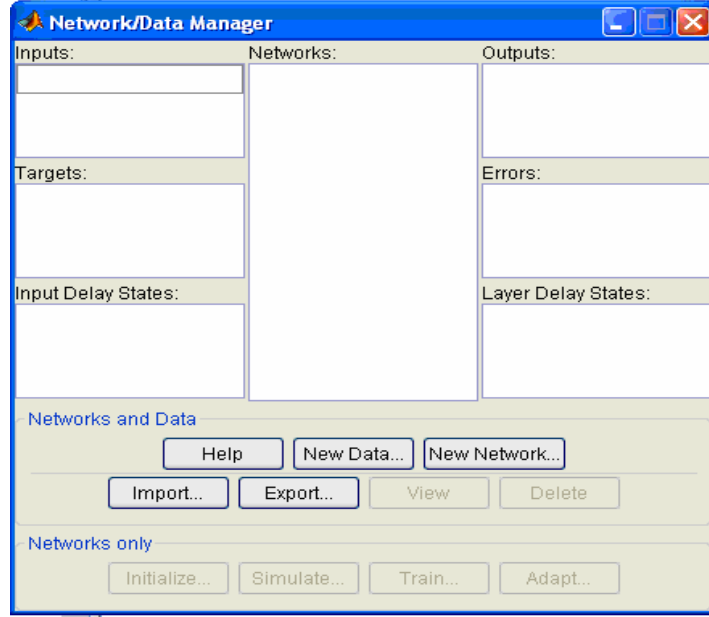
Şekil 5.1 : MATLAB ana sayfası

Sayfa üzerinde nntool komutu alttaki şekilde olduğu gibi girilir.



Şekil 5.2 : MATLAB nntool komutu

Komutun girilmesi ile birlikte Grafiksel Arayüz açılacaktır. Aşağıdaki şekilde Grafiksel Arayüz görülmektedir.



Şekil 5.3 : MATLAB Grafiksel Arayüz

Öncelikle ağın oluşturulması için verilerin girilmesi gerekmektedir. Bu noktada Arayüz üzerindeki New Data seçeneği kullanılır.

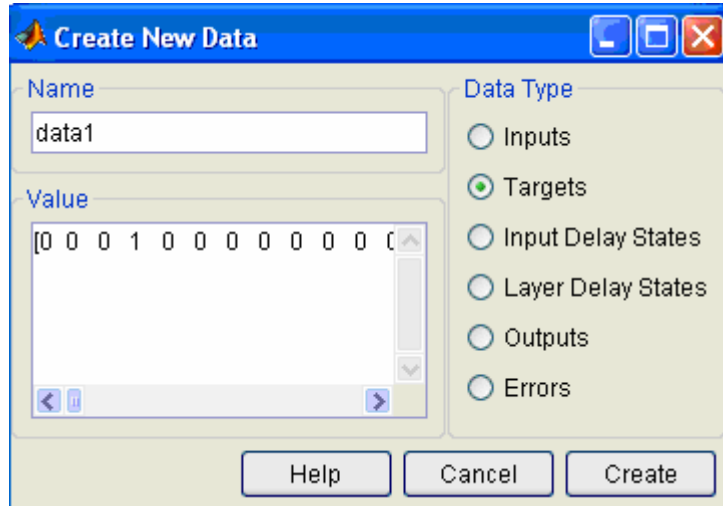
Girilecek olan veri formatı aşağıdaki gibi olmalıdır.

[0 0 1 10 1 0 1]

Örnek olarak verilen veriler iki değişken için alınmış ve dört kişiye uygulanmıştır. Bu veriler için ağa tanıtılacak olan sonuçlar ise aşağıdaki şekildedir.

[0 0 0 1]

New Data seçeneği seçildiğinde aşağıdaki pencere açılacaktır.



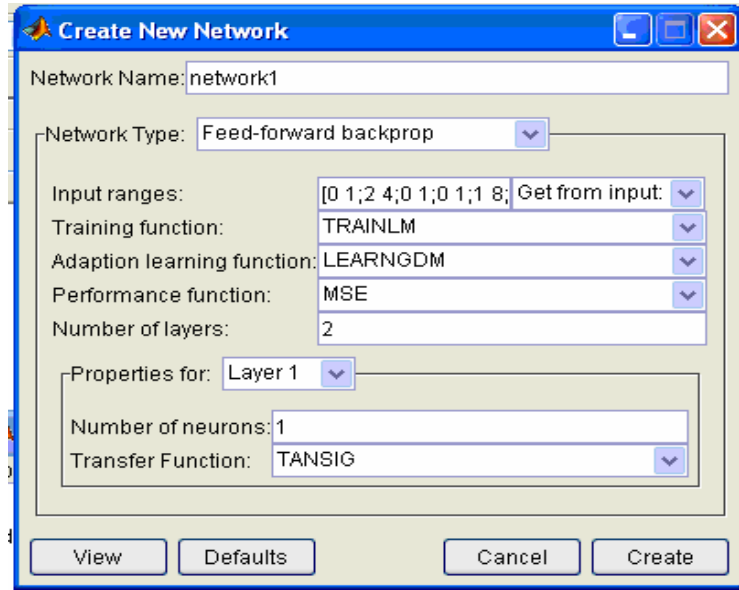
Şekil 5.4 : MATLAB Veri Giriş Penceresi

Pencere üzerinde name yazan bölüme verilerin ismini, bu isim veriler için genellikle p ve hedefler için genellikle t olarak alınır, value yazan bölüme daha önce belirtmiş olduğumuz değerleri girerek veri çeşidini sağ taraftan seçmeniz gerekmektedir. Veriler için Inputs, sonuçlar içinse Targets seçeneği seçilerek Create tuşuna basılır. Bu durumda ağ için kullanılacak veriler ve hedefler programa tanıtılmış olunur.

Ağın Oluşturulması

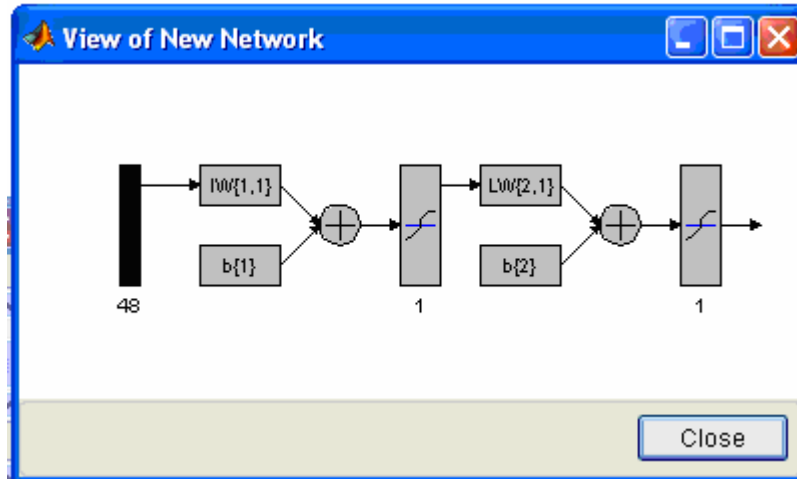
Ağın oluşturulması için arayüz üzerindeki new network seçeneği seçilir. Bu seçenek ile birlikte aşağıdaki pencere açılacaktır. Burada oluşturulacak ağ için gerekli seçimler yapılır. Input ranges kısmında yer alan veriler aşağıdaki formatta olup, veriler içerisindeki her bir değişken için minimum ve maksimum değerleridir. Dilenirse bu değerler Get From Input seçeneği ile birlikte otomatik olarak belirlenebilir. En son olarak Create tuşuna basılır ve ağ yaratılmış olunur.

[0 10 1].



Şekil 5.5 : MATLAB ağ oluşturma

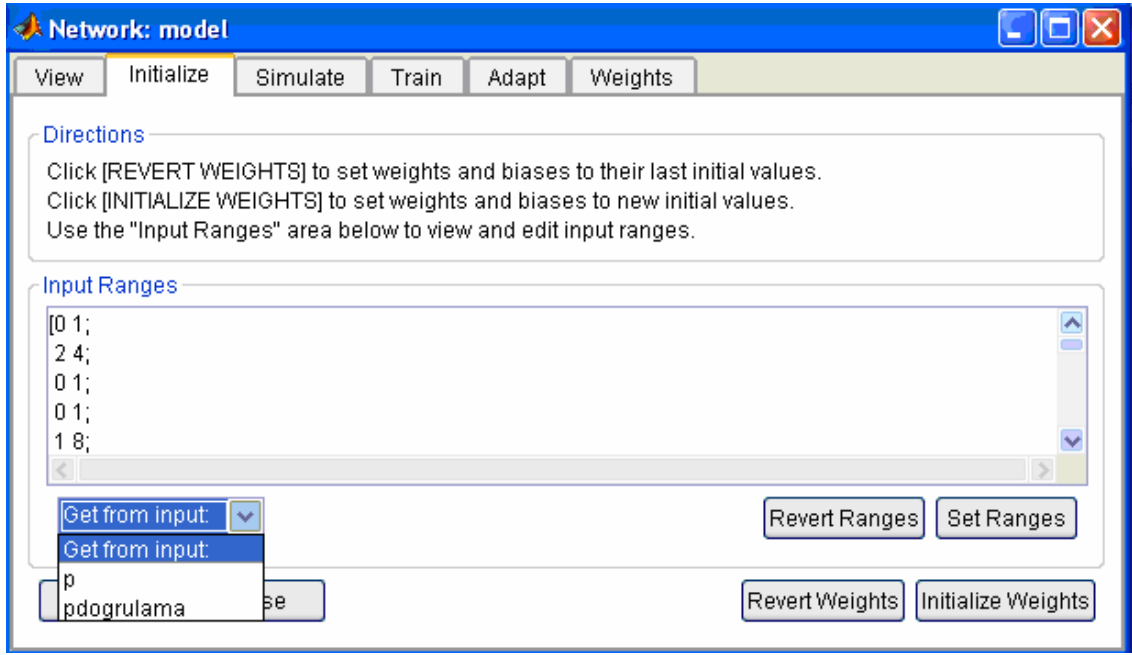
Arayüz üzerinde ağ ismi ile birlikte Network sekmesinin altında oluşturulan ağ görülmektedir. Bu işlem sonrasında ağın ismi seçilirse ve arayüzdeki view seçeneği seçilirse ağ görüntülenecektir. Aşağıdaki örnekte 48 adet için değişken hazırlanmış tek gizli katmanlı bir ağ yapısı görülmektedir.



Şekil 5.6 : Oluşturulan ağın yapısı

Ağ Üzerinde Çalışma

Ağ üzerinde çalışma için ağın ismi seçildikten sonra arayüz üzerindeki Initialize seneçeği seçilirse aşağıdaki ekran görülecektir. Bu ekran üzerinde ağ için aralıkların belirlenmesi, ağın eğitilmesi, adapte edilmesi, ağırlıkların ayarlanması ve ağın simule edilmesi seçenekleri mevcuttur.



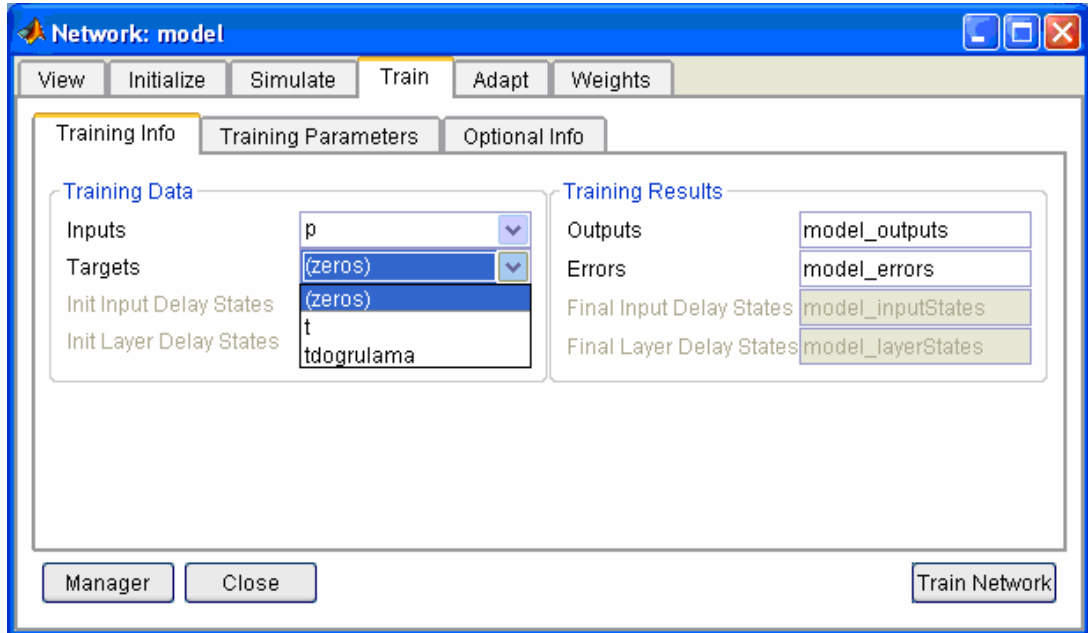
Şekil 5.7 : Model üzerinde çalışma penceresi

Ağ İçin Aralıkların Belirlenmesi

Ağ için kullanılacak olan aralıklar get from input seçeği ile birlikte elde edilebilmektedir. Sonrasında Set Ranges ve Initialize Weights ile birlikte işlemler tamamlanmış olacaktır.

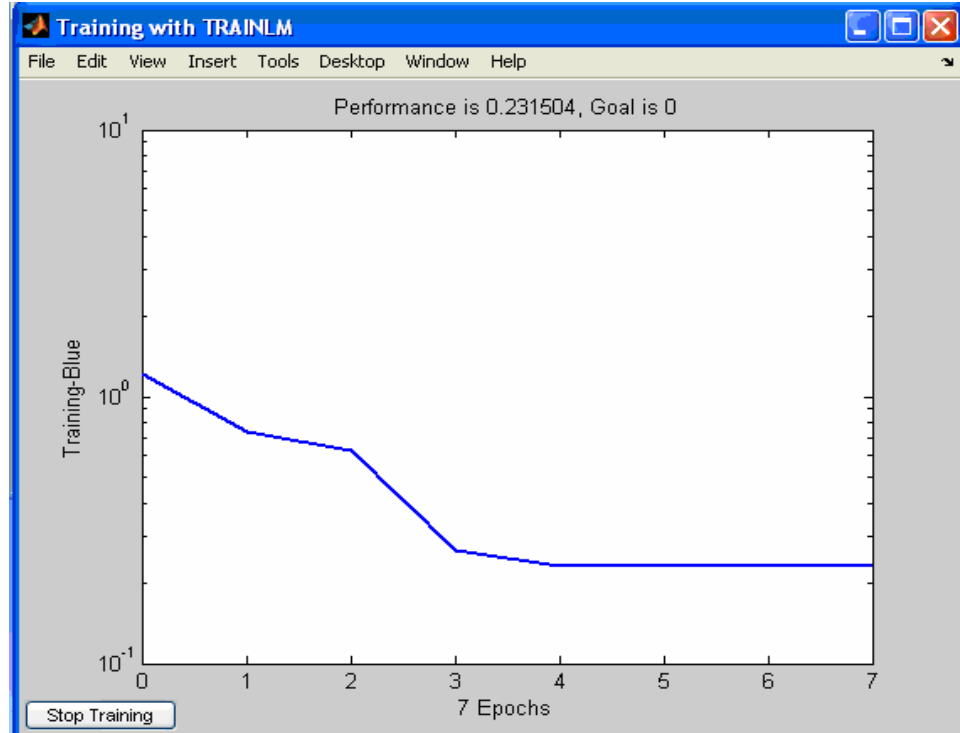
Ağın Eğitilmesi

Ağın eğitilmesi için Train seçeneği seçilmelidir. Bu durumda aşağıdaki pencere açılacaktır. Pencere üzerinde verileri ve hedef değerler seçilir. Çıktı ve hata dosyaları içim isimler ise sağ tarafta belirtilir. Dosyaların isimlerini belirlerken dikkat edilmesi gereken nokta daha önce aynı isimle bir dosya seçilmemiş olmasıdır. Bu durumda oluşturulacak yeni dosya aynı isimdeki eski dosyanın üzerine yazılabilir ve eski veriler kaybedilebilir. Ağ ile ilgili daha fazla eğitim bilgisi girilmek istenirse Parametre veya Seçimli Bilgi kısımları kullanılır. Değerlerin girilmesi sonrasında Train Network seçeneği ile birlikte ağ eğitimi başlatılır.



Şekil 5.8 : Modelin Eğitilmesi

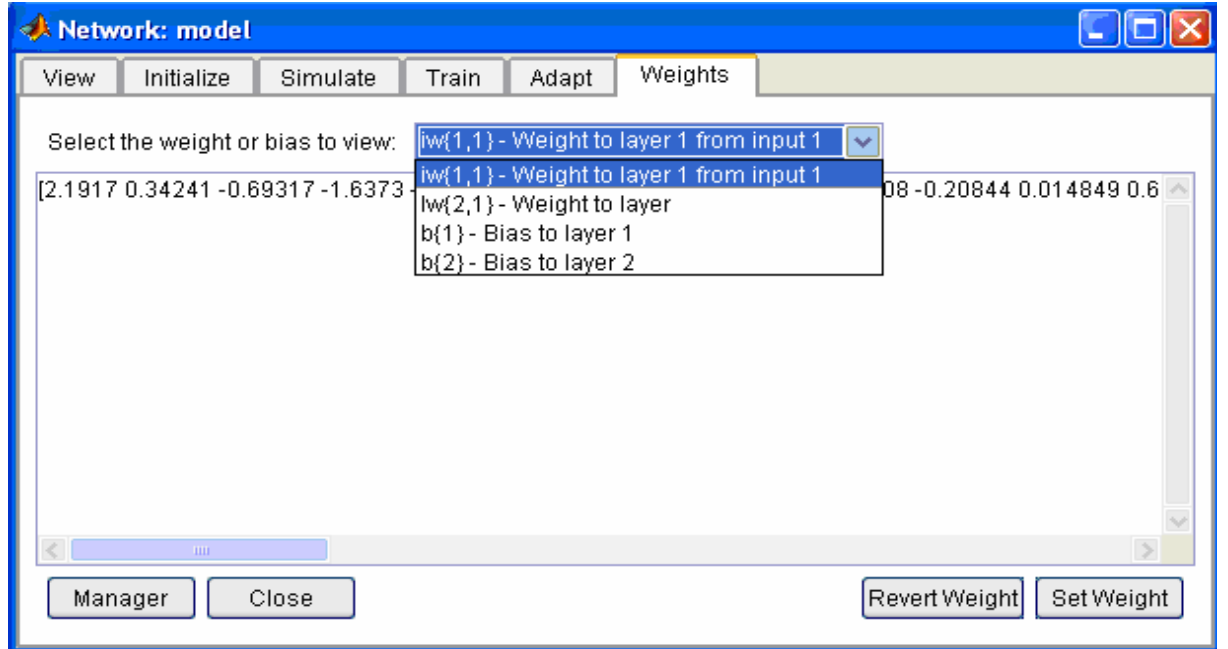
Eğitim sonrasında aşağıdaki gibi eğitim esnasında ağ üzerindeki değişiklikler belirtilir.



Şekil 5.9 : Eğitim Performansı

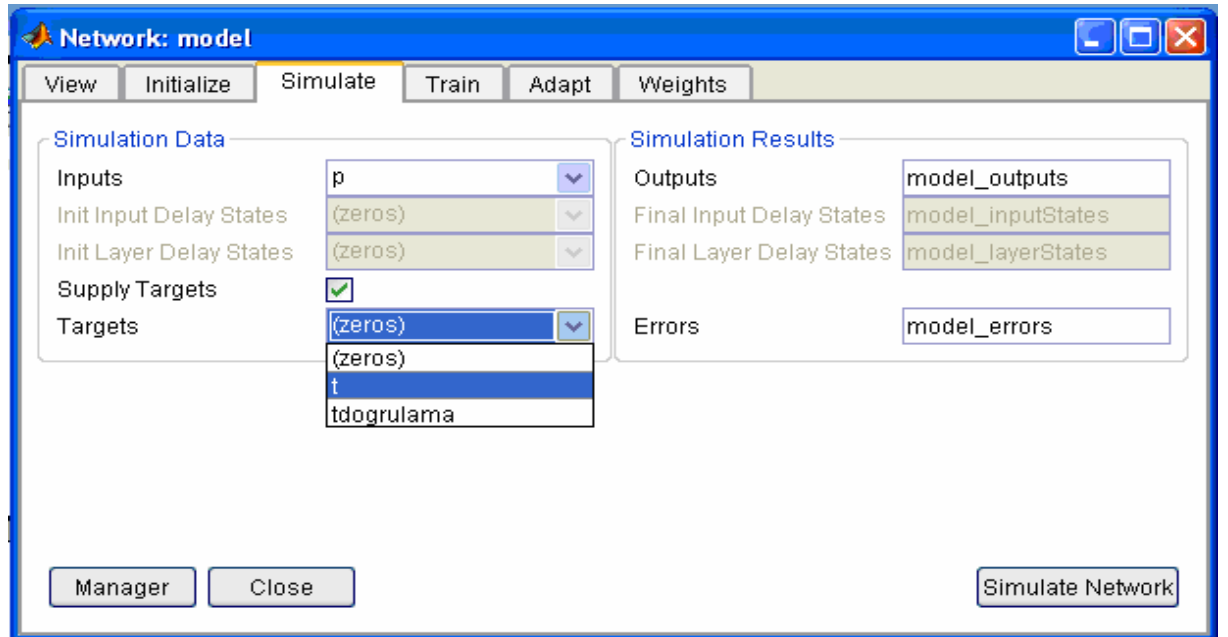
Ağ Ağırlıkları

Ağ üzerinde eğitim sonrasında ağın bağlantılarındaki ağırlık değerleri Weights bölümünde görülebilir ve bu bölümden değişiklikler gerçekleştirilebilir. Değişiklik yapılması durumunda Adapt ile birlikte ağ tekrar modifiye edilebilir.



Şekil 5.10 : Ağırlıkların Belirlenmesi

Ağın Simülasyonu

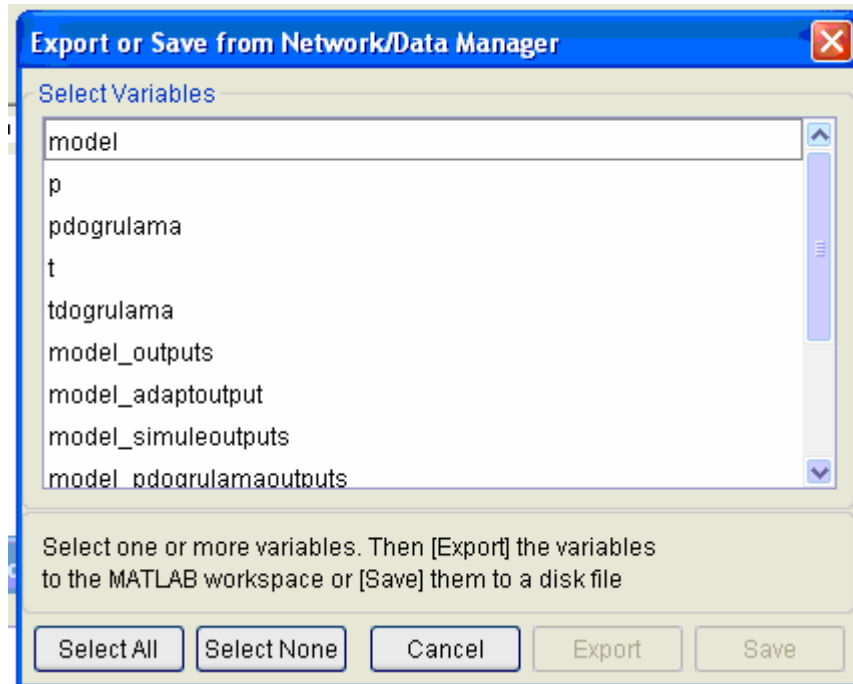


Şekil 5.11 : Simülasyon Verileri

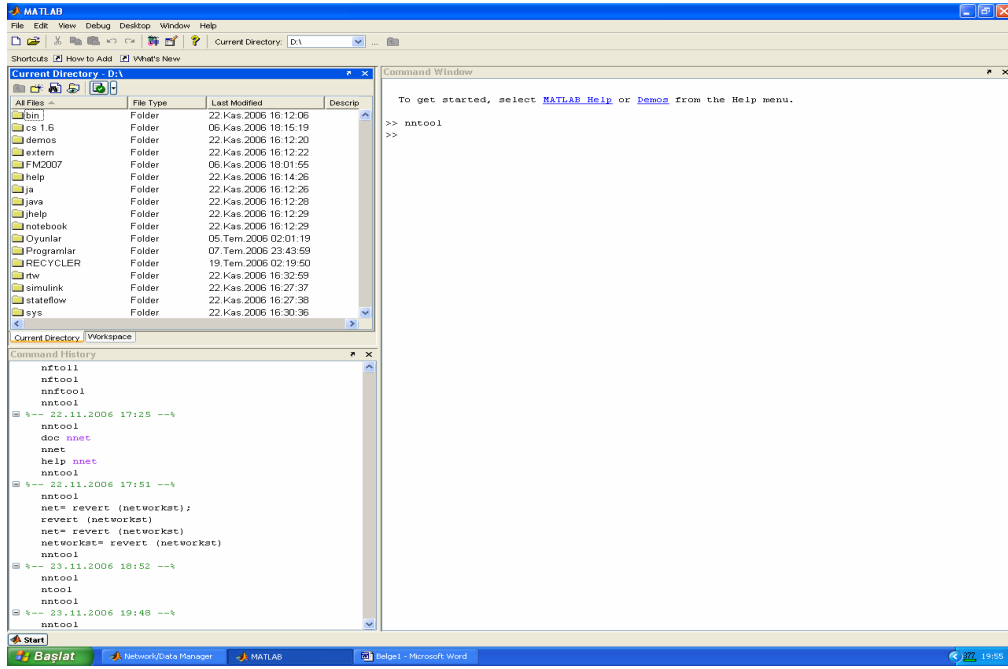
Ağa girilen değerlerin eğitim sonrasında simülasyonu için Simulate seçeneği kullanılmaktadır. Bu seçenek ile birlikte veriler sonrasında sonuçların elde edilmesi ve başarı oranı bulunması mümkündür.

Verilerin Aktarılması

Tüm sonuçların programdan alınabilmesi için arayüzdeki export seçeneği kullanılabilir. Bu bölümde aktarılmak istenen veriler seçilir ve export tuşuna basılırsa veriler dosyalar halinde MATLAB ana sayfasına aktarılır. Aşağıdaki şekillerde bu işlemler gösterilmiştir.



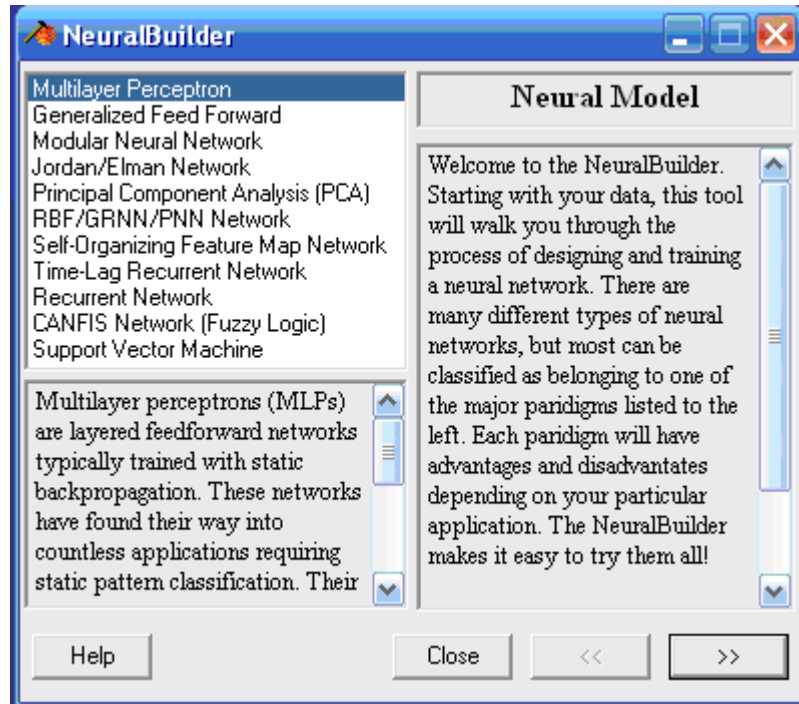
Şekil 5.12 : Verilerin MATLAB ana sayfasına aktarılması



Şekil 5.13 : MATLAB ana sayfası – veri aktarımı

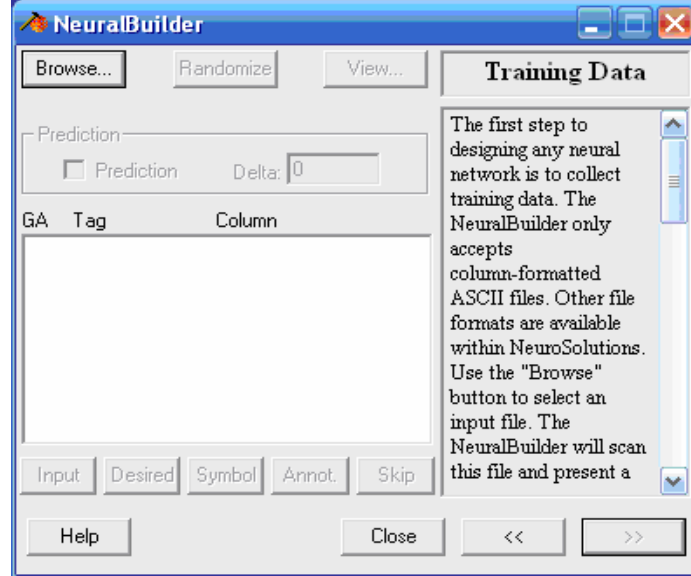
5.2 NeuroSolutions

NeuroSolutions, Neuro Builder, Neuro Expert ve NeuroSolutions for Excel menülerinden oluşan bir programdır. Bu bölümde Neuro Builder ile ilgili detaylar aktarılacaktır. Neuro Builder başlangıç menüsü Şekil 5.14’te görülmektedir.



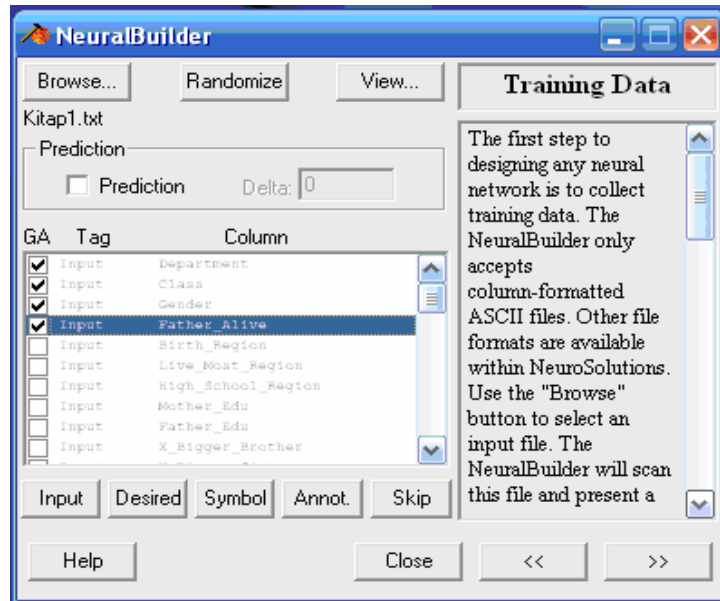
Şekil 5.14 : NeuroBuilder

Başlangıç menüsü içerisinde oluşturulacak ağ tipi seçilmektedir. Ağ tipi seçimi sonrasında bu ağın oluşumu için kullanılacak veriler sisteme tanıtılmaktadır. Bunun için kullanılan ekran görüntüsü Şekil 5.15'te görülmektedir. NeuroBuilder ile txt, col, asc veya csv uzantılı dosyalar kullanılabilir. Bu ekran üzerinden browse tuşu kullanılarak ilgili dosyalar programa aktarılır.



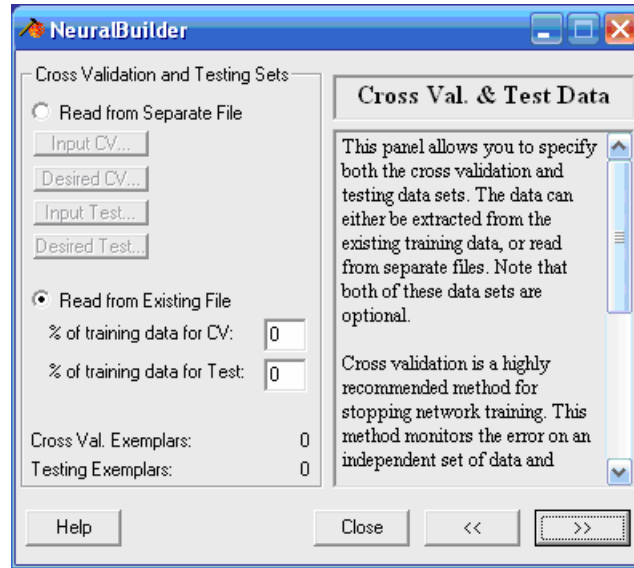
Şekil 5.15 : NeuroSolutions Veri Aktarımı

Verilerin programa aktarılması sonrasında yapılacak işlem her bir değişken tipini belirlemektir. Değişkenler input, desired, symbol ve annotate olarak etiketlenebileceği gibi değişkenlerden birinin veya bir kaçının ağın oluşumu dışında tutulması da mümkündür.



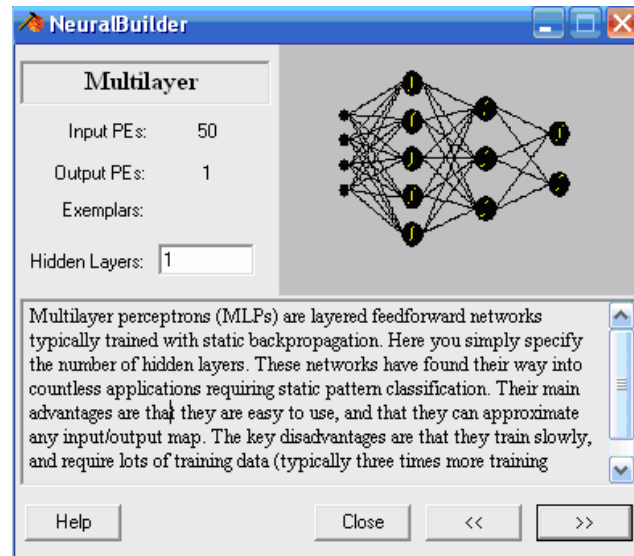
Şekil 5.16 : NeuroSolutions Değişken Tipleri

Veri tiplerinin belirlenmesi sonrasında eğitim esnasında kullanılacak doğrulama ve eğitim sonrası test verisi için veriler belirlenir. Bu aşamada eğitim setinin içerisinden seçilerek eğitim işlemi haricinde tutulmak üzere belirli yüzdelerde veri seçilebileceği gibi farklı bir dosya ile veri aktarmak ta mümkündür.



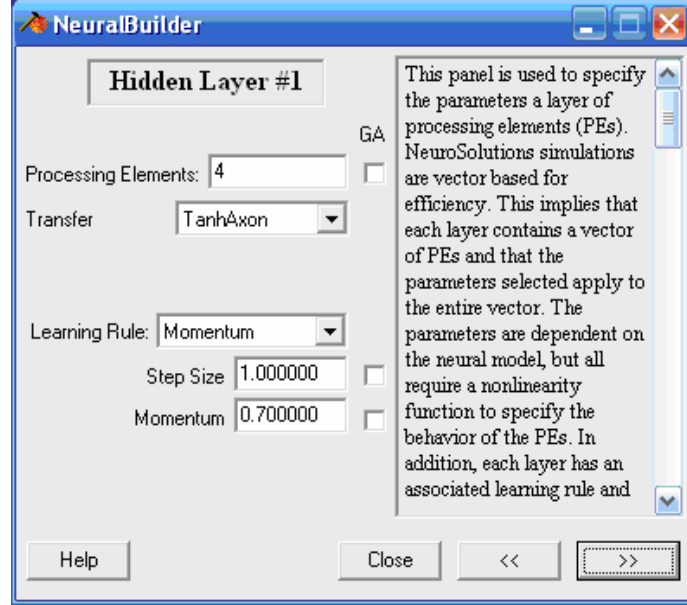
Şekil 5.17 : NeuroSolutions Verilerin Ayrıştırılması

Verilerin hangi işlemlerde veya hangi yüzdelerde kullanılacağını belirlenmesi sonrasında ağ yapısı oluşturulur. Bu aşamada ilk olarak gizli hücre sayısı belirtilir.



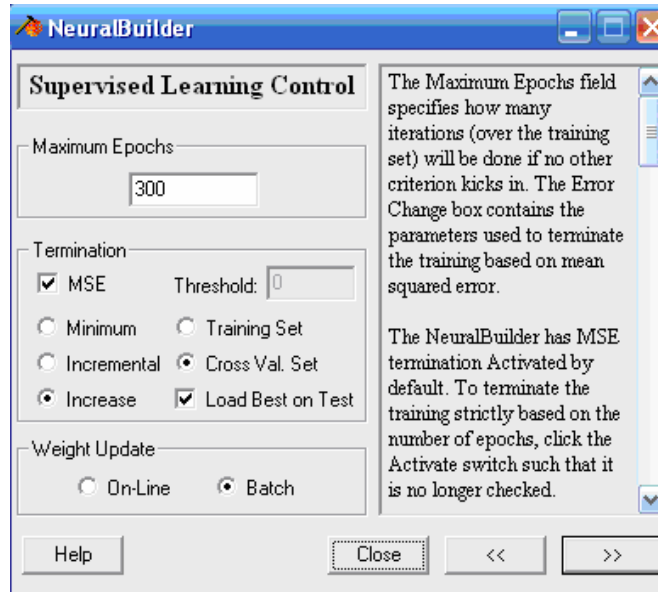
Şekil 5.18 : Gizli Hücre Sayısının Belirlenmesi

Gizli hücre sayısının belirlenmesi sonrasında sırası ile her bir gizli hücre için ve çıkış hücresi için nöron sayısı ve momentum değeri belirtilir. Ayrıca transfer fonksiyonu ve eğitim metodu da burada seçilmektedir.



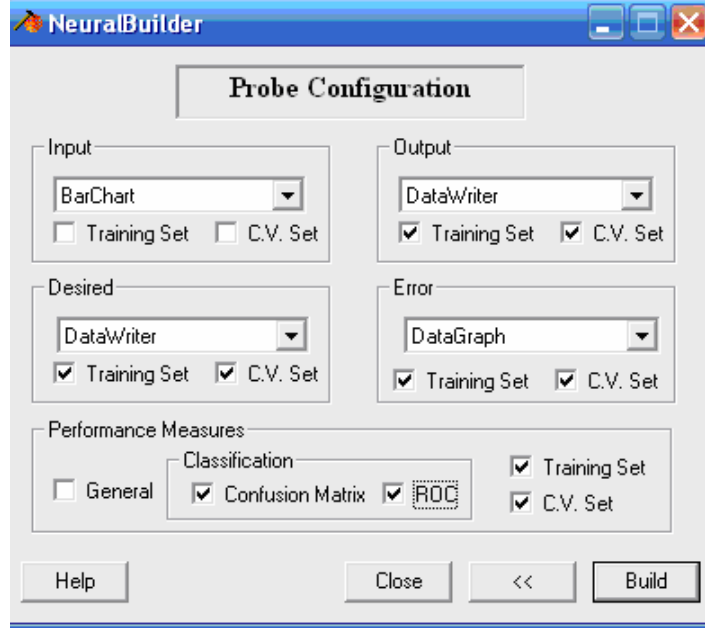
Şekil 5.19 : NeuroSolutions Eğitim Parametrelerinin Seçilmesi

Ağın oluşturulmasında bir sonraki aşamada eğitimdeki çevrim sayısı, hata analizinde kullanılacak metot ve ağırlıkların güncellenme metodu belirtilecektir.



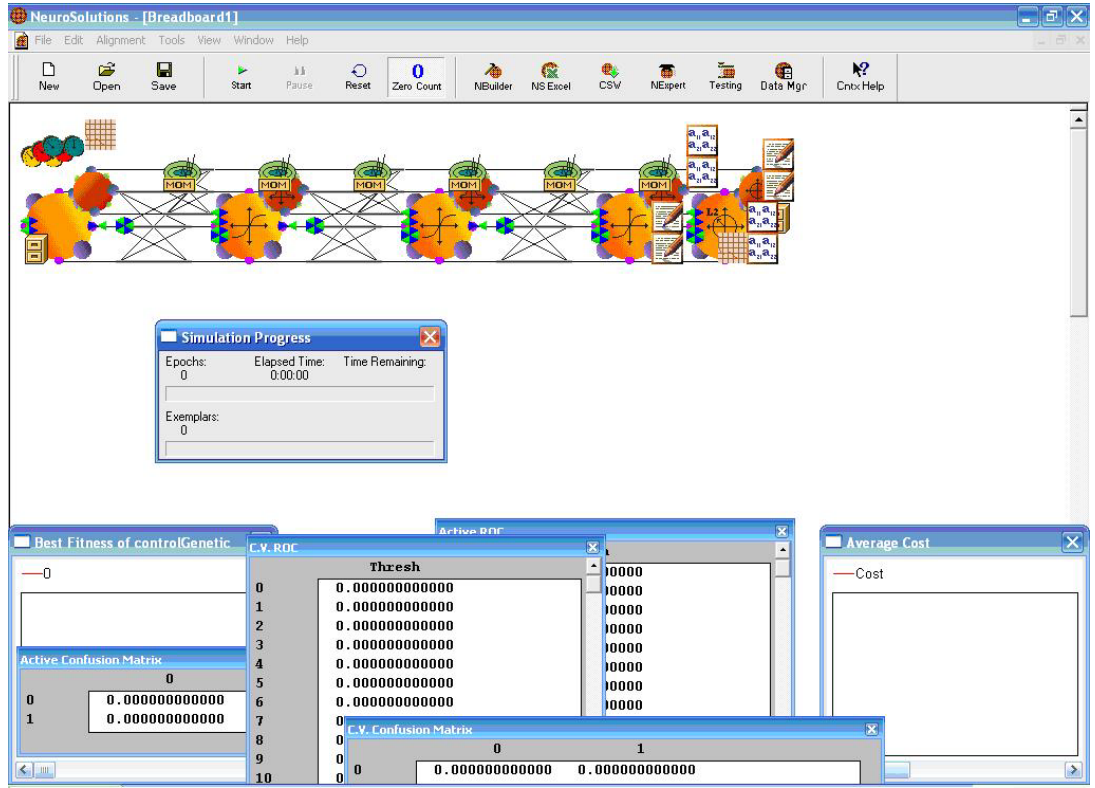
Şekil 5.20 : NeuroSolutions Parametrelerinin Seçilmesi -2

Ağ ile ilgili verilerin ve değişkenlerin seçilmesi sonrasında sonuçların gösterilmesindeki seçimler yapılır.



Şekil 5.21 : NeuroSolutions Sonuçların Gösterilmesi için Düzenleme

Tüm bu işlemlerin tamamlanması sonrasında “Build” tuşu kullanılarak ağ oluşturulur. Ağın oluşturulması sonrasında aşağıdaki şekildeki ekran görülecektir.



Şekil 5.22 : NeuroSolutions Ağ Yapısı

Menü üzerinde yer alan “Start” tuşu kullanılarak ağın eğitimi başlatılır. Tüm sonuçlar diğer ekranlarda ağ eğitimindeki sonuçlara göre güncellenmektedir.

6. YAPAY SİNİR AĞLARI UYGULAMASI

Yapay sinir ağı uygulamasında toplam 424 adet veri bulunmaktadır. Bu veriler hem modelin eğitilmesinde hem de uygulanmasında kullanılmıştır. Toplam 50 adet değişken için bir adet hedef değeri yer almaktadır.

Ağın eğitilmesi için seçilmiş olan parametrelerin listesi aşağıda yer almaktadır.

Tablo 6.1: Uygulama Değişkenleri

Eğitim Tipi	Gözetimli
Fonksiyon Tipi	Çok katmanlı algılayıcı tipli geri yayılım algoritması
Performans Fonksiyonu	MSE
Gizli Katman Sayısı	2
Nöron Sayısı	20
Tranfer Fonksiyonu	Tansig
Girdi Sayısı	50
Momentum	0,1
Çevrim Sayısı	300
Öğrenme Modu	Online

Ağın oluşturulmasında ve eğitim modelinin seçiminde hedef değerlerinin bilinmesi rol oynamıştır. Hedef değerlerinin bilinerek ağın eğitilmesinde gözetimli öğrenme metotları kullanılmaktadır.

Fonksiyon tipi seçiminde sınıflandırma analizi için kullanılan fonksiyonlar belirlenmiş ve bunlardan en yaygın olarak kullanılan tip seçilmiştir. Ayrıca bu fonksiyon tipi, diğer fonksiyon tiplerinden daha başarılı sonuçlar vermektedir.

Seçilen çok katmanlı algılayıcılar metodu için yapılan işlemin sınıflandırma analizi olması sebebi ile geri yayımlı algoritma ile çalışmak uygun görülmüştür.

Gizli hücre sayısı artırılarak performans test edilmiştir. İdeal gizli hücre sayısı mevcut veriler için 2 olarak gözlenmiştir.

Her bir gizli hücredeki nöron sayısı da artırılarak işleme sokulmuş ve test performansı göz önünde bulundurulduğunda 20 nöronun uygun olduğu gözlenmiştir.

Çevrim sayısı da artırılarak işlem yapılmıştır. Az sayıda olması eğitimin gerçekleşmemesine ve çok sayıda olması da ezber yapan bir ağa sebep olduğu için bu işlem gerçekleştirilmiş olup, 300 çevrim ile ideal başarıya ulaşılmıştır.

Uygulama sonucunda başarısız öğrencilerin %99,78'i ve başarılı öğrencilerin %78,57 si doğru olarak sınıflandırılmıştır. Ağın kendi başarısı daha fazla olmasına rağmen uygulama sonuçları olarak test grubunun sonuçları kabul edilmiştir. Bu şekilde eğitim esnasında cevapları öğrenilen veri setlerinin başarıyı yanlış bir şekilde yükseltmesi de engellenmiştir.

Toplam başarı oranı %91,77'dir.

Değişkenlere ait katsayılar karşılaştırılarak bu değişkenlerin sonuçlar üzerindeki etkileri gözlenmiştir. Tablo 6.2'de bu değerler yer almaktadır.

Tablo 6.2: Başarıyı Etkileyen Değişkenler ve Ağırlıkları

Değişken İsmi	Önem
Devamlılık Yüzdesi	1,00
Sınıf	0,97
Ortalama Çalışma Süresi	0,64
Baba Hayatta	0,63
Bursluluk	0,59
Ortalama İnternet Kullanımı	0,59
Ortalama Televizyon İzleme Süresi	0,58
İstanbul'da İkametgah Yeri	0,54
Lise Öğrenimi Bölgesi	0,52
Lise Tipi	0,52
Baba Eğitim Düzeyi	0,51
Yaşça Büyük Kız Kardeş Sayısı	0,51
Anne Çalışma Durumu	0,51
Öğrenim Kredisi	0,51
Hobi/Kişisel Gelişim Kursu	0,50

7. HER İKİ METODUN KARŞILAŞTIRILMASI

Üniversite öğrencilerinin başarılarını etkileyen değişkenlerin sınıflandırılması için karar ağaçları ve yapay sinir ağları uygulamaları gerçekleştirilmiştir. Her iki uygulamada da aynı veriler kullanılarak alınacak olan sonuçların veri farklılığından etkilenmesi engellenmiştir. Uygulama sonunda her bir metod için toplam öğrencilerin yüzdesel olarak kaçının doğru olarak sınıflandırıldığı ve bu sınıflandırmada hangi etkenlerin daha önemli olduğu sonuçları bulunmuştur.

Karar Ağaçları uygulaması sonucunda verilerin başarılı veya başarısız olarak sınıflandırılmasındaki başarı oranı %86'dır.

Karar ağaçları uygulamasında, 208 öğrenci başarılı olarak belirlenmiştir. Bunların 173 tanesi ilk baştaki grup içerisindeki başarılı öğrencilerdir. 27 adet başarılı öğrenci hatalı olarak başarısız olarak sınıflandırılmıştır.

Karar ağaçları uygulamasında, 216 öğrenci başarısız olarak belirlenmiştir. Bu öğrencilerden 189'u başlangıç verilerinde başarısız olan öğrencilerdir. Toplam 27 adet başarısız öğrenci sınıflandırması hatası görülmektedir.

Karar ağaçları sonucunda elde edilen toplam hata 62'dir.

Yapay sinir ağları uygulaması sonucunda verilerin başarılı veya başarısız olarak sınıflandırılmasında elde edilen başarı oranı %91,77 olarak bulunmuştur.

Uygulama sonucunda başarısız öğrencilerin %99,78'i, başarılı öğrencilerin %78,57 si doğru olarak sınıflandırılmıştır.

Sınıflandırma işlemleri karşılaştırılmak istenilirse yapay sinir ağları uygulamasının karar ağaçları uygulaması karşısındaki başarısı açıkça görülmektedir. Klasik doğrusal modelleme anlayışından farklı olarak tüm etkenleri ele alan ve veri eğitimleri sonrasında tahmin işleminde analiz ve sınıflandırmayı etkin bir şekilde yapan yapay sinir ağları ideal model olarak görülmektedir.

Yapay sinir ağlarının başarısındaki etkenlerin başında verileri kullanarak eğitim süreci gerçekleştirilmesi ve bu süreç sonrasında kendini daima yenileyerek hata oranının azaltmaya çalışması gelebilir.

Yapay sinir ağlarında verilerin işlenmesi için seçilen fonksiyonlar ve bu farklı fonksiyonlar kullanabilme imkanı başarı üzerinde etkili olmaktadır.

Karar ağaçları ve yapay sinir ağları sınıflandırma uygulamaları sonucunda aşağıdaki değişkenler başarı üzerinde etkili olarak bulunmuştur.

Tablo 7.1: Başarı Etkileyen Değişkenler ve Önem Dereceleri (Karar Ağaçları)

Değişken İsmi	Önem
Devamlılık Yüzdesi	1,0000
Cinsiyet	0,6192
İstanbul’da İkametgah Yeri	0,5022
En Çok Sevilen Ders Grubu	0,4518
İş Bulma	0,3755
Doğum Yeri/Bölge	0,3745
Sınıf	0,3264
Bursluluk	0,3056
Baba Eğitim Düzeyi	0,2792
Ortalama Çalışma Süresi	0,2687
İ.T.Ü.’de Olmaktan Duyulan Memnuniyet	0,2546
Yüksek Lisans Kararı	0,2403
Baba Hayatta	0,2362
Sinemaya Gitme Sıklığı	0,2204
Aylık Ortalama Harcama	0,1976

Tablo 7.2: Başarı Etkileyen Değişkenler ve Önem Dereceleri (Yapay Sinir Ağları)

Değişken İsmi	Önem
Devamlılık Yüzdesi	1,00
Sınıf	0,97
Ortalama Çalışma Süresi	0,64
Baba Hayatta	0,63
Bursluluk	0,59
Ortalama İnternet Kullanımı	0,59
Ortalama Televizyon İzleme Süresi	0,58
İstanbul’da İkametgah Yeri	0,54
Lise Öğrenimi Bölgesi	0,52
Lise Tipi	0,52
Baba Eğitim Düzeyi	0,51
Yaşça Büyük Kız Kardeş Sayısı	0,51
Anne Çalışma Durumu	0,51
Öğrenim Kredisi	0,51
Hobi/Kişisel Gelişim Kursu	0,50

Değişkenler incelendiğinde etkin olan ilk 15 tanesi ele alınmıştır. Her iki uygulamada da etkili olan değişkenler ise aşağıda toplu olarak belirtilmiştir.

Tablo 7.3: Ortak Başarı Değişkenleri

Devamlılık Yüzdesi
Sınıf
Ortalama Çalışma Süresi
Baba Hayatta
Bursluluk
İstanbul'da İkametgah Yeri
Baba Eğitim Düzeyi

Veriler incelendiğinde öğrencilerin derslere devamlılığının başarı üzerinde en etkili yöntem olduğu görülmektedir. İ.T.Ü'nde devamsızlık için minimum oran %70'tir. Bu ise öğrencilerin derse katılımında etkili olmakta ve derse katılım ile birlikte başarı oranı da artmaktadır. Eğitim ve öğrenim sürecinde derse katılım bireysel öğrenmeden daha etkilidir. Bu ise öğretmenlerin öğrenciler için bireysel çalışmalardan daha etkili şekilde öğrenmede etkili olduğunu göstermektedir.

Üniversite eğitimi ile birlikte ilerleyen sınıflarda öğrenciler daha başarılı olabilmektedirler. Bu değişken üzerinde başarının sosyal faktörü ve kişinin psikolojisi ve alışma süreci gibi etkenler ortaya çıkmaktadır. Öğrencilerin çoğunluğu farklı şehirlerden gelmektedirler. Bu ise öğrencilerin ilk yıllarda alışma süreci içerisine girmesine sebep olmaktadır. Ayrıca ailelerinden ilk defa ayrı kalma, farklı bir ortamda olma gibi faktörler de bu alışma sürecinde etkilidir. Öğrencilerin yeni şehre ve bulundukları ortama alışmaları sonrasında başarıları artmaktadır. Ek olarak öğrencilerin alışma sürecinde bir diğer odaklandıkları nokta ise okulları ve derslerini aldıkları öğretmenleridir. Bu süre içerisinde yerleşim yeri ve koşullarına ayak uyduran ve sosyal olarak yeniliklere alışan öğrenci aynı zamanda yeni öğretim yerine de alışmakta ve farklılıkların başarı üzerinde yapmış olduğu etki zamanla azalmaktadır.

Öğrencilerin başarıları üzerinde yerleşim yerinin etkili olması aslında alışma sürecinin etkisini tekrar göstermektedir. Öğrencilerin yaşadıkları yerler üniversite öncesi yaşam ortamlarına ne kadar yakınsa öğrencilerin alışma süreci azalacak ve başarı erken ortaya çıkacaktır. Yerleşim yerindeki düzensizlikler ve farklılıklar ise bu

süreci artıracak ve alışma süreci sonucunda öğrencinin başarılı olma zamanı gecikebilecek veya başarılı olması mümkün olmayacaktır.

Başarı üzerinde bir diğer etken çalışma saatleridir. Öğrencilerin üniversite öğrenimleri boyunca ders çalışma haricinde proje, uygulama vb. çalışmaları da yer almaktadır. Bu sebeple öğrenciler daha önceki yıllarda, örneğin lise eğitiminde, almış oldukları eğitim-öğrenim yapısından biraz farklı olarak araştırmaya yönelmektedir. Bilgiye erişmek ve öğrenilenleri uygulayarak görmek ve bunlarla birlikte öğrenimi pekiştirmek üniversitede temel etkenlerdir. Bu ise öğrencilerin çalışma sürelerinin fazla olmasını gerektirmektedir. Her ne kadar öğrenme kişisel zeka ile de etkili olsa da artan zaman ile birlikte başarı da artmaktadır.

Başarının getirdiği bir sonuç olarak öğrenciler burs kazanabilmektedirler. Aslında bursluluğun başarı üzerindeki etkisinden önce başarı bursluluk üzerinde etkilidir. Bu ilk etkinin oluşması için öncelikle başarı gerekmektedir. Başarı ile kazanılan burs öğrencinin motivasyonunu artırmaktadır. Ayrıca öğrencinin maddi anlamda sorunlarını azaltmakta ve daha rahat bir zihin ile çalışma imkanı oluşmaktadır. Bursluluk ile birlikte öğrencilerin para kazanmak için çalışma gereksinimleri ortadan kalmakta ve bu sayede öğrenciler dersleri için daha fazla zaman ayırabilmektedirler. Bu ise ikinci etkiyi yani bursluluğun başarıyı artırmasını oluşturmaktadır.

Son olarak babanın hayatta olması ve babanın eğitimi beraber ele alınacaktır. Burada ilk başta dikkat çeken öğrencilerin yaklaşık %68'inin erkek olmasıdır. Erkek öğrencilerin çoğunluğu için baba bir örnek model oluşturmaktadır. Örnek modelin hayatta olması onların daha fazla motive olmasını sağlamaktadır. Motivasyonda bir diğer etken olan takdir edilmenin önemi burada anlaşılabilir. Öğrencinin başarısının takdir edilmesi veya ona yol gösterilmesi öğrenciler tarafından örnek model olarak görülen baba tarafından gerçekleştirildiğinde başarı artmaktadır. Bu sebeple babanın hayatta olması başarı üzerinde etkili olabilmektedir. Ayrıca öğrencilerin %52'sinin annesinin çalışmadığı görülmektedir. Bu ise öğrenciler için maddi anlamda etkinin çoğunlukla baba olduğunu göstermektedir. Babanın hayatta olması da öğrencinin finansal anlamda daha rahat bir öğrenim süreci yaşaması imkanını oluşturmaktadır.

Rol model olarak eğitim incelendiğinde veri grubunda gözlemlenen annelerin 118 tanesinin babaların ise 246 tanesinin üniversitede eğitim gördükleri anlaşılmaktadır. Bu ise öğrencilerin babalarını eğitim-öğrenim süreçlerinde rol model

olarak aldıklarını gösteren bir diğ er sonu tur.  ğrencilerin babalarının eğitimlerinin başarılarında etki ise burada açık a g r lmektedir.

Sonuç olarak her iki model birbiri ile karşılaştırıldığında yapay sinir ağılarının esnek yapısı ile sınıflandırmada daha başarılı olduğı ve başarı üzerinde etkin olan fakt rlerde ortak olanların derse katılım, sınıf,  alıřma saatleri babanın hayatta olması, bursluluk, yařanılan yer ve babanın eğitimi olduğı g r lmektedir.

8. SONUÇ ve ÖNERİLER

Bu çalışma içerisinde yapay sinir ağları ve karar ağaçları metotları sınıflandırma analizi yapılarak karşılaştırılmıştır. Karşılaştırma için üniversite öğrencilerinin başarılarını etkileyen faktörlerin belirlenmesi amacıyla yapılmış olan anket kullanılmıştır. Çalışmaya 438 öğrenci yanıt vermiş, uygulamalara anketlerde not ortalamasını belirtmiş olan 424 öğrenci dahil edilmiştir.

Karar ağaçları ile öğrencileri başarılarına göre sınıflandırma işlemi sonrasında %86 oranında bir başarı elde edilmiştir. Aynı verilerle yapılan yapay sinir ağları uygulaması sonrasında başarı oranı %91,77 olarak bulunmuştur. Bu sonuçlarla birlikte yapay sinir ağlarının sınıflandırma analizinde karar ağaçlarından daha başarılı sonuçlar ortaya koyduğu görülmektedir.

Başarı olarak yapay sinir ağlarının daha iyi olması yapılacak akademik çalışmalarda veya iş ortamında yapay sinir ağlarının etkin olarak kullanılabileceğini göstermektedir. Bu sayede hem daha doğru sonuçlara ulaşma imkanı elde edilecek hem de bu alanın gelişmesine katkıda bulunulmuş olunacaktır.

Öğrencilerin başarılarını etkileyen değişkenlerin belirlenmiş olması da bu çalışma sonucunda ortaya çıkan önemli verilerdendir. Her iki uygulamanın da sonuçları baz alınarak yapılan inceleme sonucunda, üniversite öğrencilerinin başarılarını etkileyen değişkenler,

- Devamlılık Yüzdesi
- Sınıf
- Ortalama Çalışma Süresi
- Baba Hayatta
- Bursluluk
- İstanbul'da İkametgah Yeri
- Baba Eğitim Düzeyi

şeklinde sıralanmaktadır.

Bu çalışmanın sonuçları eğitimciler açısından faydalı bilgiler sunmaktadır ancak örneklem kümesinin büyütülmesi (1000 öğrenciye çıkış) ve çeşitlendirilmesi (mühendislik, hukuk, matematik vb eğitim alanların bileşimi) bulunan etkenlerin kesinlikle kullanılabilir olduğunu kanıtlayacaktır.

Üniversite eğitimi sürecinde hem öğrencilerin hem de akademisyenlerin dikkat etmesi gereken faktörleri ortaya koymaktadır. Sosyal açıdan öğrencilerin daha başarılı olabilmeleri için kalacakları yerlerin, maddi olanaklarının ve derse devamlarının iyileştirilmesi bu noktada yapılabilecek bir çabadır. Öğrencilerin başarıları üzerinde etkili olan bu faktörlerin iyileştirilmesi ile daha başarılı ve daha üretken kişiler kazanmak mümkün olacaktır.

KAYNAKLAR

1. **Bolat, S.**, 2003. Yapay Sinir Ağı ile Elektrot ve İzolatör Biçim Optimizasyonu, *Yüksek Lisans Tezi*, İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.
2. **Bölükbaşı İ.**, 2005. Defining Critical Factors Affecting Student Success: A Data Mining Approach , *Yüksek Lisans Tezi*, İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.
3. **Chaudhury, A, Stahl, D.O., Whinston, A.B.**, 1994. The Economic Foundation for Neural Computing Systems in, Advances in AI in Economics, Finance and Management. *Eds. Johnson J.D.*. Whinston A.B.. JAI Pres Inc.
4. **Chen, S., Mangiameli, P., West, D.**, 1995. The Comparative ağabeylity Of Sef-Organizing Neural Networks To Define Cluster Structure. *Omega Int. J. Mgmt. Sci.* 23/3, 271-279.
5. **Demuth, H. ve Beale M.**, Neural Network Toolbox For Use With MATLAB, User's Guide. *The Mathworks*, 1993.
6. **Efe, Ö., Kaynak, O.**, 2000. Yapay Sinir Ağları Uygulamaları, Boğaziçi Üniversitesi Yay., İstanbul.
7. **Gately, E.**, 1996. Neural Networks for Financial Forecasting, John Wiley& Sons, Inc, USA.
8. **Güzeliş, C.**, 1993. *Genelleştirilmiş Hücresel Yapay Sinir Ağları*, Elektrik Mühendisliği 5. Ulusal Kongresi, Trabzon.
9. **Hawley, D.D., Johnson, J. D.**, 1994. ANN: Past, Present and Future, in Advances in Ecnomics, Finance and Management, Eds. Jonhson J. D., Whisnton A.B..JAI Pres Inc.
10. **Haykin, S.**, 1999. Neural Networks: A Comprehensive Foundation Second Edition, prentice Hall, New Jersey.
11. **Hebb, D. O.**, 1949. Organizational Behaviour. John Wiley&Sons, Inc.
12. **Hecht-Nielsen, R.**, 1990. Neurocomputing, Addison Wesley.
13. <http://bagem.tripod.com/bagem/yz2.html>
14. <http://batitrakya.dostweb.com/yapaysiniraglari/yapaysiniraglari.htm>

15. <http://www.filesland.com/software/neural-network-1.html>
16. <http://www.mathworks.com/access/help desk /help /toolbox/nnet/>
17. http://www-ee.uta.edu/eeweb/ip/Software/OldSW/Linear_class.c
18. **Kuo, R.J., Chen, C.H., Hwang, Y.C.**, 2001 An Intelligent Stock Trading Support System Through integration of Genetic Algorithm Based Fuzzy Neural Network And Artificial Neural Network, *Fuzzy Sets And Systems*,
19. **Lee, C.W., Park, J-A.**, 2001. Assesment of HIV/AIDS-Related Hesalth Performance Using An Artificial Neural Network. Information & Management.
20. **Masters, T.**, 1993. Practical Neural Network Recipes in C++, Academic Pres, USA.
21. **May, G. S.**, 1994. Manufacturing ICs the Neural way. *IEEE Spectrum*. **31 (9)**: 47-51
22. **Moumin, Ali**, 1995. Yapay Sinir Ağları ve Genetik Algoritmalarının Uçuş Kontrol sistemlerine Uygulanması, *Yüksek Lisans Tezi*, İTÜ Fen Bilimleri Enstitüsü, İstanbul.
23. **Narendra, K.S. ve Parthasarathy, K.**, “Identification and Control Dynamic Systems Using Neural Networks”, *IEEE Trans. Neural Networks*, **vol. 1.**, pp. 4-27, Mar. 1990.
24. **Özen, Ü.Y.**, 2000. Modeling and Analyzing Strategic Thoughts Using Cognitive Mapping, *Ph.D. Thesis*, İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.
25. **Seven, A.**, Temmuz 1993. Yapay Sinir Ağları ile Doku Sınıflandırma, *Yüksek Lisans Tezi*, İ.T.Ü Fen Bilimleri Enstitüsü, İstanbul.
26. **Simpson, P. K.**, 1990. Artificial Neural Systems, Pergamon Pres.
27. **Smith, S. Palaniswami, M. Ve Krishnamoorthy, M.** , 1996. A Hybrid Neural Approach to Combinatorial Optimization. Computers and Operations Research: **23 (6)**
28. **Şahin, Ş. Ö.**, 2001, Yapay Sinir Ağları Yardımı ile Dinamik Bir Senaryo Analizi, *Doktora Tezi* İTÜ Fen Bilimleri Enstitüsü , İstanbul.
29. **Şeker, S.**, 2001. Artificial Neural Networks for System Identification and Control, Ders Notları
30. **Yoon, Y., Swales, G., Margavio, T.M.**, 1993. A Comparison of Discriminant Analysis Versus Artificial Neural Networks. *Journal of Operational Research*, **44/1**, 51-60.

31. **Yüksel, İ.** 1996. Matlab İle Mühendislik Sistemlerinin Analizi Ve Çözümü. U.Ü.Makine Mühendisliği Bölümü.
32. **Zurada, J. M.**, 1992. *Introduction to Artificial Neural Systems*, West Publishing,

EKLER

Ek-1 Anket

Öğrenci Numarası :

Bölüm :

Sınıf :

DEMOGRAFİK SORULAR

1. Doğum Tarihi (Yıl- Ay-Gün formatında yazın lütfen)

-- \ -- \ --

2. Cinsiyetiniz?

Bayan ☐ Bay ☐

3. Medeni durumunuz?

Evli ☐ Bekar ☐

4. Çocuk sahibi misiniz?

Evet ☐ Hayır ☐

5. Anneniz hayatta mı?

Evet ☐ Hayır ☐

6. Babanız hayatta mı?

Evet ☐ Hayır ☐

7. Doğum Yeriniz il ismi ve il plaka kodu (Örn : İstanbul / 34)

----- / --

8. En uzun süre yaşadığınız il ismi ve il plaka kodu

..... /

9. Lise öğrenimi aldığınız il ismi ve il plaka kodu

..... /

10. Annenizin Eğitim Durumu

Tahsili yok	
İlk okul	
Orta okul	
Lise	
Önlisans	
Lisans	
Yüksek lisans	
Doktora	

11. Babanızın Eğitim Durumu

Tahsili yok	
İlk okul	
Orta okul	
Lise	
Önlisans	
Lisans	
Yüksek lisans	
Doktora	

12. Kendinizden büyük Erkek / Kız kardeş sayınız nedir?

Erkek Kız.....

13. Kendinizden küçük Erkek / Kız kardeş sayınız nedir?

Erkek Kız.....

14. Türkiye Cumhuriyeti vatandaşı mısınız?

Evet ☐ Hayır ☐

15. Anneniz çalışıyor mu / çalıştı mı?

Evet ☐ Hayır ☐

16. Babanız çalışıyor mu / çalıştı mı?

Evet ☐ Hayır ☐

17. Anneniz ve Babanız beraberliği için şu anda hangisi geçerli? (Anne – Baba hayatta ise cevaplayın)

Evli	
Boşanmış	
Ayrı yaşıyorlar	

EĞİTİM

18. Mezun olduğunuz lise türü nedir?

Devlet lisesi	
Meslek lisesi	
Ticaret lisesi	
Anadolu lisesi	
Fen lisesi	
Özel lise	
Yurt dışı	
Devlet lisesi	

19. Lise mezuniyet ortalamanız nedir? (ÖRN : 4,30 (Sizin Lise Mezuniyet Ortalamanız) / 5,00 (Kullanılan skala) formatında yazın lütfen.

----- / -----

20. ÖSYM sınavına ilk girişinizde mi İTÜ'ye giriş hakkı kazandınız?

Evet ☐ Hayır ☐

21. Devam ettiğiniz bölüm ÖSYM tercih formunda kaçınca sıradaydı?

AKADEMİK

22. Okul kütüphanesinden kitap alıyor musunuz / faydalıyor musunuz?

Evet ☐ Hayır ☐

23. Akademik olmayan Kitap / Dergi / Gazete okuma sıklığınız nedir?

Hiç ☐ Nadiren ☐ Bazen ☐ Çok sık ☐

24. Akademik (Mesleki) Kitap / Dergi okuyor musunuz? (Ders kitapları dışında)

Evet ☐ Hayır ☐

25. Eğitim dalınızla ilgili, herhangi bir konferans veya Sempozyuma katıldınız mı?

Evet ☐ Hayır ☐

26. Lisansüstü eğitime devam etmeyi düşünüyor musunuz?

Evet ☐ Hayır ☐ Karar vermedim ☐

27. Bir önceki yıl sonu Cumulative GPA'nız nedir?

---,---

28. Mezuniyet sırasındaki tahmini Cumulative GPA'nız ne olabilir?

---,---

29. Okulda hazırladığınız proje / ödevleriniz eğitiminize katkıda bulunarak, sizi akademik ve iş hayatına hazırlıyor mu?

Evet ☐ Hayır ☐ Fikrim Yok ☐

30. Derslere genel olarak katılım sıklığınız için aşağıdakilerden hangi geçerli?

%70 ☐ %80 ☐ %90 ☐ %100 ☐

KİŞİSEL BİLGİLER

31. Part-time çalışıyor musunuz?

Evet ☐ Hayır ☐

32. Sinemaya ne sıklıkta gidiyorsunuz?

Hiç ☐ Ayda bir kez ☐ Haftada bir kez ☐ Hafta da bir kezden daha sık ☐

33. Spor yapıyor musunuz?

Evet ☐ Hayır ☐

34. Herhangi bir hobi / kişisel gelişim kursuna gidiyor musunuz?

Evet ☐ Hayır ☐

35. Ortalama günde kaç saat Internet kullanıyorsunuz?

İki saatten az ☐ İki – Dört saat arası ☐ Dört saatten fazla ☐

36. Ortalama günde kaç saat televizyon seyrediyorsunuz?

İki saatten az ☐ İki – Dört saat arası ☐ Dört saatten fazla ☐

37. Ortalama günde kaç saat ders çalışıyorsunuz?

İki saatten az ☐ İki – Dört saat arası ☐ Dört saatten fazla ☐

38. Okul dışında da okuldan arkadaşlarınızla görüşüyor musunuz?

Evet ☐ Hayır ☐

39. Okul kulüplerinde görev alıyor musunuz?

Evet ☐ Hayır ☐

40. İstanbul'da ikamet etme şekliniz aşağıdakilerden hangisidir?

Kendi aileniz ile	☐
Aile yanında	☐
Yurtta	☐
Evde (tek başına)	☐
Evde (ev arkadaşı ile)	☐

41. Okula ulaşımınız için aşağıdakilerden hangisi geçerlidir?

Özel araç ☐ Toplu Taşıma ☐ Yürüyerek ☐

42. Genel olarak, okula ulaşım süreniz için aşağıdakilerden hangisi geçerli ?

15 dakikadan az	☐
15 dakika – 30 dakika arası	☐
30 dakika – 1 saat arası	☐
Bir saatten fazla	☐

43. Kaldığınız yerdeki ders çalışma ortamınızdan memnun musunuz?

Memnun değilim ☐ Memnunum ☐ Çok memnunum ☐

44. Okul dışında da kullanabildiğiniz özel bilgisayar erişiminiz var mı?

Evet ☐ Hayır ☐

45. Ortalama aylık harcadığınız tutar?

< 200 YTL ☐ 201- 400 YTL arası ☐ 401 – 600 arası ☐ >600 YTL ☐

46. Burs alıyor musunuz?

Evet ☐ Hayır ☐

47. Harç kredisi alıyor musunuz?

Evet ☐ Hayır ☐

48. Mezun olunca istediğiniz zaman bir işe girebileceğinize inanıyor musunuz?

Evet ☐ Hayır ☐

49. Mezun olunca ne kadar süre içinde işe gireceğinizi tahmin ediyorsunuz?

Hemen ☐ İlk 6 ay içinde ☐ Bir yıl içinde ☐ Fikrim Yok ☐

50. İTÜ'de sosyal ortamı nasıl buluyorsunuz?

İyi ☐ Orta ☐ Kötü ☐

51. İTÜ'de en çok sevdiğiniz ders grubu aşağıdakilerden hangisidir?

Mesleki tasarım	☐
Temel mühendislik	☐
İnsan ve toplum bilimleri	☐
Temel bilimler	☐

52. Genel olarak İTÜ'de okumaktan memnuniyet dereceniz nedir?

Memnun değilim ☐ Memnunum ☐ Çok memnunum ☐

Ek- 2 Uygulama Parametreleri Türkçe İsim Karşılıkları Listesi

Department – Bölüm	X_Smaller_Sister - Yaşça Küçük Kız Kardeş Sayısı	Conferance_Involvement - Konferans Katılımı	School_Clubs - Üniversite Kulüplerine Katılım
Class – Sınıf	X_Total_Number_of_Brothers – Toplam Erkek Kardeş Sayısı	Master_Degree - Yüksek Lisans Kararı	Residence - İstanbul’da İkametgah Yeri
Gender - Cinsiyet	X_Total_Number_of_Sisters – Toplam Kız Kardeş Sayısı	Project_Aid - Okul Projeleri	School_Arrival - Okula Ulaşım Şekli
Father_Alive - Baba Hayatta	X_Total_Number_of_Sibling – Toplam Kardeş Sayısı	Attendance_Perc - Devamlılık Yüzdesi	Arrival_Duration - Ortalama Ulaşım Süresi
Birth_Region - Doğum Yeri/Bölge	Mother_Working - Anne Çalışma Durumu	Parttime_Working - Yarı Zamanlı Çalışma	Studying_Environment - Çalışma Çevresi
Live_Most_Region - En Uzun Süre Yaşanılan Yer/Bölge	High_School_Type - Lise Tipi	Cinema_Frequency - Sinemaya Gitme Sıklığı	Special_Computer_Usage - Bilgisayar Sahipliği
High_School_Region - Lise Öğrenimi Bölgesi	High_School_GPA2 - Lise Mezuniyet Notu	Sports - Spor Yapma Alışkanlığı	Average_Monthly_Expenditure - Aylık Ortalama Harcama
Mother_Edu - Anne Eğitim Düzeyi	Uni_Entrance_First- Üniversiteye Giriş (İlk Sınav)	Hobby_Course - Hobi/Kişisel Gelişim Kursu	Scholarship - Bursluluk
Father_Edu - Baba Eğitim Düzeyi	X_Order_of_Department_Selection- Bölüm Tercih Sırası	Internet_Hours - Ortalama İnternet Kullanımı	Fee_Credit - Öğrenim Kredisi
X_Bigger_Brother - Yaşça Büyük Erkek Kardeş Sayısı	Library_Usage - Kütüphane Kullanımı	TV_Hours - Ortalama Televizyon İzleme Süresi	Finding_Job - İş Bulma
X_Bigger_Sister - Yaşça Büyük Kız Kardeş Sayısı	Nonacademic_Interest - Akademik Olmayan Kitap/Dergi Okuma İlgisi	Studying_Hours - Ortalama Çalışma Süresi	Find_Job_Duration - İş Bulma Süresi
X_Smaller_Brother - Yaşça Küçük Erkek Kardeş Sayısı	Acedemic_Interest - Akademik Yayın Okuma İlgisi	Meeting_Uni_Friends - Üniversitedeki Arkadaşlarla Görüşme	ITU_Social_Environment - İ.T.Ü. Sosyal Çevresi
Most_Liked_Course_Group - En Çok Sevilen Ders Grubu	General_Pleasure_of_ITU - İ.T.Ü. ‘de Olmaktan Duyulan Memnuniyet		

Ek- 3 C Kodları

```

/*****
*****

Title   :      Linear Network (Classification)
Inputs  :      1. Input File Containing Training Data
(Infile)

                                2. No. of Inputs
(N)

                                3. Possible No. of Classes
(Nc)

Output :      classification error %

Notations Used:
Nv  --> No. of patterns
N   --> No. of inputs
Nout --> No. of outputs
X   --> Input Vector
Xa  --> Augmented input vector
Wo  --> Weight vector connecting input to the output
to  --> Output threshold
ty  --> Desired output
y   --> Actual output
R   --> Auto correlation Matrix
C   --> Cross correlation matrix
dstd --> desired mean
dstd --> desired standard deviation

*****
*****

! IMPORTANT: Significant changes have been made to the program. Have
a
                                thorough look at it.

*****
*****/

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>

#define PI 3.1415926
#define dmean 0.0
#define dstd 0.5

int IX=3
int IY=4009
int IZ=234

char Infile[14]

void main(void)
```

```

{

    int i, j
    int N, Nc, Nv
    float *X, *Xa, *ty
    float **Wo,*to, **W, **R, **C
    int ClassId
    FILE *ifs

    void CGrad(int ,float * ,float **,float *)
    void error(float **, int, int)
    void classification_error(float **, int, int)
    float slete(float, float)

    printf("Enter Input data file : ")
    gets(Infile)

    printf("Enter No. of Inputs, N : ")
    scanf("%d",&N)

    printf("Enter Possible no. of classes, Nc : ")
    scanf("%d",&Nc)

    ifs=fopen(Infile,"r")

    if (ifs == NULL ) {
        perror(Infile)
        exit(1)
    }

    /*      Allocating memory dynamically for the variables used
*/

    X= (float *)malloc(sizeof(float)*(N) )
    Xa= (float *)malloc(sizeof(float)*(N+1) )
    ty= (float *)malloc(sizeof(float)*Nc )
    W= (float **)malloc(sizeof(float*)*Nc)
    Wo= (float **)malloc(sizeof(float*)*Nc)
    to= (float *)malloc(sizeof(float)*Nc )
    R= (float **)malloc(sizeof(float*)*(N+1))
    C= (float **)malloc(sizeof(float*)*Nc)

    for (i=0i<N+1i++) {
        R[i]= (float*)malloc(sizeof(float)*(N+1))
    }

    for (i=0i<Nci++) {
        C[i]=(float* )malloc(sizeof(float)*(N+1))
    }

    for (i=0i<Nci++) {
        Wo[i]= (float*)malloc(sizeof(float)*N)
    }

    for (i=0i<Nci++) {
        W[i]= (float*)malloc(sizeof(float)*(N+1))
    }
}

```

```

/* Initializations */

/* Initialzing weights and thresholds to gaussian random
nos. */

for (i=0i<Nci++){

    to[i]=slete(dstd, dmean)

    for (j=0j<Nj++){
        Wo[i][j]=slete(dstd, dmean)
    }
}

/* W --> Weight vector containing Wo and to */

for (i=0i<Nci++){

    for (j=0j<Nj++){
        W[i][j]=Wo[i][j]
    }
    W[i][N]= to[i]
}

/* Initializing R and C with zeroes */

for (i=0i<N+1i++) {
    for (j=0j<N+1j++) {
        R[i][j]=0.0
    }
}

for (i=0i<Nci++) {
    for (j=0j<N+1j++) {
        C[i][j]=0.0
    }
}

Nv=0

/*      Here we find the auto and cross correlation matrices
*/

while (!feof(ifs))
{
    for (i=0i<N && !feof(ifs)i++) {
        fscanf(ifs,"%f",&X[i])
    }

    if (!feof(ifs) ) {
        fscanf(ifs,"%d", &ClassId)
    }
}

```

```

        for (i=0i<Ni++) {
            Xa[i]= X[i]
        }

        Xa[N]=1.0

        for (i=0i<Nci++) {
            ty[i]=0.0
        }

        ty[ClassId-1]=1.0

        /* The o/p node corr. to the class id is made equal
to 1 */
        /* all other nodes have desired o/p = 0 */

        /* Since C allocates memory starting from zero and
the classes are usually
named from 1 onwards, the node ty[Class_Id -1] is
made equal to 1 instead
of just ty[Class_Id] */

        /* Finding the auto-correlation */

        for (i=0i<N+1i++) {
            for (j=0j<N+1j++) {
                R[i][j]+=Xa[i]*Xa[j]
            }
        }

        /* Finding the cross-correlation */

        for (i=0i<Nci++) {
            for (j=0j<N+1j++) {
                C[i][j]+=ty[i]*Xa[j]
            }
        }

        Nv++

    }

    fclose(ifs)
    Nv--

    /*      Normalizing auto and cross correlations */

    for (i=0i<N+1i++) {
        for (j=0j<N+1j++) {
            R[i][j]/=Nv
        }
    }

    for (i=0i<Nci++) {
        for (j=0j<N+1j++) {
            C[i][j]/=Nv
        }
    }

```

/* e

```

/*      Calling Conjugate gradient Subroutine */
for (i=0i<Nci++) {
    CGrad(N+1, W[i], R, C[i])
}

/* calling "error" subroutine that calculates mapping error
*/
error(W, N, Nc)

/* Calling subroutine that calculates the classification
error */
classification_error(W, N, Nc)

printf("\n\n\n\n\nTraining Complete!")

} // End of main

/*****
*****

Title      : Subroutine- Conjugate gradient
Comments : Follow the derivation done in the class for better
understanding of
                this routine

*****
*****/

void CGrad(int Nu,float *w,float **r,float *c)
{
    float XD, XN, Num, Den, tempg, B1, B2
    float *p, *g
    int l, m , iter, pass

    XD = 1

    p= (float *)malloc(sizeof(float)*Nu)
    g= (float *)malloc(sizeof(float)*Nu)

    /* passing through the CG routine two times for improved
results */
    for (pass =0pass<2pass++)
    {
        for(l=0l<Nul++) {

```

```

        p[l] = 0.0
        g[l] = 0.0
    }

    for(iter= 0 iter<Nu iter++)
    {
        for(l=0l<Nul++) {
            tempg = 0.0

            for (m=0m<Num++) {
                tempg += w[m]*r[m][l]
            }
            /* gradient vector */
            g[l] = -2.0*c[l]+2.0*tempg
        }
        /*
end of l -loop*/

        XN = 0

        for(l=0l<Nul++) {
            XN += (g[l]*g[l])
        }

        B1 = XN/XD
        XD = XN

        /* direction vector */
        for(l=0l<Nul++) {
            p[l] = -g[l]+B1*p[l]
        }

        Den = Num = 0.0

        for(l=0l<Nul++)
        {
            /* Num --> numerator of B2 */
            Num += p[l]*g[l]/-2.0

            /* Den --> denominator of B2 */
            for (m=0m<Num++) {
                Den += p[m]*p[l]*r[m][l]
            }
        }
        /* e

        B2 = Num/Den

        for(l=0l<Nul++) {
            w[l]+= B2*p[l]
        }

    }

}

/* end of CjGrad */

```

```

/***** Subroutine :error
*****

This subroutine determines the difference between the desired and
obtained
output. This requires passing through the data once.

*****/

void error(float **W, int N2, int Nout2)
{

    float *Error, y
    float *x, *xa, *ty
    int i, j
    int classid
    FILE *ifs1
    float MSE =0.0
    int Nv=0

    /* memory allocation */

    x= (float *)malloc(sizeof(float)*(N2))
    ty= (float *)malloc(sizeof(float)* Nout2)
    xa= (float *)malloc(sizeof(float)*(N2+1))
    Error= (float *)malloc(sizeof(float)* Nout2)

    /*opening the input file in the read mode */
    ifs1 = fopen(Infile,"r")

    /* checking for the existance of the file */
    if ( ifs1 == NULL ) {
        perror(Infile)
        exit(1)
    }

    /* Initializing error to zeroes */
    for (i=0i<Nout2i++) {
        Error[i]=0.0
    }

    while (!feof(ifs1) )
    {

        Nv++
        for(i=0i<N2 && !feof(ifs1)i++) {
            fscanf(ifs1,"%f",&x[i])
        }

        if ( ! feof(ifs1) ) {
            fscanf(ifs1,"%d",&classid)
        }

        for (i=0i<N2i++) {
            xa[i] =x[i]
        }
    }
}

```

```

        for (i=0i<Nout2 i++) {
            ty[i]=0.0
        }

        ty[classid-1]=1.0

        xa[N2]=1.0
        /* Threshold */

        /* for each set of input features, calculating and
accumulating error */
        for (i=0i<Nout2i++)
        {

            y = 0.0

            for(j=0j<N2+1j++) {
                y += xa[j]*W[i][j]
            }

            Error[i]+= (ty[i]-y)*(ty[i]-y)
        }

    }
    fclose(ifs1)
    Nv--

    /* printing errors at each node */

    printf("\nMapping error:\n")
    printf("-----\n")

    for (i=0i<Nout2i++) {
        MSE+=Error[i]/Nv
        printf("Node %d: Error = %f\n", (i+1),Error[i]/Nv)
    }

    printf("-----\n")
    printf("Total MSE = %f\n",MSE)
    printf("-----\n\n")
}

```

```

/*****
*****
Title : subroutine- Classification error
Comments : This subroutine calculates the classification error
percentage

*****/

```

```

void classification_error(float **W1, int N1, int Nc1)

```

```

{
    float *y
    float Max_y
    int Class_obtained
    int Class_err=0
    float *X
    int ClassId
    int i, j
    FILE *ifsc
    int Nv
    float percentage_error
    Nv=0

    /* Memory allocation */
    X=( float *)malloc(sizeof(float)*N1)
    y= (float *) malloc ( sizeof(float) *Nc1)

    for (i=0i<Nc1i++)
        y[i]=0.0

    ifsc = fopen(Infile,"r")

    while ( !feof(ifsc) )
    {

        Nv++

        for (i=0i<N1 && !feof(ifsc)i++) {
            fscanf(ifsc,"%f", &X[i])
        }

        X[N1]=1.0

        if ( !feof(ifsc) ) {
            fscanf(ifsc,"%d",&ClassId)
        }

        for (i=0i<Nc1i++) {
            y[i]=0.0
        }

        for (i=0i<Nc1i++) {
            for (j=0j<N1+1j++) {
                y[i]+= X[j]*W1[i][j]
            }
        }

        /* Here we find the maximum of the outputs:
        The node with the maximmm o/p represents the desired
        class */

        Max_y= y[0]
        Class_obtained=1

        for (i=1i<Nc1i++)
        {
            if(y[i]>Max_y) {
                Max_y= y[i]
                Class_obtained= i+1
            }
        }
    }
}

```

```

/*      Checking whether the obtained class is equal to the
desired class or not.
      if they are not equal it means that an error has
occurred. so we increment
      the classification error count by 1 */

      if ( ClassId != Class_obtained ) {
          Class_err += 1
      }

}

/* calculating classification error% */
percentage_error=((float)Class_err/(float)Nv)*100.0

printf("\nclassification error = %d\n",Class_err)
printf("\nclassification error percentage = %f
\n",percentage_error)

fclose(ifsc)

} // end of  classification error subroutine


/*****
*****/

Subroutine : Random no. Generator
      -the random nos. generated are uniformly distributed
      between 0 and 1.
      -IX, IY, IZ are the SEEDS

*****/

float rand1(int *ix, int *iy, int *iz)
{
    int ixx, iyy, izz
    float itemp
    float temp

    ixx=(*ix)/177
    *ix=171*(*ix%177)-2*ixx

    if(*ix < 0)
        *ix+=30269

    iyy=(*iy)/176
    *iy=176*(*iy%176)-2*iyy

    if(*iy < 0)
        *iy+=30307

    izz=(*iz)/178

```

```

        *iz=170*(*iz%178)-2*izz

        if(*iz < 0)
            *iz+=30323

        temp=(float) (*ix)/30629.0+(float) (*iy)/30307.0+(float) (*iz)/
30323.0
        itemp=floor(temp)
        return (temp-itemp)

    }

```

```

/* Gaussian Random No. generator */

float slete(float std, float xmean)
{
    float rand1(int *, int *, int *)
    return (xmean+std*cos(2*PI*rand1(&IX, &IY, &IZ))*sqrt(-
2.0*log(rand1(&IX, &IY, &IZ))))
}

```

ÖZGEÇMİŞ

1982 yılında, Eskişehir'in Sivrihisar ilçesinde doğdu. İlköğrenimini Ali Fuat Cebesoy İlköğretim Okulu'nda ve liseyi Özel Ertuğrulgazi Lisesi'nde tamamladı. 2000 yılında İ.T.Ü. Endüstri Mühendisliği bölümünü kazandı. 2004 yılında Endüstri Mühendisliği'nden mezun oldu. Aynı yıl İ.T.Ü. Fen Bilimleri Enstitüsü Endüstri Mühendisliği Ana Bilim Dalı Mühendislik Yönetimi Programı'nda Yüksek Lisans eğitimine başladı. 2004 yılından bu yana Bankacılık sektöründe çalışmaktadır. Sektör içerisinde Bireysel Bankacılık Ürün Geliştirme Uzmanı olarak Kredi Kartları ve Banka Kartları, Kredili Mevduat Hesabı ve Bireysel Krediler alanlarında görev aldı. 2006 Temmuz ayından itibaren Yapı ve Kredi Bankası A.Ş.'de Kredi Kartları ve Tüketici Kredileri Pazarlama Bölümü'nde Ürün Geliştirme görevindedir.