

**TC.
GAZİ ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
İŞLETME ANABİLİM DALI
ÜRETİM YÖNETİMİ BİLİM DALI**

**YAPAY SİNİR AĞLARI VE BEKLEME SÜRESİNİN
TAHMİNİNDE KULLANILMASI**

YÜKSEK LİSANS TEZİ

**Hazırlayan
Muhammet DEVECİ**

**Tez Danışmanı
Prof. Dr. Abdullah ERSOY**

Ankara - 2012

**TC.
GAZİ ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
İŞLETME ANABİLİM DALI
ÜRETİM YÖNETİMİ BİLİM DALI**

**YAPAY SİNİR AĞLARI VE BEKLEME SÜRESİNİN
TAHMİNİNDE KULLANILMASI**

YÜKSEK LİSANS TEZİ

**Hazırlayan
Muhammet DEVECİ**

**Tez Danışmanı
Prof. Dr. Abdullah ERSOY**

Ankara – 2012

ONAY

Muhammet Deveci tarafından hazırlanan “Yapay Sinir Ağları ve Bekleme Süresinin Tahmininde Kullanılması” başlıklı bu çalışma, 14.08.2012 tarihinde yapılan savunma sınavı sonucunda oybirliği ile başarılı bulunarak jürimiz tarafından İşletme Anabilim Dalı, Üretim Yönetimi Bilim Dalında Yüksek Lisans Tezi olarak kabul edilmiştir.

[İmza]

.....

PROF. DR. ABDULLAH ERSOY (Başkan)

[İmza]

.....

PROF. DR. NURETTİN PARILTI

[İmza]

.....

YRD. DOÇ. DR. CEMALETTİN AKTEPE

ÖNSÖZ

Bu çalışmanın gerçekleştirilmesinde, beni her zaman destekleyen ve yol gösteren başta tez danışmanım değerli hocam Prof. Dr. Abdullah ERSOY olmak üzere, tüm değerli hocalarıma ve tüm arkadaşlarıma içtenlikle teşekkür ederim.

Bu günlere gelene kadar hep yanımda olan, büyük özverilerde bulunarak bana sabır ve anlayış gösteren sevgili aileme sonsuz teşekkürlerimi sunarım.

İÇİNDEKİLER

ÖNSÖZ	i
İÇİNDEKİLER	ii
KISALTMALAR	vi
TABLolar DİZİNİ	vii
ŞEKİLLER DİZİNİ	viii
GİRİŞ	1

BİRİNCİ BÖLÜM

YAPAY SİNİR AĞLARINA GİRİŞ

1.1. YAPAY SİNİR AĞLARI	3
1.1.1. Genel Bakış	3
1.1.2. Yapay Sinir Ağların Tarihçesi	5
1.1.3. Yapay Sinir Ağının Tanımı	7
1.1.4. Biyolojik Bir Sinir Hücresinin Yapısı	8
1.1.5. Yapay Sinir Ağlarının Genel Özellikleri	10
1.1.6. Yapay Sinir Ağlarının Avantajları	12
1.1.7. Yapay Sinir Ağlarının Dezavantajları	12
1.2. YAPAY SİNİR HÜCRESİ.....	14
1.2.1. Girdiler	15
1.2.2. Ağırlıklar	15
1.2.3. Toplam Fonksiyonu	15
1.2.4. Aktivasyon (Transfer) Fonksiyonu	16
1.2.5. Çıkış İşlevi	18
1.2.5.1. Yapay Sinir Hücresinin Çalışmasına Bir Örnek.....	18
1.3. BİR YAPAY SİNİR AĞI'NIN YAPISI	19

İKİNCİ BÖLÜM

ÇOK KATMANLI ALGILAYICILAR (ÇKA)

2.1. ÇOK KATMANLI ALGILAYICI VE YAPISI	23
--	----

2.2. ÇOK KATMANLI ALGILAYICILARDA ÖĞRENME KURALI	24
2.2.1. İleriye Doğru Hesaplama	25
2.2.2. Geriye Doğru Hesaplama	26
2.3. ÇOK KATMANLI ALGILAYICILARIN ÇALIŞMA PROSEDÜRÜ.....	27
2.4. YAPAY SİNİR AĞLARININ SINIFLANDIRILMASI	27
2.4.1. Yapılarına Göre Yapay Sinir Ağları.....	28
2.4.2. Öğrenme Algoritmalarına Göre Yapay Sinir Ağları	30
2.4.2.1. Danışmalı (Eğitici) Öğrenme	30
2.4.2.1.1. Perceptron (İdrak-Algılayıcı)	31
2.4.2.2. Danışmasız (Eğitici) Öğrenme.....	32
2.4.2.3. Takviyeli (Destekleyici) Öğrenme	33
2.4.2.4. Karma Stratejiler	34
2.4.2.4.1. Radyal Tabanlı Yapay Sinir Ağları	34
2.4.3. Öğrenme Zamanına Göre Yapay Sinir Ağları	36
2.4.3.1. Statik Öğrenme.....	37
2.4.3.2. Dinamik Öğrenme.....	37
2.5. YAPAY SİNİR AĞLARINDA ÖĞRENME KURALLARI	37
2.5.1. Hebb Kuralı.....	38
2.5.2. Delta Kuralı	38
2.5.3. Hopfield Kuralı	39
2.5.4. Kohonen Kuralı	40
2.5.5. Geri Yayılım Ağı.....	41
2.5.6. Levenberg – Marquardt Öğrenme Algoritması.....	42
2.5.7. LVQ (Öğrenmeli Vektör Kuantalama)	45
2.5.7.1. LVQ Ağı'nın Yapısı	45
2.5.8. Hızlı Yayılım	46
2.5.9. Eşleştirmeli Eğilim Algoritmaları.....	47
2.5.10. Kuasi-Newton Öğrenme Algoritması.....	47
2.5.11. Online Geri Yayılım.....	48
2.5.12. Grup Geri Yayılım Algoritması	48
2.6. YAPAY SİNİR AĞLARIN UYGULAMA ALANLARI.....	48
2.7. YAPAY SİNİR AĞLARINA YÖNELTİLEN ELEŞTİRİLER	51

2.8. YAPAY DİĞER METOTLARLA KARŞILAŞTIRILMASI	52
2.9. YAPAY SİNİR AĞLARI UYGULAMA ÖRNEKLERİ	54
2.9.1. Endüstriyel Uygulama	54
2.9.1.1. Problemin Tanımlanması	54
2.9.1.2. Öğrenme Setinin Oluşturulması	55
2.9.1.3. ÇKA Ağının Oluşturulması	57
2.9.2. İmalat Sektöründe Kalite Kontrol Uygulaması	58
2.9.2.1. LVQ-Örüntü Tanıma	58
2.9.2.1.1. Problemin Tanımlanması	58
2.9.2.1.2. Öğrenme Setinin Oluşturulması	61
2.9.2.1.2.1. Artan veya Azalan Trendin Üretilmesi	61
2.9.2.1.2.2. Normal Dağılım Şeklin Üretilmesi	62
2.9.2.1.2.3. Yukarı veya Aşağı Doğru Ortalamadan Sapma	62
2.9.2.1.3. LVQ Ağının Oluşturulması	62
2.9.3. Banka'da Müşteri Bekleme Süresi Öngörüsü	64
2.9.4. Günlük Sıcaklık Tahmini	67

ÜÇÜNCÜ BÖLÜM

UYGULAMA

3.1. PROBLEMİN TANIMLANMASI.....	73
3.2. ARAŞTIRMANIN AMACI VE YÖNTEMİ	74
3.2.1. Çalışmanın Yöntemi	75
3.2.2. Girdi ve Çıktıların Nümerik Gösterimi	76
3.2.3. Yapay Sinir Ağı Yapısının Seçimi	79
3.2.4. Başlangıç Değerlerinin Atanması	79
3.2.5. Öğrenme Katsayısı ve Momentum Katsayılarının Belirlenmesi ...	80
3.2.6. Normalizasyon İşlemi	80
3.2.6.1. İstatiksel veya Z-Değeri Normalizasyonu	81
3.2.6.2. Min-Maks Normalizasyonu.....	82
3.2.6.3. Medyan Normalizasyonu	82
3.2.6.4. Sigmoid Normalizasyonu	83

3.2.6.5. D-Min-Maks Normalizasyonu.....	83
3.3. ALYUDA İLE YSA BEKLEME SÜRESİ TAHMİNİ	84
SONUÇ ve ÖNERİLER	90
KAYNAKÇA	92
ÖZET	100
ABSTRACT	101

KISALTMALAR

ADALINE	: Yinelemeli Doğrusal Sinir Ağı
ART	: Adaptif Rezonans Teorisi
BM	: Bulanık Mantık
BP	: Geri Yayılım
ÇKA	: Çok Katmanlı Algılayıcılar
GA	: Genetik Algoritma
GRNN	: Genel Regresyon Ağları
LM	: Levenberg-Marquardt
LMSR	: En Küçük Kareler Yöntemi
LVQ	: Öğrenmeli Vektör Kuantalama
MADALINE	: Çoklu Yinelemeli Doğrusal Sinir Ağı
MAPE	: Ortalama Mutlak Yüzde Hata
MSE	: Hata Kareleri Ortalaması
PBNN	: Olasılıksal Tabanlı Sinir Ağları
PNN	: Probabilistik Ağlar
RBF	: Radyal Temelli Fonksiyon
RBN	: Radyal Tabanlı Sinir Ağları
RTFA	: Radyal Taban Fonksiyonlu Ağlar
RTFYSA	: Radyal Tabanlı Fonksiyon Yapay Sinir Ağı
SOM	: Kendi Kendini Haritalama
YSA	: Yapay Sinir Ağları

TABLOLAR DİZİNİ

Tablo 1.1. Biyolojik Sinir Ağı ile Yapay Sinir Ağının Karşılaştırılması	10
Tablo 1.2. Yapay Sinir Hücresinin Girdi ve Ağırlık Değerleri	19
Tablo 2.1. Öğrenme Algoritmaları ve Kullanım Alanları Tablosu	36
Tablo 2.2. Yapay Sinir Ağları ve Uygulama Alanları	51
Tablo 2.3. YSA'ların İstatiksel Yöntemler ile Terminolojik Benzeşimi	52
Tablo 2.4. YSA ve Geleneksel Talep Tahmini Metotların Karşılaştırılması .	53
Tablo 2.5. Uzman Sistemler ve Yapay Sinir Ağları Arasındaki Farklılıklar	53
Tablo 2.6. Piston Üzerindeki Sehime Neden olan Faktörler ve Değerleri	55
Tablo 2.7. Öğrenme Seti	55
Tablo 2.8. Örneklerin Nümerik Gösterimi	56
Tablo 2.9. Ölçeklendirilmiş Eğitim Seti	56
Tablo 2.10. LVQ Ağının Parametreleri	64
Tablo 2.11. Bekleme Süresini Etkileyen Faktörler	64
Tablo 2.12. Eğitim Seti	65
Tablo 2.13. Eğitim Setinin Nümerik Gösterimi	66
Tablo 2.14. YSA Girdi Parametreleri	68
Tablo 2.15. Modelde Kullanılan Parametreler ve Özellikleri	68
Tablo 2.16. Hava sıcaklığı Eğitim Verileri	69
Tablo 2.17. En İyi İstatiksel Sonuçlar	70
Tablo 3.1. Bekleme Süresini Etkileyen Faktörler	74
Tablo 3.2. Modelde Kullanılan Parametreler	74
Tablo 3.3. Bekleme Süresine Etki Eden Faktörler ve Değerleri	77
Tablo 3.4. Verilerin Nümerik Değer ile Gösterimi	78
Tablo 3.5. Veri Setinin Sınıflandırılması	84
Tablo 3.6. Öğrenme Algoritmasına Göre Sistem Çıktıları 1	85
Tablo 3.7. Öğrenme Algoritmasına Göre Sistem Çıktıları 2	86
Tablo 3.8. Öğrenme Algoritmasına Göre Sistem Çıktıları 3	87
Tablo 3.9. En iyi MAPE (%) ve MSE	87

ŞEKİLLER DİZİNİ

Şekil 1.1. Basit Nöron Modeli	4
Şekil 1.2. Biyolojik Sinir Hücresi ve Bileşenleri	9
Şekil 1.3. Sinir Ağında Parellelik.....	11
Şekil 1.4. Yapay Sinir Hücresi	14
Şekil 1.5. Yapay Sinir Ağları için Kullanılan Eşik Fonksiyonları	17
Şekil 1.6. Transfer Fonksiyonun Çalışma Yapısı	18
Şekil 1.7. Basit Yapay Sinir Hücresi Örneği	18
Şekil 1.8. Basit Tek Katmanlı Algılayıcı Modeli.....	19
Şekil 1.9. Çok Katmanlı Bir Yapay Sinir Ağı	21
Şekil 2.1. Çok Katmanlı Algılayıcı Model	24
Şekil 2.2. Yapay Sinir Ağlarının Ağ Yapıları	28
Şekil 2.3. İleri Beslemeli Nöron Ağı	29
Şekil 2.4. Geri Beslemeli Nöron Ağı	30
Şekil 2.5. Danışmalı Öğrenme Yapısı	31
Şekil 2.6. Algılayıcı Modeli Örnekleri	32
Şekil 2.7. Danışmansız Öğrenme Yapısı.....	33
Şekil 2.8. Takviyeli Öğrenme Yapısı.....	34
Şekil 2.9. Radyal Taban Fonksiyonlu Ağ Yapısı.....	35
Şekil 2.10. Bir Hopfield Ağı Örneği	40
Şekil 2.11. Geri Yayılım Ağı.....	42
Şekil 2.12. Hızlı ve Yüksek Performanslı Algoritmalar.....	43
Şekil 2.13. LVQ Ağının Topolojik Yapısı.....	46
Şekil 2.14. Sehim Tahmini için ÇKA Modeli	57
Şekil 2.15. İstatiksel Kalite Kontrol Sistemi.....	60
Şekil 2.16. LVQ Ağının Tanıdığı Kontrol Şeması Üzerindeki Şekiller	61
Şekil 2.17. LVQ Ağ Topolojisi	63
Şekil 2.18. Kuyruk Modeli için Çok Katmanlı Ağın Gösterimi.....	67
Şekil 2.19. Yöntem Sonuçlarının MAPE(%) Karşılaştırmaları.....	70
Şekil 2.20. D-Min-Maks Normalizasyon Yöntemi için Regresyon Analizi	71
Şekil 2.21. D-Min-Maks Normalizasyon Yöntemi için Performans Eğrisi	71

Şekil 2.22. D-Min-Maks Normalizasyon Yöntemi için Eğitim Durumu.....	72
Şekil 3.1. Başlangıç Değerlerin Çok Katmanlı Ağlara Etkisi	79
Şekil 3.2. Öğrenmenin Hata Uzayındaki Gösterimi	88
Şekil 3.3. Gerçek Bekleme Süresi ile Tahmini Sürelerin Karşılaştırılması...	88
Şekil 3.4. Ağ Eğitim Raporu	89
Şekil 3.5. Eğitim Grafiği	89
Şekil 3.6. Eğitim Grafiği Parametreleri.....	89

GİRİŞ

İnsan beyni hakkındaki çalışmalar binlerce yıl öncesine dayanır. Modern elektroniğin gelişmesiyle birlikte, bu düşünce işlemini kullanmaya başlamak doğal bir hale gelmiştir. İlk yapay sinir ağ modeli 1943 yılında, bir sinir hekimi olan Warren McCulloch ile bir matematikçi olan Walter Pitts tarafından gerçekleştirilmiştir. McCulloch ve Pitts, insan beyninin hesaplama yeteneğinden esinlenerek, elektrik devresiyle basit bir sinir ağını modellemiştir. 1948 yılında Wiener "Cybernetics" isimli kitabında sinirlerin çalışması ve davranış özelliklerine değindi. 1949 da ise Hebb "Organization of Behavior " isimli kitabında, öğrenmeyle ilgili temel teoriyi ele aldı (Elmas, 2003).

Yapay sinir ağları, insan beynindeki sinir hücrelerinin (nöron) bilgisayar ortamında matematiksel modellenmiş durumudur. Yapay sinir ağlarında kullanılan öğrenme algoritmaları, klasik bilgisayar algoritmalarından farklıdır. Bu algoritmalar insan beyninin sezgisel gücünü içinde taşırlar. Bu sebeple, birçok bilim dalı yapay sinir ağlarıyla ilgilenmektedir. Genellikle, yapay sinir ağlarıyla oluşturulan modeller zaman serisi analizi, optimizasyon (eniyileme), sınıflandırma, ilişkilendirme veya doğrusal olmayan sistem modellemelerinde kullanılmaktadır (Warner ve Misra, 1996).

Bir sinir ağının en temel bileşeni olan öğrenme yeteneğini, insan beyninin yapısından esinlenerek geliştirilmelerine karşın, sinir ağının bazı özellikleri insan beyniyle aynı değildir. Bazı özelliklerinin ise beyinde biyolojik karşılığı yoktur. Bununla birlikte sinir ağları biyolojik beyinle büyük benzerlikler gösterirler (Elmas, 2003).

Yapay sinir ağları son yıllarda oldukça ilgi gören bir modelleme tekniğidir. Biyolojik sinir sisteminin çalışma düzenini örnek alarak işler. Bir yapay zeka tekniği olan Yapay Sinir Ağı (YSA); Şekil tanıma, sinyal filtreleme, öğrenme, sınıflandırma, analiz, tahminleme ve optimizasyon konularında kullanılan etkili bir tekniktir. Geleneksel yöntemler, eksik veya hatalı veriler için uygun değildir. Öte yandan, YSA yaklaşımı ise, verilere

bağlı olmayıp; eksik kısmen hatalı veya aşırı sapmalı verileri değerlendirebilir, hatta karmaşık ilişkileri öğrenebilir, genelleyebilir ve bu sayede daha önce hiç karşılaşmadığı sorulara kabul edilebilir bir hatayla cevap bulabilir (Öztemel, 2003).

Bu çalışmada, yeni bir yöntem olan Yapay Sinir Ağları konusu incelenmekte, öngörü modellemesi tekniği olarak hastanede hastanın kuyrukta bekleme süresini etkileyen değişkenlere uygulanmakta ve başka algoritmalar ile karşılaştırmalı olarak performans değerlendirilmesi yapılmaktadır. Doğru tahminin veya öngörünün başarılı kararları ve beraberinde getireceği maksimum fayda sağlayabileceği gerçeği, öngörü modellemesine olan ilgiyi artırmaktadır. Bu yoğun ilgiyle birlikte, sinir ağları alanında önemli gelişmeler olmaktadır. Ortaya çıkan gelişmelerden bir tanesi öngörü doğruluğuna yönelik çalışmaların artış göstermesidir. Öngörü modellemesinde kullanılabilecek yöntemlerin çeşitliliği, model seçiminde bazı zorlukları da beraberinde getirmiştir. Amaç öngörü performansını artırmak olduğundan model seçiminde zaman zaman teorik kriterler yerine öngörü performansına yönelik kriterler ön plana çıkmaktadır. Öngörü doğruluğuna yönelik çalışmalar hızla yoğunlaşmıştır. Diğer taraftan, alternatif bir yöntem olarak farklı modellerden alınan öngörülerin birleştirilmesiyle öngörü kalitesinin yükseltilmesine yönelik çalışmalar da bulunmaktadır.

Bu kapsamda, çalışmanın birinci bölümünde yapay sinir ağlarıyla ilgili tanımlamalar yapılmış, özellikleri açıklanmış ve sinir ağının yapısından bahsedilmiştir. İkinci bölümde çok katmanlı ağlar, yapay sinir ağların sınıflandırılması ve sinir ağlarının öğrenme kurallarından bahsedilmiştir. Üçüncü bölümde ise gerçekleştirilen uygulama hakkında bilgi verilmiş, elde edilen sonuçlar tablolarla sunulmuş ve tez çalışması özetlenerek sonuç ve değerlendirme yapılmıştır.

BİRİNCİ BÖLÜM

YAPAY SİNİR AĞLARI VE TEMEL ELEMANLARI

1.1. YAPAY SİNİR AĞLARI

1.1.1. Genel Bakış

Yapay sinir ağları, insan beyninin özelliklerinden esinlenerek; öğrenme yoluyla yeni bilgiler türetebilme, keşfetme ve oluşturma gibi yetenekleri herhangi bir yardım almadan otomatik olarak gerçekleştirmek amacıyla geliştiren bilgisayar sistemleridir. Bu yeteneklerin gerçekleştirilmesi geleneksel programlama yöntemleriyle çok zordur. Bu nedenle, yapay sinir ağlarını mümkün olmayan olaylar veya programlanması çok zor olan olaylar için geliştirilmiş bilgi işlemeye ilgilene bir bilgisayar bilim dalı olduğı söylenebilir (Öztemel, 2003: 29). YSA karmaşık yapıda içsel ilişkilere sahip ya da aralarındaki bilinmeyen ilişki problemlerine çözüm bulabilmek için geliştirilen bir yapay zekâ tekniğidir.

Yapay sinir ağları bir programcının yeteneklerini gerektirmeyen, kendi kendine öğrenen sistemlerdir. Bu ağlar öğrenme gibi yeteneklerinin yanı sıra, ezberleme ve bilgiler arasında bağlantı veya ilişki kurma yeteneğine de sahiptirler. Yapay sinir ağlarının birçok çeşidi bulunmakla birlikte bunların bazılarının kullanımı diğerlerine göre daha yaygındır. Bunlardan bir tanesi geriye yayılımlı yapay sinir ağı olarak bilinendir. Bu tip ağları özellikle tahmin ve sınıflandırma işlemlerinde başarılı sonuçlar vermektedir. Kullanımı fazla olan bir başka yapay sinir ağı “Kohonen Özürgütlemeli” dir. Bu tip yapay sinir ağları, karışık bilgi kümeleri arasında ilişki bulma konusunda başarılı sonuçlar vermektedir (Elmas, 2003: 24).

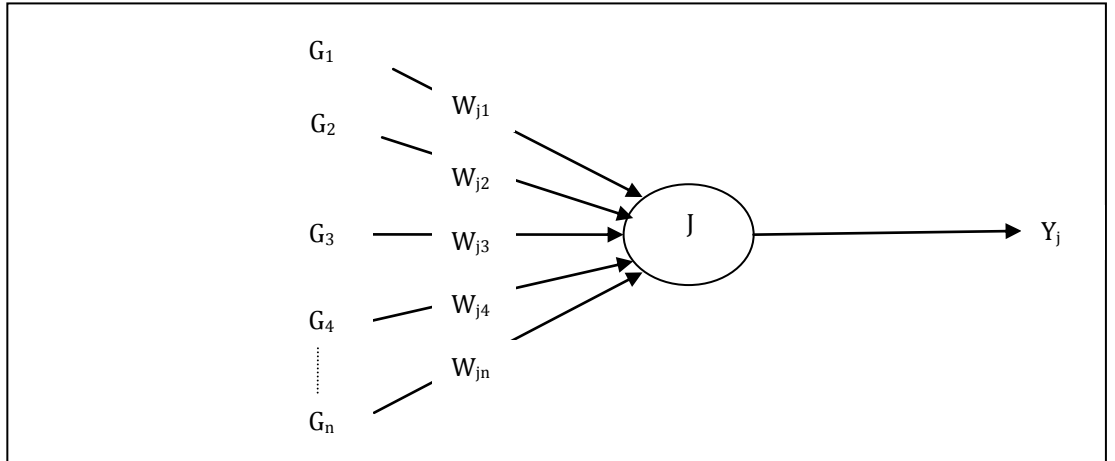
Yapay sinir ağları eksik, hatalı ve karmaşık bilgileri işlemedeki başarıları nedeniyle birçok problem çözümünde uygulanmaktadır. Yapılarındaki paralellikten dolayı çok hızlı çalışabilmeleri, onların özellikle gerçek zamanlı olaylarda çalışmalarını sağlamaktadır

İnsan beyninin fonksiyonel özelliklerine benzer şekilde,

- Öğrenme,
- İlişkilendirme,
- Sınıflandırma,
- Modelleme,
- Tahmin,
- Özellik belirleme,
- Optimizasyon,

gibi konularda başarılı bir şekilde uygulanmaktadır (Öztemel, 2003).

Basit bir nöron modeli Şekil 1.1'de görülmektedir. Burada G_n girdileri ($n=1, 2, 3, \dots, N$), w_{ij} ağırlık katsayılarını ($n=1, 2, 3, \dots, N$; $j=1, 2, 3, \dots, M$) ve Y_j çıktıları göstermektedir.



Şekil 1.1. Basit Nöron Modeli

1.1.2. Yapay Sinir Ağların Tarihçesi

İnsan beyniyle ilgili çalışmalar yüz yıl öncesine dayanır. 1890 yılında William James “Psychology (Briefer Course)” adlı yayınıyla insan beyni hakkında bilgi veren ilk eser yayınlamıştır (James, 1890: 253-279). Daha sonra ilk yapay sinir ağ modeliyle ilgili 1943 yılında, bir sinir hekimi olan Warren McCulloch ile bir matematikçi olan Walter Pitts tarafından yayınlanmış olan “A Logical Calculus of İdea İmmanent in Nervous Activity” adlı bir eser ortaya koyuluyor. Bu eser sinir aktivitelerin, sinirsel olaylar ve aralarındaki ilişkilerin “ya hep ya hiç” önerme mantığıyla tedavi edilebilir olduğunu söylüyordu. Ayrıca McCulloch ve Pitts, insan beyninin özelliklerini dikkate alarak, elektrik devresiyle basit bir sinir ağını modellemiştir (McCulloch ve Pitts, 1943: 115-133).

Amerikan matematikçisi olan Norbert Wiener 1948 yılında “Cybernetics” isimli kitabında insan ve makineler arasındaki ilişkiler, iletişim sistemi, sinirlerin çalışması ve davranış özelliklerine değindi (Wiener, 1948). Bu yıldan sonra 1949 da ise Hebb “Organization of Behavior (Organizasyonların Davranışı)” isimli eserinde sinir substratı hakkında çeşitli öğrenme ve bellek hipotezlerini tanıttı (Hebb, 1949: 45-78). Rosenblatt tarafından 1958 yılında “Algılayıcı (Probabilistik Modeller ve İnsan Beynin Organizasyonu)” önerilmiş ve bilinen tek katmanlı ilk yapay sinir ağı modeli ortaya çıkmıştır (Rosenblatt, 1958: 386-408).

1982 yılında Grossberg “Zeka ve Akıl Çalışmaları” ve 1987 yılında “Adaptif Rezonans Teorisi 2 (ART2)” hakkında eserler yayınlayarak öğretmensiz (danışmasız) öğrenme olan bir YSA yapısını geliştirmiştir (Carpenter ve Grossberg, 1987).

Sinir ağında oluşturulan algoritmalar insan beyninin sezgisel gücünü içinde barındırırlar. Bu yüzden, birçok bilim dalı yapay sinir ağlarıyla ilgilenmektedir (Warner ve Misra, 1996).

Yapay sinir ağılarıyla ilgili geçmişten günümüze kadar yapılan çalışmaları 1970 öncesi ve sonrası diye ikiye ayırmak gerekmektedir. Çünkü 1970 yılı tarihin bir dönüm noktası olup ve o zamana kadar olmaz diye düşünülen çoğu sorun çözülmüş ve yeni gelişmeler yaşanmıştır. Şimdiye kadar yapılan çalışmaların bazıları kronolojik olarak aşağıdaki gibi sıralanabilir (Öztemel, 2003: 37-39; Elmas, 2003: 27-29).

- 1890-İnsan beyni hakkında bilgi veren “Psychology (Briefer Course)” adlı eserin yazılması,
- 1911-İnsan beynini oluşturan bileşenlerinin sinir hücrelerinden oluştuğu fikrin benimsenmesi,
- 1943-Yapay sinir hücrelerine dayalı hesaplama teorisinin ortaya atılması ve sinir aktiviteleri hakkında eser yayınlanması,
- 1948-Sinirlerin çalışmasıyla, insan ve makineler arasındaki davranış özelliklerine değinilmesi,
- 1949-Öğrenme işlemlerinin bilgisayar tarafından gerçekleştirecek biçimde geliştirilmesi,
- 1956-1962-Widrow öğrenme algoritmasının geliştirilmesi,
- 1957-1962-Tek katmanlı algılayıcının geliştirilmesi,
- 1959-Bernand ve Hoff tarafından ADALINE ve MADALINE modellerin geliştirilmesi,
- 1967-1969-Öğrenme algoritmalarının geliştirilmesi (Grosberg öğrenme algoritması gibi),
- 1969-Tek katmanlı algılayıcıların problemleri çözme yeteneklerinin olmadığına gösterilmesi ve bundan dolayı iki katmanlı ileri beslemeli ağların ileri sürülmesi,
- 1969-DARPA’nın yapay sinir ağı çalışmalarına verilen destekleri durdurup diğer çalışmalarına destek vermesi,
- 1974-Geriye yayılım modelleriyle ilgili ilk çalışmalarının yapılması,
- 1976-1985-Danışmasız öğrenme modellerin geliştirilmesi (ART ve Kohonen modelin geliştirilmesi),

- 1980’lerde Fukushima, NEOCOGNITRON (şekil ve örüntü tanıma amaçlı) modelinin tanıtılması (Fukushima, 1980: 193-202; Fukushima, 1982: 455-469),
- 1982-Hopfield ağların geliştirilmesi,
- 1986-Rumelhart ve McClelland çok katmanlı ağlar için geriye yayımlı öğrenme modelini ortaya koyması,
- 1987-Elektrik Elektronik Mühendisliği Enstitüsü tarafından sinir ağlarını konu alan ilk uluslararası konferans yapılması,
- 1988-Broomhead ve Lowe “Radial Basis Functions” modelini çok katmanlı algılayıcılara alternatif olarak geliştirdiler (Broomhead ve Lowe, 1988). Daha sonra Specht bu ağların daha gelişmiş olan Probabilistik Ağları (PNN) geliştirdi (Specht, 1988),
- 1991-Specht tarafından Genel Regrasyon Ağların (GRNN) geliştirilmesi (Specht, 1991),

1.1.3. Yapay Sinir Ağının Tanımı

Yapay sinir ağlarının genel kabul görmüş bir tanımı bulunmamakla beraber literatürde sıkça kullanılan çok sayıda tanımı vardır.

Tanım 1:

Zurada (1992), “Yapay sinir sistemleri veya sinir ağları deneysel bilgiyi kullanan, depolayan fiziksel hücre sistemlerinden meydana gelir.” olarak tanımlamaktadır.

Tanım 2:

Öztemel (2003), “Yapay sinir ağları, insanlar tarafından gerçekleştirilen örnekleri kullanarak olayları öğrenebilen, çevreden gelen olaylara karşı nasıl tepkiler üreteceğini belirleyebilen bilgisayar sistemlerdir.” olarak tanımlar.

Tanım 3:

Haykin (1999) ise, “Bir sinir ağı, basit işlem birimlerinden oluşan, deneyimsel bilgileri biriktirmeye yönelik doğal bir eğilimi olan ve bunların kullanılmasını sağlayan yoğun bir şekilde paralel dağıtılmış bir işlemcidir.” olarak tanımlar.

Bu işlemcinin insan beyniyle ilişkisi iki şekilde açıklanabilir;

- a) İşlemci tarafından öğrenilen bilgiler, belirli bir öğrenme işlemi sayesinde elde edilir.
- b) İşlemciyi oluşturan yapay nöronlar arasındaki bağlantı ağırlıkları, elde edilen bilginin saklanması için kullanılır.

Tanım 4:

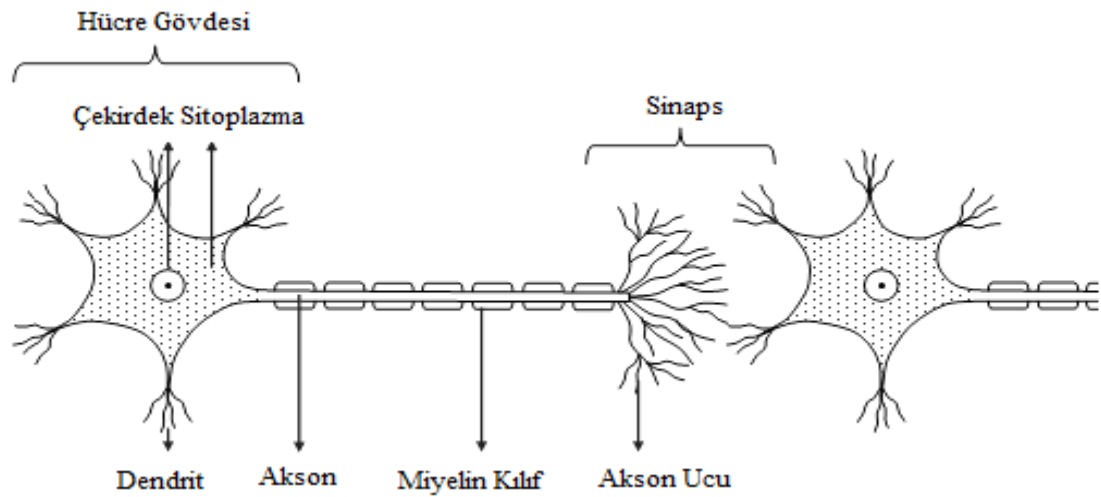
Yurtoğlu (2005), “YSA, ilgili bağlantı ağırlıklarıyla (synaptic wheights) bir ağa bağlanmış basit işlem elemanlarından (nöron) oluşan bir sistemdir.” diye tanımlamaktadır.

1.1.4. Biyolojik Bir Sinir Hücresinin Yapısı

Sinir ağı kelimesi basitleştirilmiş modelleri benimseyen insan beyninin işleyişine benzetilen biyolojik bir sinir ağından geliyor. Tipik bir nöronun içinde, çekirdek ile bir başka nöronlarla bağlantıları sağlayan bir lif ağı yoluyla yapılmış dentritler bulunur. Bir hücrenin birden çok dentriti olmasına rağmen tek bir aksonu vardır. Akson çekirdekten dışarı uzanarak bağlı olduğu diğer hücrelere karmaşık kimyasal süreç ve nöronların elektrik potansiyellerini iletir. Sinapslar ise sinir hücrelerinin kesişim noktasıdır. Bu hücrenin görevi elektriksel iletinin diğer hücrelere iletilmesini sağlar (Alavala, 2008: 121-128).

Biyolojik sinir ağları beynimizde bulunan birçok sayıda sinir hücresinin bir parçasıdır. İnsan beyni birçok sinir hücresinden oluşup, bu hücreler birbiriyle ilişkilidir. Yapay sinir ağların çalışma sistemi biyolojik sinir ağlarının

karakteristik özelliklerine benzemektedir (Gan, 2012: 340-349). İnsan beyninin en temel parçası, hatırlama, düşünme, her harekette daha önceki deneyimlere başvurma yeteneğini sağlayan kendine özgü sinir hücreleridir. Bir sinir ağı milyarlarca sinir hücresinin veya bağlantının bir araya gelmesiyle meydana gelmektedir (Öztemel, 2003: 45). Biyolojik sinir sistemi yaklaşık 10^{12} adet sinir hücresi ve bu sinir hücrelerinin 6×10^{13} kadar bağlantılarının oluşturduğu; bilgiyi veya girdiyi alan, yorumlayan, değerlendiren ve uygun bir karar üreten bir sistemdir (Seçme, 2006: 50-51).



Şekil 1.2. Biyolojik Sinir Hücresi ve Bileşenleri

Kaynak: Bilen, 2011: 9

Bir yapay nöron temel olarak girişler, ağırlıklar, toplama fonksiyonu, aktivasyon (etkinlik) fonksiyonu ve çıkış olmak üzere beş kısımdan oluşmaktadır. YSA'lar, insan beyninin çalışma prensibinden esinlenerek geliştirilmeye çalışılmış olup aralarında yapısal olarak bazı benzerlikler vardır. Bu benzerlikler Tablo 1'de verilmiştir.

Tablo 1.1. Biyolojik Sinir Ağı ile Yapay Sinir Ağı'nın Karşılaştırılması

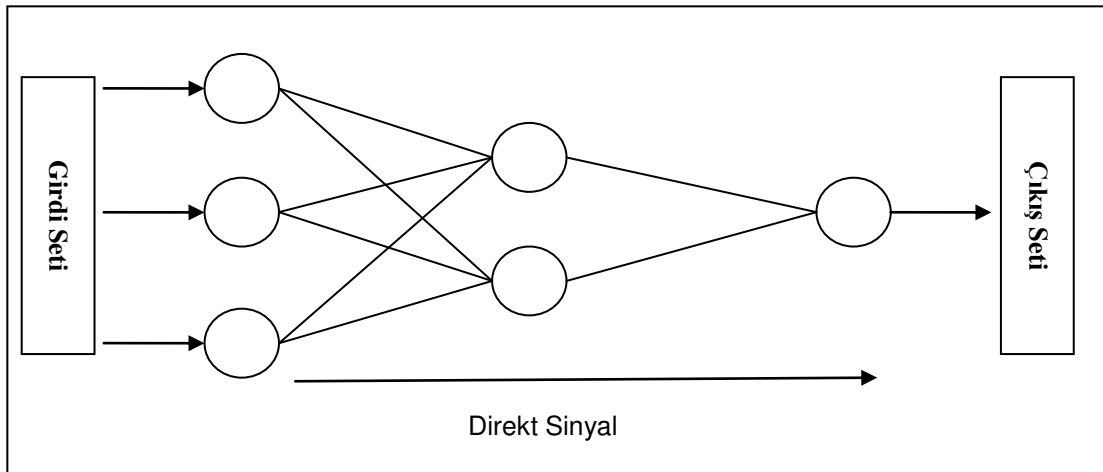
Biyolojik Sinir Ağı	Yapay Sinir Ağı
Sinir Sistemi	İşlem elemanı
Sinir	Toplama fonksiyonu
Sinapslar	Sinirler arası bağlantı ağırlıkları
Dendrit	Toplama fonksiyonu
Aksonlar	Yapay sinir çıkışı
Hücre Gövdesi	Transfer fonksiyonu (Etkinlik İşlevi)

Kaynak: Bose ve Liang, 1996: 3-31

1.1.5. Yapay Sinir Ağlarının Genel Özellikleri

Yapay sinir ağları hesaplama ve bilgi işleme gücünü, paralel dağılmış yapısından, öğrenbilme ve genelleme yeteneğinden aldığı söylenebilir. Genelleme, eğitim ya da öğrenme sürecinde karşılaşılmayan veri seti veya girdiler için yapay sinir ağlarının uygun tepkileri (çıkışları) üretmesine olanak tanır. Bu üstün yetenekleriyle YSA'nın karmaşık problemleri çözebilme gücünü gösterir. Sinir ağların paralel çalışması, seri çalışmaya göre hesaplama hızının artmasına olanak sağlar (Çelebi, 2010: 52).

Yapay sinir ağlarının karakteristik özellikleri uygulanan ağ modeline göre değişmektedir. Bu özellikler, yapay sinir ağlarının problemleri çözebilme yeteneğini gösterir. Yapay sinir ağları, aşağıdaki özellikleri veya üstünlükleri nedeniyle birçok alanda uygulanmaktadır (Öztemel, 2003).



Şekil 1.3. Sinir Ağlarında Paralellik

Bütün modeller için geçerli olan genel karakteristik özellikleri aşağıda verilmiştir.

- Bilginin saklanması,
- Sadece nümerik bilgiyle çalışması,
- Analiz ve tasarım kolaylığı,
- Paralel işlem yapma özelliği,
- Yapay sinir ağları örnekleri kullanarak öğrenmesi,
- Sisteme uygun çözümler sağlamak için ağ tekrar ve tekrar eğitilebilir,
- Genelleme yapabilme ve uyarlanabilme,
- Matematiksel modele ihtiyaç duymazlar,
- Öğrenme kabiliyeti vardır ve farklı öğrenme algoritmalarıyla öğrenirler,
- Şekil (örüntü) ilişkilendirme ve sınıflandırma yapabilirler,
- Hızlı hesaplama yapabilmesi,
- Eksik bilgiyle çalışabilirler,
- Görülmemiş örnekler hakkında bilgi üretebilme ve yeni durumlara adapte olması,
- Hata toleransına sahip olması,
- Sinyal işleme,

1.1.6. Yapay Sinir Ağlarının Avantajları

Yapay sinir ağları kullanılarak yapılan analizlerde, elde edilebilecek birçok avantajdan bazıları aşağıda verilmiştir. Yapay sinir ağları diğer sistemlerle karşılaştırıldığında birtakım üstün özellikleri bulunmaktadır (Symeonidis, 2000: 17-18).

- Şekil tanıma, ilişkilendirme, sınıflandırma ve genelleme için güçlü bir tekniktir.
- Yapay sinir ağları, geleneksel hesaplama yöntemlerine göre problemi çözerken zamanı daha verimli kullanırlar.
- Yapay sinir ağları değişen bir ortamda esnekliğe sahiptir.
- Yapay sinir ağları, programlama yerine öğrenme yoluyla kendini geliştirir.
- Yapay sinir ağların performansı klasik (geleneksel) istatistiksel yöntemler kadar iyidir. Sinir ağları daha az zamanda veri yapısını açık şekilde yansıtan modeller oluşturur.
- Yapay sinir ağları uyumlu bilgisayar donanımıyla daha iyi çalışabilmektedir. Bir yapay sinir ağı doğrusal veya doğrusal olmayabilir.
- YSA'lar doğrusal olmayan verilerle ve matematiksel olarak modellenmesi mümkün olmayan veya zor olan problem için rahat uygulanabilir.
- Yapay sinir ağları uygulamalarının maliyetinin daha ucuz olması seçilme sebebi olabilir.

1.1.7. Yapay Sinir Ağlarının Dezavantajları

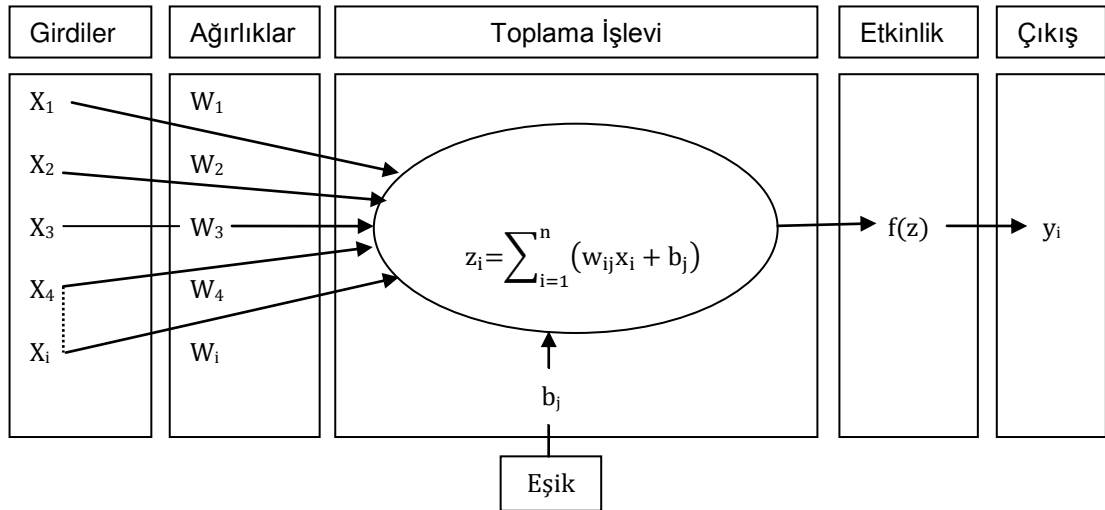
Yapay sinir ağlarının yukarıda belirtilen birçok avantajına karşılık bazı dezavantajları da vardır (Elmas, 2003; Öztemel, 2003). Bunları aşağıdaki gibi ifade edilebilir:

- YSA, probleme uygun ağ yapısını belirlemesi ancak deneme yanılma yoluyla gerçekleştirmesi,
- Uygun çözüme ulaşamama,
- Sinir ağlarının donanım bağımlı olarak çalışmaları önemli bir sorun teşkil etmesi,
- Ağ eğitime ne zaman son verileceğine karar vermek için bir yöntem olmaması,
- Farklı sistemlere uyarlanması zor olması,
- Bazı ağlarda, ağın parametre değerlerinin öğrenme katsayısı, katman sayısı vb. belirlenmesinde bir kural olmaması diğer bir problemidir.
- Ağın davranışlarının açıklanamaması,
- İstatistiksel çözümlerlerin beraberinde sorun alanına ilişkin anlaşılabilir ve yorumlamaya olanak veren parametreler üretmesine rağmen, bu parametrelerin henüz yorumlama imkanı bulunmamasıdır.

1.2. YAPAY SINIR HÜCRESİ

Yapay sinir ağları ve sinir hücreleri arasında hem mimari hem de yetenek açısından farklılıklar vardır (Zurada, 1992). YSA'nın çalışmasına esas teşkil eden en küçük birimler yapay sinir hücresi ya da işlem elemanı olarak isimlendirilir. En basit yapay sinir hücresi Şekil 1.4'de görüleceği üzere girdiler, ağırlıklar, toplama (birleştirme) fonksiyonu, aktivasyon (etkinlik) fonksiyonu ve çıkış olmak üzere 5 ana bileşenden oluşmaktadır (Öztemel, 2003). Bunlar;

1. Girdiler
2. Ağırlıklar
3. Toplama fonksiyonu
4. Aktivasyon fonksiyonu
5. Çıkış



Şekil 1.4. Yapay Sinir Hücresi

Girdiler x_i sembolüyle gösterilmiştir. Bu girdilerin her biri ağırlık (w_{ij}) ile çarpılır ve eşik değeri (b_j) ile toplanır (Eşitlik 1'de gösterilmiştir). Daha sonra sonucu oluşturmak üzere etkinlik işlevi veya aktivasyon fonksiyonuyla işlem yapılarak y_i çıkışı elde edilir (Haykin, 1994).

1.2.1. Girdiler

Bir yapay sinir hücresine dış dünyadan gelen bilgilerdir. Dış dünyadan veya bir önceki katmandan alınan bilgiler giriş olarak yapay sinir hücrelerine gönderilir (Özveren, 2006).

1.2.2. Ağırlıklar

Ağırlıklar bir hücreye gelen bilginin önemini, sayısal değerini ve nöron üzerindeki etkisini gösterir. Yukarıdaki şekilde ağırlık w_1 , girdi x_1 'in hücre üzerindeki etkisini göstermektedir. Ağırlık değerlerinin küçük veya büyük olması önemli veya önemsiz olduğu anlamına gelmez (Öztemel, 2003: 49). Yapay sinir hücresine gelen bilgiler nöronlar üzerinden çekirdeğe ulaşmadan önce ağ üzerindeki ağırlık değeriyle çarpılarak çekirdeğe iletilir. Bu şekilde giriş verilerinin istenen çıktı değerinin üzerine olan etkisi ayarlanabilir. Ağırlığın artı veya eksi olması, o girdinin etkisinin pozitif veya negatif olduğunu gösterir. Eksi değerler önemsiz anlamında değildir. Ağırlığın sıfır olması durumu herhangi bir etkinin olmadığını gösterir (Haykin, 1994). Ağırlıklar ($w_1, w_2, w_3, \dots, w_i$), yapay sinir ağları tarafından alınan girdilerin sinir üzerindeki etkisini belirleyen uygun katsayılardır (Elmas, 2003).

1.2.3. Toplam Fonksiyonu

Bu fonksiyon, bir hücreye gelen net girdiyi hesaplar.

$$z_i = \sum_{i=1}^n (w_{ij}x_i + b_j) \quad (1)$$

Birleştirme işlemi, kullanılan ağın yapısına göre değişiklik gösterebilir. Örneğin, ağırlık değerleri ile çarpılmış girdi değerlerinin ortalaması, toplanması, en büyüğü veya en küçüğünün alınması gibi yöntemlerden biri

kullanılabilir. Toplam fonksiyonu sonucunda elde edilen değer doğrusal ya da doğrusal olmayan türevlenebilir bir transfer fonksiyonundan geçirilerek işlem elemanının çıktısı hesaplanır. Eşitlik 2’de gerekli olan denklem gösterilmiştir (Öztemel, 2003).

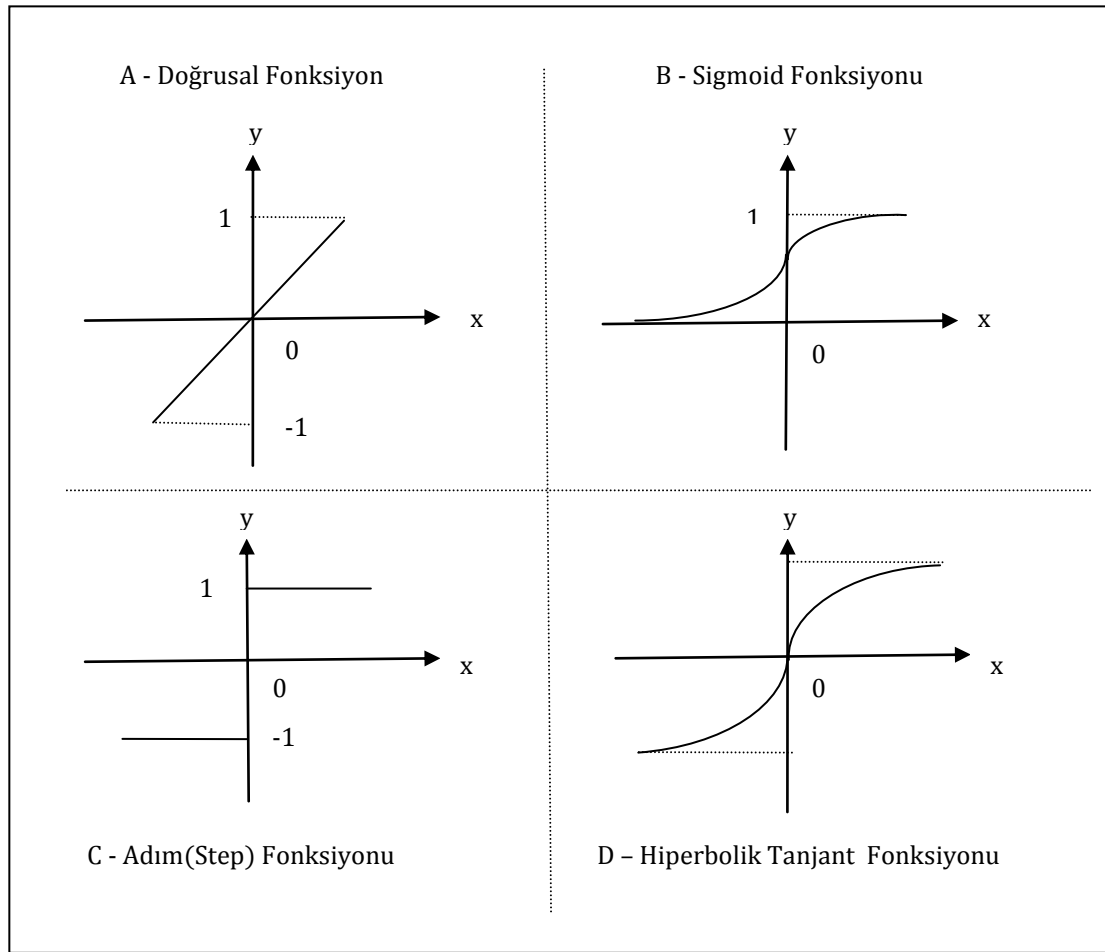
$$y = f(z_i) = f\left(\sum_{i=1}^n (w_{ij}x_i + b_j)\right) \quad (2)$$

1.2.4. Aktivasyon (Transfer) Fonksiyonu

Aktivasyon fonksiyonu ağı giren ve çıkan birimler arasındaki eğrisel eşleşmeyi sağlar (Hwang ve Ding, 1997). Aktivasyon fonksiyonunun doğru seçilmesi, ağın performansını önemli derecede etkiler. Seçilen aktivasyon fonksiyonu doğrusal değilse, eğim parametresinin belirlenmesi gerekmektedir. Eğim parametresi, uygun sonuca yeterli derecede ulaşmasında önemli rol oynayan bir faktördür (Eğrioğlu ve Aladağ, 2007).

Genellikle hatayı minimum yapan, günümüzde en çok olarak kullanılan ÇKA modelinde çoğunlukla aktivasyon fonksiyonu olarak sigmoid kullanılmaktadır (Öztemel, 2003: 50).

Kullanılan transfer fonksiyonuna göre çıktı değeri genellikle $[-1,1]$ veya $[0,1]$ arasındadır. Genellikle doğrusal olmayan bir fonksiyondur. Doğrusal olmayan transfer fonksiyonlarının kullanılması yapay sinir ağlarının karmaşık ve çok farklı problemlere uygulanmasını sağlamıştır. Çoğunlukla kullanılan dört tane aktivasyon fonksiyonu vardır. Şekil 1.5’de bu fonksiyonlar gösterilmiştir. Bunlar arasında en çok kullanılan aktivasyon fonksiyonları sigmoid Eşitlik 3’de ve hiperbolik tanjant ise Eşitlik 4 ile ifade edilmiştir.



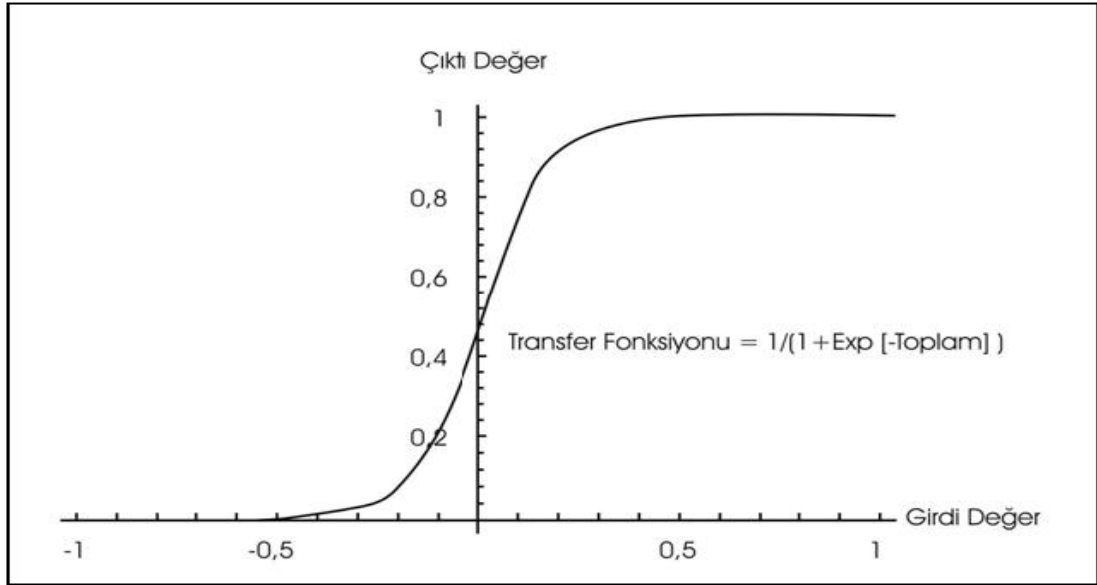
Şekil 1.5. Yapay Sinir Ağları için Kullanılan Aktivasyon Fonksiyonları

Kaynak: Yalçın, 2012: 15

$$f(z_i) = \frac{1}{1 + e^{-z_i}} \quad (3)$$

$$f(z_i) = \frac{e^{z_i} - e^{-z_i}}{e^{z_i} + e^{-z_i}} \quad (4)$$

Transfer fonksiyonunun işleyişi aşağıdaki Şekil 1.6'da sigmoid fonksiyon kullanılarak gösterilmektedir. Sigmoid transfer fonksiyonu, toplam fonksiyonundan gelen değerleri alır ve sıfır (0) ile bir (1) arasında bir değere dönüştürür. Sıfır (0) ile bir (1) arasındaki bu değer, işlem elemanın çıktısıdır. Elde edilen bu çıktı başka bir ağın girdisi olabilir.



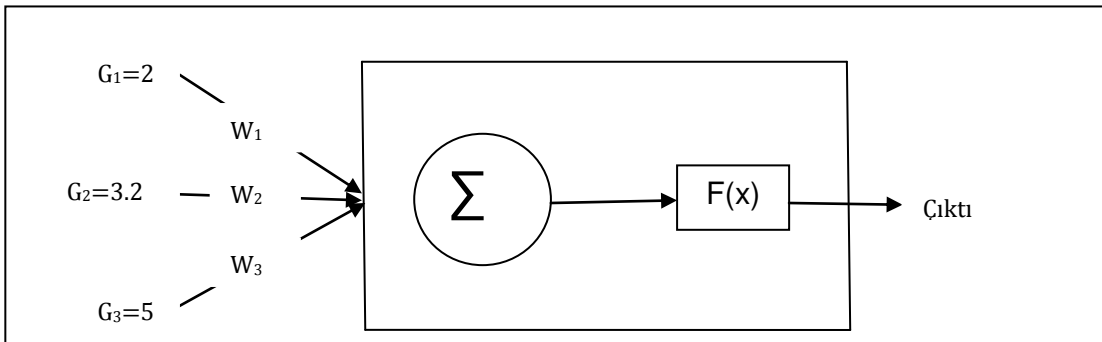
Şekil 1.6. Transfer Fonksiyonunun Çalışma Yapısı

Kaynak: Yurtoğlu, 2005: 19

1.2.5. Çıkış İşlevi

Her bir sinir ağının tek bir çıkışı vardır. Aktivasyon fonksiyonu uygulanmasıyla elde edilen çıktı değeridir. Üretilen çıktı başka bir hücrenin girdisi olabilir.

1.2.5.1. Yapay Sinir Hücresinin Çalışmasına Bir Örnek



Şekil 1.7. Basit Yapay Sinir Hücresi Örneği

Örneğimizde girdi ve ağırlık değerleri aşağıdaki Tablo 1.2’de verilmiştir.

Tablo 1.2. Yapay Sinir Hücresinin Girdi ve Ağırlık Değerleri

G_1	G_2	G_3	W_1	W_2	W_3
2	3.2	5	0.1	0.2	0.5

Aktivasyon fonksiyonuna olarak ise sigmoid seçilmiş olsun:

$$\text{Toplam} = \sum_{j=1}^3 G_j * W_j$$

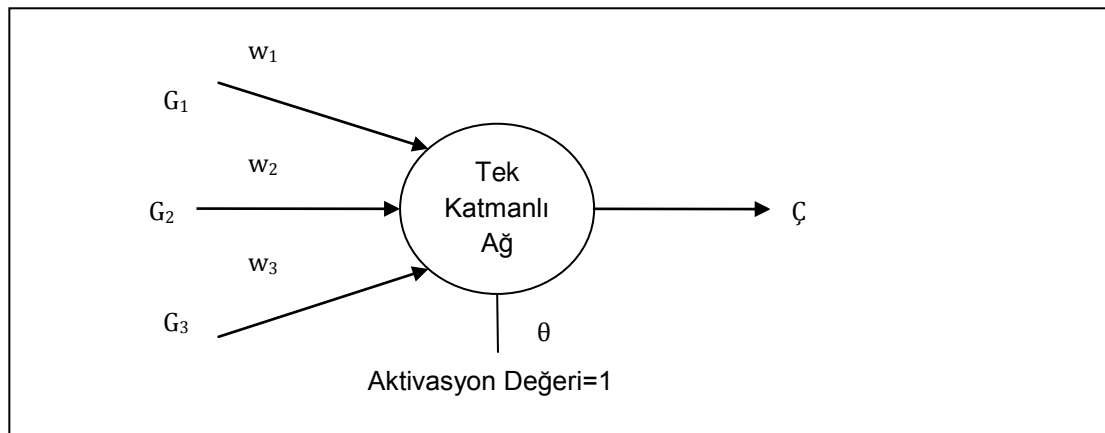
$$\text{Toplam} = 2 * 0.1 + 3.2 * 0.2 + 5 * 0.5$$

$$\text{Toplam} = 3.34$$

$$\text{Çıktı} = f(3.34) = \frac{1}{1 + e^{-\text{Toplam}}} = \frac{1}{1 + e^{-3.34}} = 0.965$$

1.3. BİR YAPAY SİNİR AĞI'NIN YAPISI

Bir yapay sinir ağı sadece girdi ve çıktı katmanlarından oluşuyorsa tek katmanlı ağıdır. Sadece giriş ve çıkış katmanı olan ağlar, karmaşık işlemleri hesaplama yeteneğinden yoksundurlar. Karmaşık işlevleri hesaplamak için en az bir ara katman olmalıdır (Elmas, 2003: 43-66). Aşağıdaki Şekil 1.8'de üç girdisi ve bir çıktıdan oluşan örnek verilmiştir.

**Şekil 1.8. Basit Tek Katmanlı Algılayıcı Modeli**

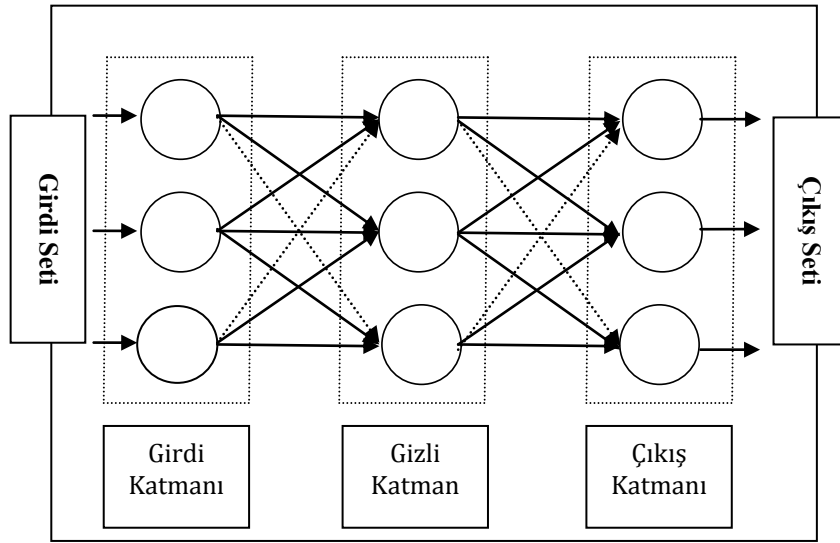
İşlem elamanlarını değerini ve ağın çıktısının sıfır (0) olmasını engelleyen bir aktivasyon (θ) vardır. Aktivasyon değerinin girdisi daima bir (1)' dir.

$$\zeta = f\left(\sum_{i=1}^n (W_i G_i + \theta)\right) \quad (5)$$

Tek katmanlı ağlarda çıktı fonksiyonu doğrusaldır. Ağa gösterilen örnekler iki sınıf arasında paylaştırılarak, doğru sınıfı bulmaya çalışır. Ağın çıktısı 1 veya -1 değerini alır (Literatürde 1 ve 0'da kullanılmaktadır.). Eğer ağın çıktı değeri 1 ise birinci sınıfta yok değilse ikinci sınıfta yer almaktadır (Öztemel, 2003: 59-74).

$$f(G) = \begin{cases} 1 & \text{Eğer } \zeta > 0 \text{ ise} \\ -1 & \text{Aksi takdirde,} \end{cases} \quad (6)$$

Çok katmanlı ağlarda ise, yapısında birden fazla gizli katman bulunmaktadır. Bu ağların çalışma prensibi temel olarak iki fazdan oluşmaktadır: eğitim ve test aşamasıdır. Eğitim aşamasında seçilen öğrenme algoritmasına göre ağırlıklar hesaplanır ve bu değerlere göre bir çıktı hesaplanır. Test aşamasında ise ağın görmediği örneklerden faydalanılarak sistem test edilir. Öğrenme algoritması yöntemleri, girdi setine karşılık uygun çıktı seti belirlenmesinde ve ağları hızlandırmada önemli rol oynamaktadır. Tasarlanan YSA modelinin yapısı ve bileşenleri Şekil 1.9'da gösterilmiştir (Ebrahimzadeh ve Ranaee, 2010: 387-393).



Şekil 1.9. Çok Katmanlı Bir Yapay Sinir Ağı

Gizli katmanları olmayan, sadece giriş ve çıkış katmanı olan ağlar, karmaşık olayları hesaplama yeteneğinden yoksundur. Bu nedenle karmaşık hesaplamalar için oluşturulan ağlarda en az bir ara katman olmalıdır. Bir yapay sinir ağının en temel görevi, kendisine gösterilen bir girdi setine karşılık gelebilecek bir çıktı seti belirlemektir. Bunu yapabilmesi için ağ, ilgili olayın örnekleriyle eğitilerek genelleme yapabilecek yeteneğe kavuşturulur. Bu genellemeye karşılık gelen çıktı seti belirlenir.

Bir yapay sinir ağında, birbirleriyle bağlantılı sinir hücrelerinin yer aldığı girdi katmanı, çıktı katmanı ve gizli katman olmak üzere temelde üç katman bulunmaktadır. Girdi katmanı ilk katmandır ve dışarıdan gelen bilgilerin yapay sinir ağına alınmasını sağlar. Gizli katmanda bulunan nöronların dış ortamla bağlantıları yoktur. Yalnızca girdi katmanından gelen sinyalleri alırlar ve çıktı katmanına sinyal gönderirler. Son katman çıktı katmanı olarak adlandırılır ve bilgilerin dışarıya iletilmesi işlevini görür. Modeldeki diğer katmanlar ise girdi katmanı ile çıktı katmanı arasında yer alır ve gizli katman olarak adlandırılır.

Temelde öğrenme yöntemleri danışmanlı, danışmansız, karma ve takviyeli olmak üzere dört gruba ayrılmıştır. Danışmanlı öğrenmede dışarıdan

bir eğitcinin yapay sinir ağına müdahalesi söz konusudur. Eğitim işlemi, sinir ağına giriş ve çıkış bilgileri sunmaktan oluşur. Danışmansız öğrenmede sistemin doğru çıkış hakkında bilgisi yoktur. Danışmansız olarak eğitilebilen ağlar, beklenen ya da hedef çıkış olmadan giriş verilerinin özelliklerine göre ağırlık değerlerini ayarlar. Ağın eğitimi tamamlandıktan sonra öğrenip öğrenmediğini anlamak için yapılan denemelere, ağın “test edilmesi” denilmektedir.

Ağı test etmek için ağın öğrenme sırasında görmediği örneklerden yararlanılır. Ağ, eğitim sırasında belirlenen bağlantı ağırlıklarını kullanarak görmediği bu örnekler için çıktılar üretir. Elde edilen çıktılar ağın doğruluk değerleri hakkında bilgiler verir (Çuhadar, Güngör ve Göksu, 2009: 99-114). YSA, hücrelerin bağlanma biçimlerine göre “ileri beslemeli” ve “geri beslemeli” olmak üzere iki çatı altında sınıflandırılabilir.

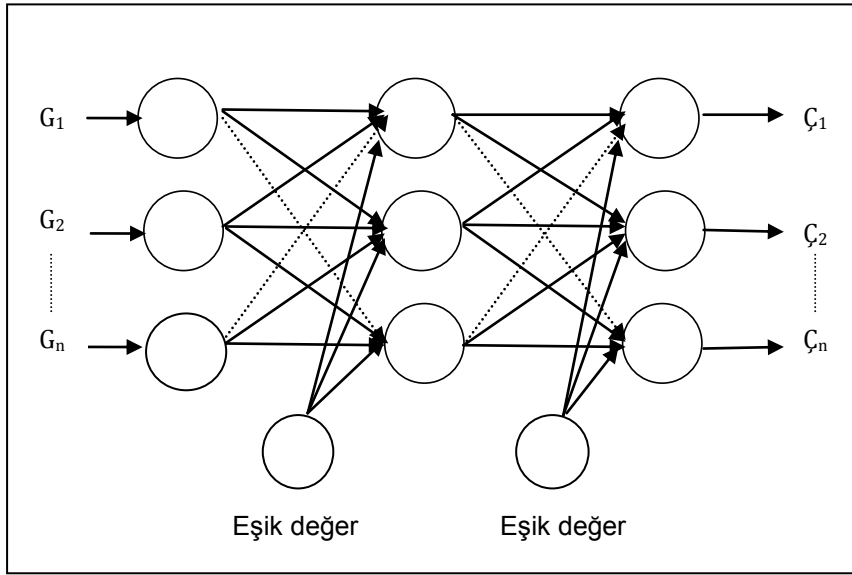
İKİNCİ BÖLÜM

ÇOK KATMANLI ALGILAYICILAR (ÇKA)

2.1. ÇOK KATMANLI ALGILAYICI VE YAPISI

Çok Katmanlı Algılayıcı (ÇKA) modeli yapay sinir ağlarına olan ilgiyi çok hızlı bir şekilde arttırmış ve YSA tarihinde yeni bir dönem başlatmıştır. Sinir ağlarının ilk modellerinin temel özellikleri doğrusal olayları çözebilme yeteneğine sahip olmalarıdır. Bu ağlar ile doğrusal olmayan modeller çözülememektedir. Bu sorunu çözmek için ÇKA geliştirilmiştir. Bu ağ modeli özellikle mühendislik uygulamalarında en çok kullanılan sinir ağı modeli olmuştur. Özellikle sınıflandırma, tanıma ve genelleme yapmayı gerektiren problemler için çok önemli bir çözüm aracıdır. Bu model Delta Öğrenme kuralı denilen bir öğrenme yöntemini kullanmaktadır. Modelin amacı ağın beklenen çıktısıyla ürettiği çıktı arasındaki hatayı en aza indirmektir (Öztemel, 2003: 75-76). Rumelhart ve arkadaşları tarafından geliştirilen bu modele hata yayma modeli veya geriye yayılım modeli denilmektedir (Rumelhart, Hinton ve Williams, 1986: 533-537).

ÇKA'ların yapısı Şekil 2.1'de gösterilmiş ve şekildeki gibi katmanlar ileriye doğru bağlantılı ve 3 katmandan oluşan bir ağ yapısıdır.



Şekil 2.1. Çok Katmanlı Algılayıcı Modeli

ÇKA dış çevreden gelen girdileri ($G_1, G_2, \dots, G_n$) olarak ara katmana gönderir. Bu katmanda her hangi bir bilgi işleme olmaz. Daha sonra ara katmanlar girdi katmanından gelen bilgileri aldıktan sonra işleyerek bir sonraki katmana gönderir. Bir ÇKA'nın yapısında birden fazla ara katman ve her katmanda birden fazla işlem elamanı olabilir. En son girdi katmanından gelen girdilere karşılık ağıın ürettiği çıktıları ($\zeta_1, \zeta_2, \dots, \zeta_n$) belirleyerek dış çevreye gönderir. ÇKA'lar öğretmenli öğrenme stratejisini kullanır. (Rumelhart, Hinton ve Williams, 1986: 533-537).

2.2. ÇOK KATMANLI ALGILAYICILARDA ÖĞRENME KURALI

Geriye yayılım algoritması iki safhadan oluşmaktadır. İleriye doğru hesaplama ve geriye doğru hesaplama oluşmaktadır. İleriye doğru hesaplama ağıın çıktısını hesaplarken, geriye doğru hesaplamada ağıın ağırlıkları değiştirilir. Bu iki safhada ağıın gerçekleştirdiği hesap adımları aşağıda özetlenmiştir (Öztemel, 2003).

2.2.1. İleriye Doğru Hesaplama

Eğitim setindeki bir örnek girdi katmanından ağa gönderilmesiyle başlar. Gelen girdiler herhangi bir değişiklik olmadan (her hangi bir bilgi işleme olmadan) ara katmana gönderilir. Bu durumda girdi katmanındaki k. işlem elemanının çıktısı ζ_k^i , Eşitlik 7’de ifade edildiği gibidir.

$$\zeta_k^i = G_i \quad (7)$$

Ara katmanlar girdi katmanından gelen bilgileri işleyerek (her bir bağlantı ağırlığını dikkate alarak) bir sonraki katmana gönderir. Ara katman elemanına gelen net girdi Eşitlik 8’de gösterilmiştir.

$$Net_j^a = \sum_{k=1}^n A_{kj} \zeta_k^i \quad (8)$$

Burada A_{kj} , k. girdi katmanını ve j. ara katmana bağlayan bağlantı ağırlık değerlerini ifade etmektedir.

Aktivasyon fonksiyonu sigmoid seçilmesi halinde j. ara katman elemanının çıktısı Eşitlik 9’da görüldüğü gibi gerekli formül yardımıyla ifade edilmiştir.

$$\zeta_j^a = \frac{1}{1 + x^{-(Net_j^a + \beta_j^a)}} \quad (9)$$

Burada B_j , ara katmanda bulunan j. elemana bağlanan eşik değerinin ağırlığını göstermektedir.

2.2.2. Geriye Doğru Hesaplama

Ağa sunulan girdinin ürettiği çıktıyla ağın hedeflenen çıktıları ($B_1, B_2, \dots, B_n$) karşılaştırılır. Girdi ile çıktı arasındaki fark alınarak hata değerini bulmuş oluruz. Burada amaç hatanın azaltılmasıdır. Ağın bundan sonraki amacı bu hatanın düşürülmesi olduğundan, geriye doğru hesaplamada bu hata değeri ağın bağlantı ağırlıklarına dağıtılır. Çıktı katmanındaki n . işlem elemanında oluşan hata ($E_1, E_2, \dots, E_n$), Eşitlik 10'daki gibi hesaplanır.

$$E_n = B_n - C_n \quad (10)$$

Çıktı katmanı sonucu oluşan hatalar toplanarak Eşitlik 11'de gösterildiği gibi toplam hatayı (TH) bulmuş oluruz. Hata değerlerinin bazıları negatif olmasından dolayı, toplamın sıfır olmasını engellemek amacıyla hata değerlerinin karesi alınarak sonucun karekökü bulunur.

$$TH = \frac{1}{2} \sum_{n=1} E_n^2 \quad (11)$$

Bu hataya sebep olan işlem elamanlarının dağıtılması gerekmektedir. Bunun anlamı ağırlıkların değiştirilmesi gerekliliğidir. Ağın ağırlıklarını değiştirmek için iki durumdan bahsedeceğiz;

1. Çıktı katmanı ile ara katman arasındaki bağlantı ağırlıklarının değiştirilmesi
2. Ara katmanlar arası veya ara katman ile girdi katmanı arasındaki bağlantı ağırlıklarının değiştirilmesi:

2.3. ÇOK KATMANLI ALGILAYICILARIN ÇALIŞMA PROSEDÜRÜ

ÇKA'ların çalışması şu adımları içermektedir (Öztemel, 2003: 81).

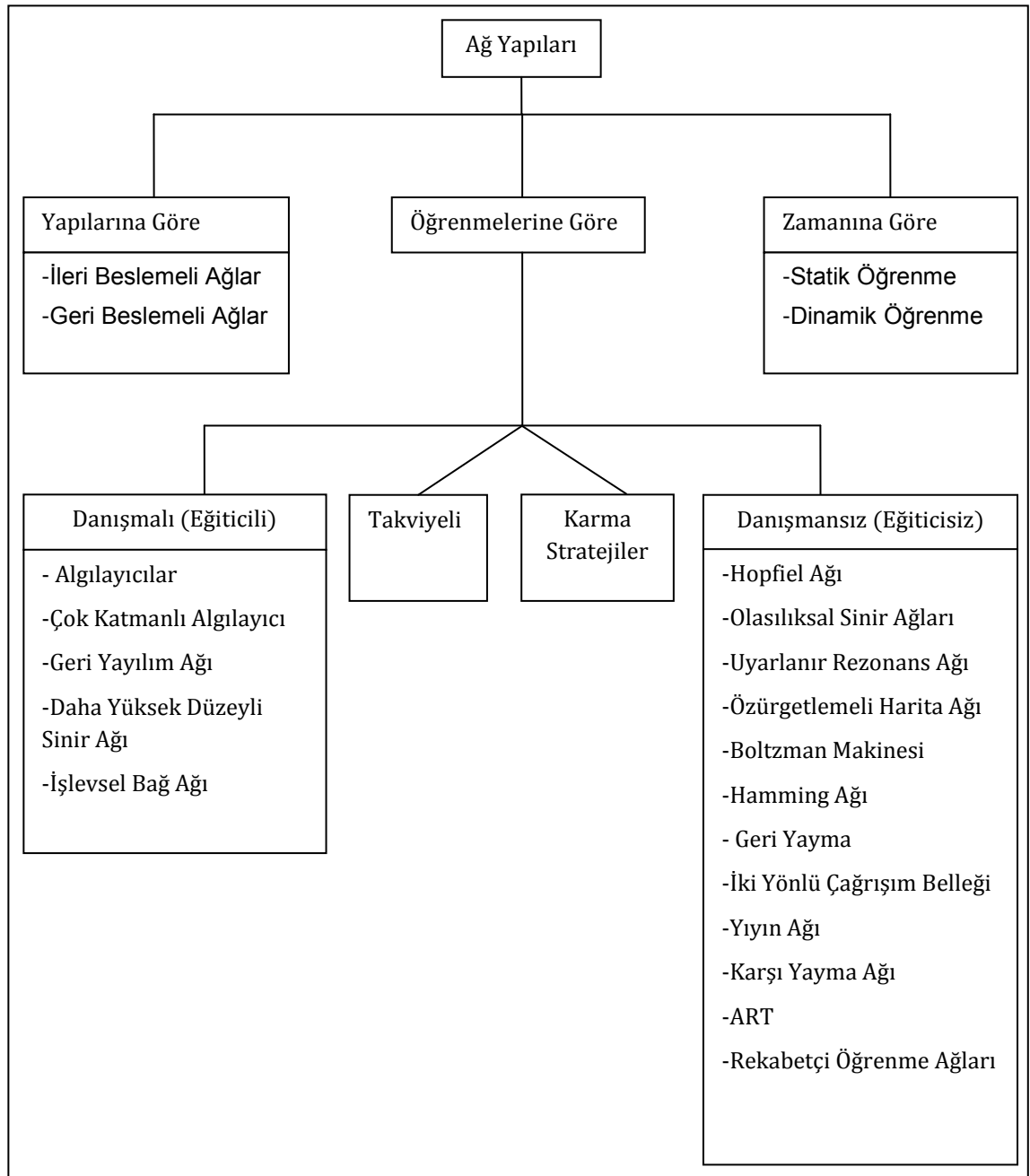
- Örneklerin toplanması,
- Ağın topolojik yapısının belirlenmesi (Gizli katman vb.),
- Öğrenme parametrelerin belirlenmesi,
- Ağırlıkların başlangıç değerlerinin atanması,
- Öğrenme setinden örneklerin seçilmesi ve ağa gösterilmesi,
- Öğrenme sırasında ileri hesaplamaların yapılması,
- Gerçekleşen çıktıyla beklenen çıktının karşılaştırılması,
- Ağırlıkların değiştirilmesi,

2.4.YAPAY SİNİR AĞLARININ SINIFLANDIRILMASI

Yapay sinir ağları işleyiş olarak birbirine benzemesine rağmen, ağın mimarisi, öğrenme yöntemi ve bağlantı yapıları gibi kriterlerden ötürü farklılık gösterebilmektedir. Genel olarak YSA'lar üç ana kritere göre sınıflandırılmaktadır (Temizel, 2012: 25).

Yapay sinir ağları genel olarak şu başlıklar altında sınıflandırılmaktadır;

1. Yapılarına göre,
2. Öğrenme algoritmalarına göre,
3. Öğrenme zamanlarına göre,



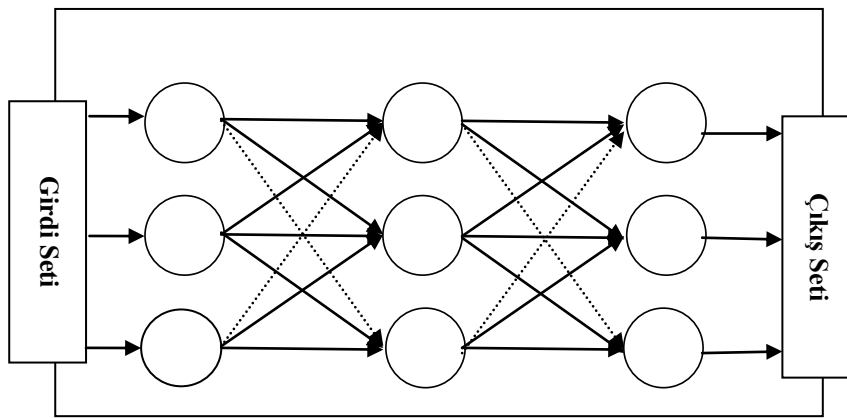
Şekil 2.2. Yapay Sinir Ağlarının Ağ Yapıları

Kaynak: Öztemel, 2003; Elmas, 2003; Temizel, 2012

2.4.1. Yapılarına Göre Yapay Sinir Ağları

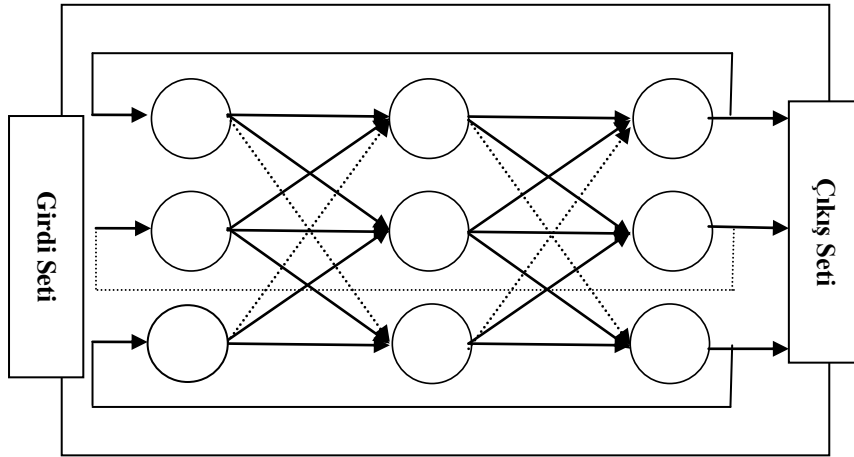
YSA sahip olduğu hücrelerin bağlanma biçimlerine göre “ileri beslemeli” ve “geri beslemeli” olmak üzere iki çatı altında sınıflandırılabilir (Slaughter, 2003).

İleri Beslemeli Ağlar: İleri beslemeli yapay sinir ağlarında tersine bir yönelme yoktur. Gelen verilerin ilk önce giriş katmanına daha sonra ara katmana ve çıktı katmanlarına doğru (ileri) aktığı ağ yapısıdır. Bir katmandaki yapay sinir hücreleri bir önceki katmanın yapay sinir hücreleri tarafından beslenir (Özveren, 2006). ÇKA ve LVQ (Öğrenmeli Vektör Kuantalama) ileri beslemeli ağlara örnek olarak verilebilir.



Şekil 2.3. İleri Beslemeli Nöron Ağı

Geri Beslemeli Ağlar: Geri beslemeli sinir ağlarında, en az bir hücre sonraki katmanlardaki hücrelerce de beslenir. Veri akışının sadece ileriye doğru değil geriye doğru da olabileceği ağ yapısıdır (Asilkan ve Irmak, 2009: 375-391). Bu çeşit sinir ağlarının dinamik hafızaları vardır. Geri beslemeli ağlar, tahmin uygulamaları için uygundur. Geri beslemeli ağlar, çeşitli tipteki zaman serilerinin tahmininde oldukça başarı sağlamışlardır (Ataş, 2011: 29).



Şekil 2.4. Geri Beslemeli Nöron Ağı

Günümüzde, belirli amaçlarla ve değişik alanlarda kullanılmaya uygun birçok yapay sinir ağı modeli (MLP, RBF, SOM, ART vb.) geliştirilmiştir.

2.4.2. Öğrenme Algoritmalarına Göre Yapay Sinir Ağları

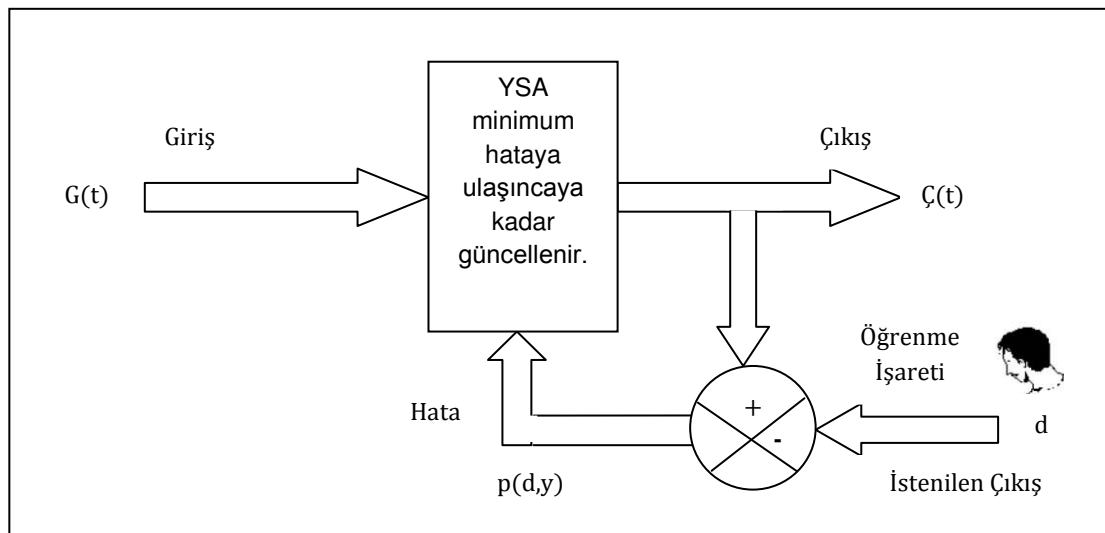
Yapay sinir ağların giriş katmanına gönderilen verilerin istenen çıktıyı üretebilmesinin yolu, ağın öğrenebilmesinden geçer. Bu durum hücreler arasındaki ağırlıkların değiştirilmesiyle olur. Bu öğrenme işleminin ise birden fazla yöntemi vardır. Yapay sinir ağları öğrenme algoritmalarına göre danışmalı, danışmansız, takviyeli ve karma öğrenme olarak 4 (dört)'e ayrılır.

2.4.2.1. Danışmalı (Eğitici) Öğrenme

Danışmalı öğrenmede, dışarıdan bir eğitcinin yapay sinir ağına müdahalesi söz konusudur. Eğitim sırasında, hem girdi değerleri hem de o girdi değerleri için istenen çıktı değerleri sisteme verilir. Gerçekleşen çıktıyla hedeflenen (beklenen) çıktılar arasındaki hata hesaplandıktan sonra, ağın yeni ağırlıkları bu hata payına göre düzenlenir. Hata payı hesaplanırken ağın bütün çıktı değerleriyle beklenen çıktıları arasındaki fark hesaplanır ve bu farka göre her ağı düşen hata payı bulunur. Daha sonra her ağ kendine gelen ağırlıkları günceller. Bu hata minimum olana kadar, ağ nöronlar

arasındaki ağırlıkları düzelterek iterasyona devam eder (Sağıroğlu, Beşdok ve Erler, 2003).

Ağ kendisine sunulan örneklerden genellemeler yaparak bir çözüm uzayı elde eder. Daha sonra bu çözüm uzayı benzer girdiler için sonuç üretir. Eğitim aşamasında genelde çok zaman harcanır. Widrow-Hoff tarafından geliştirilen Delta Kuralı ve Rumelhart ve McClelland tarafından geliştirilen Genelleştirilmiş Delta Kuralı veya Geri Yayılım algoritması, Çok Katmanlı Ağlar danışmalı öğrenme algoritmalarına örnek olarak verilebilir (Yalçın, 2012: 16-17).



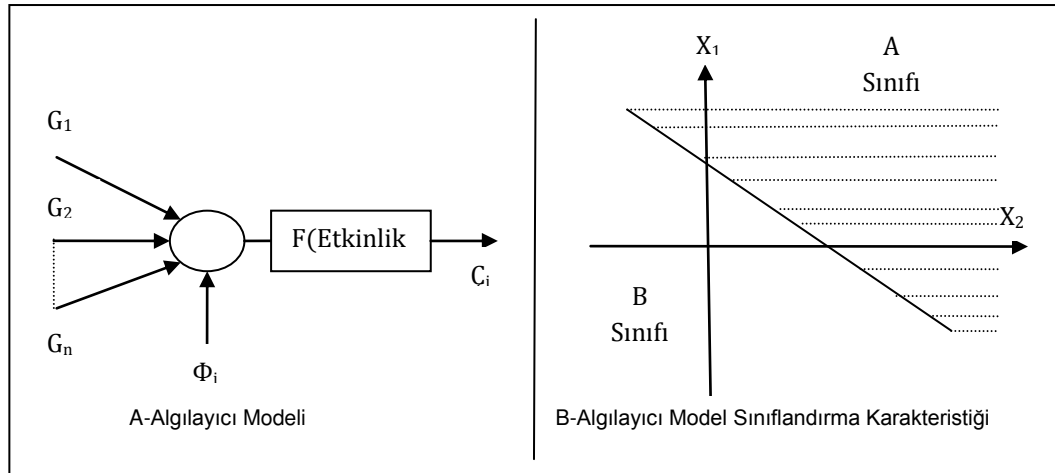
Şekil 2.5. Danışmanlı Öğrenme Yapısı

Kaynak: Yazıcı, 2010: 20

2.4.2.1.1. Perceptron (İdrak-Algılayıcı)

Algılayıcılar, ilk defa 1943 yılında Mc Culloch ve Pitts tarafından ortaya atılmıştır. Daha sonra 1958 yılında Frank Rosenblatt “Perceptron (İdrak-Algılayıcı)” tarafından sinir ağları ve şekil tanıma olarak ortaya çıkarılmıştır. Rosenblatt, Algılayıcıları bir bilgisayar olarak, makinelerin sıradan çözemediği problemlere uygulanacağını ifade ediyordu. Algılayıcıların temeli,

sinir ağının birden fazla girdiyi alarak bir çıktı üretmesine dayanır (Rosenblatt, 1958). Şekil 2.6'da tek katmanlı algılayıcıları belirten model görülmektedir.



Şekil 2.6. Algılayıcı Model Örnekleri

Kaynak: Haykin, 1999

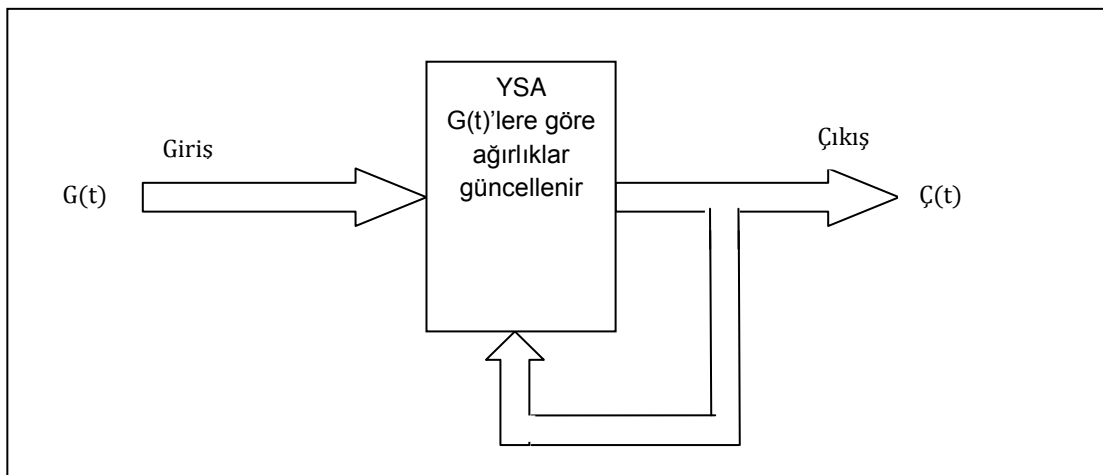
Bu basit ağ geliştirildiğinde, basit şekilleri öğrenme kabiliyetinden dolayı oldukça ilgi toplamıştır. Algılayıcılar daha çok nesneleri iki ayrı sınıfa ayırmak için kullanılmıştır. Şekil 2.6'da bir girişin iki sınıftan hangisine ait olacağı gösterilmiştir.

A veya B hangisine ait olduğuna karar veren bir algılayıcı şekli görülmektedir. Düğüm, giriş elemanlarının ağırlıklarının toplamını hesaplar ve bu aktivasyon değeri olan ϕ ile karşılaştırır, çıkışı +1 veya -1 olacak şekilde sonucu bir süreksizlikten geçirir. Karar durumu, eğer çıkış +1 ise A sınıfı ile, -1 ise B sınıfı ile ifade edilir. Bu ilk algılayıcı modeli, giriş bilgisinin mevcut iki sınıftan hangisine eşit olabileceğini bulacak şekilde gösteren basit bir ağıdır (Kaya, 2010: 47).

2.4.2.2. Danışmasız (Eğitici-siz) Öğrenme

Sistemin öğrenmesine yardımcı olan herhangi bir öğretmen veya danışman yoktur. Bu nedenle çoğu zaman bu yöntem kendi kendine

öğrenme de denilmektedir. Ağ, kendine sunulan örnek girdileri alır ve belli bir kriterle göre sınıflandırır. Bu kriterler önceden bilinmeyebilir. Ağ, kendi öğrenme kriterlerini kendisi oluşturmaktadır. Örnekteki parametreler arasındaki ilişkileri sistemin kendi kendine öğrenmesi beklenir. Danışmalı öğrenmeye göre çok daha hızlıdır ve matematiksel algoritmaları daha basittir. Adaptif Rezonans Teorisi (ART) ağıları örnek olarak verilebilir (Yalçın, 2012: 16). Grossberg tarafından geliştirilen ART veya Kohonen tarafından geliştirilen SOM öğrenme kuralı, danışmansız öğrenmeye örnek olarak verilebilir. Danışmansız öğrenme yapısı Şekil 2.7'de olduğu gibidir (Carpenter ve Grossberg, 1987).

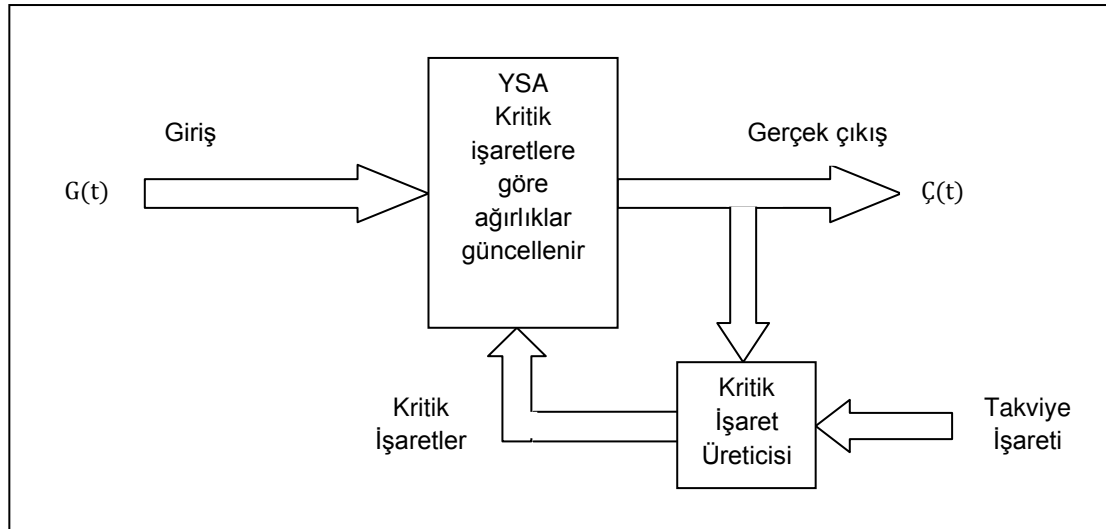


Şekil 2.7. Danışmansız Öğrenme Yapısı

2.4.2.3. Takviyeli (Destekleyicili) Öğrenme

Bu öğrenme kuralı danışmalı öğrenmeye yakın bir yöntemdir. Bu yöntemde, ağın üretmesi gereken sonuç yerine, onun ürettiği sonucun sadece doğru veya yanlış olduğunu söyleyen bir takviye işaretinin olmasıdır. Bu sistemin işleyişi ağa bir takviye işaretinin gönderilmesiyle gerçekleşir. Bu yöntem, ilgili veri seti için beklenen çıkışın oluşturulamadığı durumlarda çok faydalıdır (Hanbay, 2007: 24). Ağın her iterasyonu sonucunda bulduğu sonucun iyi ya da kötü olup olmadığına dair bir bilgi verilir. Ağ bu bilgilere göre kendini tekrar düzenler. Bu sayede ağ herhangi bir girdi dizisiyle hem

öğrenerek hem de sonuç çıkararak işlemeye devam eder (Doğan, 2010: 46-49).



Şekil 2.8. Takviyeli Öğrenme Yapısı

Kaynak: Doğan, 2010: 49

2.4.2.4. Karma Stratejiler

Yukarıdaki üç stratejiden bir kaçını birlikte kullanarak öğrenme gerçekleştiren ağlardır. Olasılıksal tabanlı ağlar (PBNN) ve radial tabanlı sinir ağları (RBNN) bunlara örnek gösterilebilir.

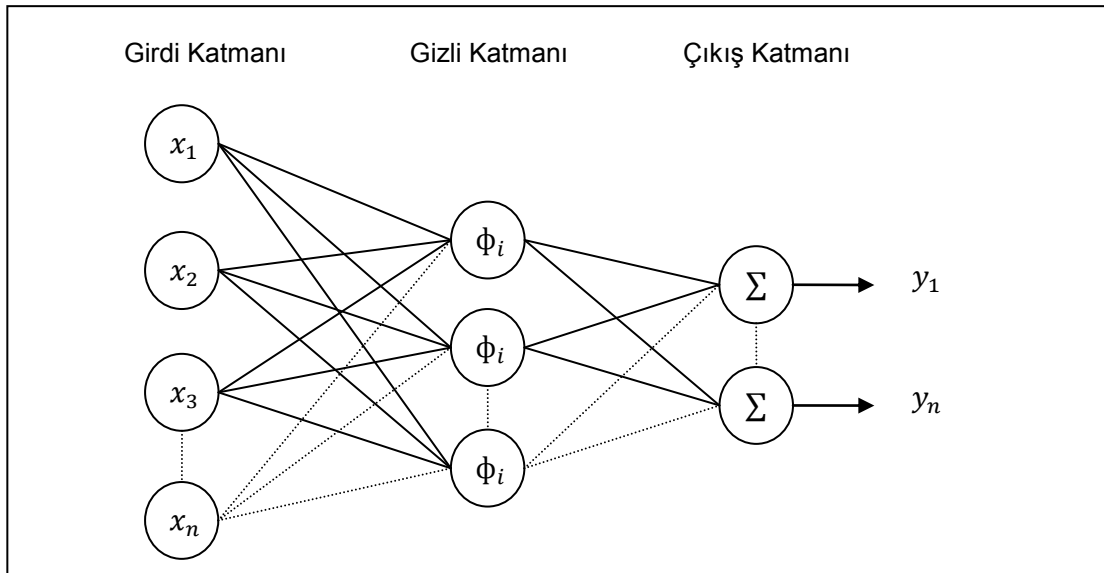
2.4.2.4.1. Radyal Tabanlı Yapay Sinir Ağları

Radyal Taban Fonksiyonlu Ağlar (RTFA) danışmalı öğrenme temeline dayanır. Çoğu araştırmacı tarafından Çok Katmanlı Algılayıcılara alternatif olarak önerilmektedir. ÇKA'ya göre daha kısa sürede öğrenebilme yeteneğine sahiptir. RTFA daha çok ileri besleme ağ yapısındadır. Dış çevreyi ağa bağlayan bir girdi katmanı, bir gizli katman ve bir çıktı katmanı olmak üzere 3 katmandan oluşmaktadır. Ağ yapısı Şekil 2.9'da görülmektedir. Bu ağ türü fonksiyon ve sınıflama gibi alanlarda yaygın olarak kullanılmaktadır. Bu ağ türünde kullanılan aktivasyon fonksiyonu genellikle

Gaussian dağılıma dayanmakta ve Eşitlik 12'deki bağıntıyla ifade edilmektedir (Kaftan, 2010: 29).

$$\phi_i(x) = \exp\left(\frac{-x^2}{\sigma^2}\right) \quad (12)$$

σ = Radyal tabanlı fonksiyonların genişliğini kontrol eden merkez yayılım parametresi



Şekil 2.9. Radyal Taban Fonksiyonlu Ağ Yapısı

Tablo 2.1. Öğrenme Algoritmaları ve Kullanım Alanları Tablosu

Öğrenme Yöntemi	Öğrenme Kuralı	Ağın Mimarisi	Öğrenme Algoritması	Kullanıldığı Yerler
Danışmanlı	Hata Düzeltme	Bir veya daha fazla katman	Geriye Yayılım Algılayıcılar	Şekil Tanıma Tahmin Kontrol
	Boltzman	Geri Dönüşlü	Boltzmann Öğrenmesi	Şekil Sınıflandırma
	Hebbian	Çok Katmanlı	Doğrusal Diskriminant	Veri Analizi Şekil Sınıflandırma
	Rakabetli	Rekabetli	Vektör Kuantalama Öğrenmesi	Veri Sıkıştırma
		ART Ağı	Art Haritalaması	Sınıflandırma
Danışmasız	Hata Düzeltme	Çok Katmanlı	Projeksiyon	Veri Analizi
	Hebbian	İleri Beslemeli	Temel Analiz	Veri Analizi Veri Sıkıştırma
		Hopfield Ağı	İlişkili Öğrenme	İlişki
	Rekabetli	Rekabetli	Vektör Kuantalaması	Veri Sıkıştırma Sınıflandırma
		Kohonen, SOM	Kohonen, SOM	Sınıflandırma Veri Analizi
		ART Ağı	ART 1, ART 2	Sınıflandırma
Karma	Hata Düzeltme	Radyal Tabanlı	Radyal Tabanlı Öğrenme	Sınıflandırma Tahmin Kontrol Şekil Tanıma

Kaynak: Yazıcı, 2010: 35

2.4.3. Öğrenme Zamanına Göre Yapay Sinir Ağları

Yapay sinir ağları öğrenme zamanına göre dinamik ve statik öğrenme olarak ikiye ayrılır.

2.4.3.1. Statik Öğrenme

Statik öğrenme kuralıyla çalışan ağların eğitim tamamlandıktan sonra sinir ağı istenilen şekilde kullanılabilir. Eğitim sırasında ağı üzerindeki ağırlıklarda herhangi bir değişiklik olmaz (Kaya, 2010: 44).

2.4.3.2. Dinamik Öğrenme

Dinamik öğrenme kuralı, sinir ağlarının öğrenmesini öngörerek tasarlanmıştır.

2.5. YAPAY SİNİR AĞLARINDA ÖĞRENME KURALLARI

Öğrenme kurallarında kullanılan çok sayıda değişik öğrenme algoritması bulunmaktadır. YSA'nın mimarisine, karşılaşılan sorunun niteliğine göre farklılık gösteren bu öğrenme algoritmalarının çok fazla çeşidi bulunmaktadır. Yapay sinir ağlarında öğrenmelerin çoğu Hebb kuralına dayanmaktadır (Öztemel, 2003). En çok kullanılan öğrenme algoritmaları ve diğerleri:

- 1) Hebb,
- 2) Delta,
- 3) Hopfield,
- 4) Kohonen,
- 5) Geri Yayılma,
- 6) Levenberg Marquardt,
- 7) LVQ (Öğrenmeli Vektör Kuantalama)
- 8) Hızlı Yayılım (Quick Propagation),
- 9) Eşleştirmeli Eğilim (Conjugate Gradient) Algoritmaları,
- 10) Kuasi-Newton (Quasi Newton) Öğrenme Algoritması,
- 11) Online Geri Yayılım (Online Back Propagation),
- 12) Grup Geri Yayılım (Batch Back Propagation) Algoritması,

2.5.1. Hebb Kuralı

1949 yılında Donald Hebb "The Organization of Behavior (Davranış Örgütü)" adlı eser yayınlayarak, sinir substratı hakkında çeşitli öğrenme ve bellek hipotezlerini tanıttı. Bu Hebb öğrenme kuralı veya Hebb sinapsı gibidir. O zamanlarda plastisite nöral mekanizmaların molekülleriyle ilgili hücresel düzeyde çok az bilgiler vardı. Yani Donald Hebb tarafından biyolojik hücreye dayalı olarak geliştirilmiş olan Hebb algoritması bilinen çok eski ve en ünlü öğrenme algoritmasıdır. Bir sinir hücresi diğer bir sinir hücresinden bilgi (girdi) alıyorsa ve eğer her ikisi de aynı işareti taşıyorsa (matematiksel olarak) yani aktif ise bu iki hücre arasındaki bağlantı kuvvetlendirilmelidir. Tersisi durumda ise zayıflatılmalıdır. Bir hücrenin kendisi pasif ise bağlantı kurduğu hücreyi pasif yapmaya yok eğer aktif ise aktif yapmaya çalışmaktadır. Diğer öğrenme kurallarının çoğu bu felsefeyi temel alarak geliştirilmiştir (Hebb, 1949: 45-78; Sejnowski ve Tesauro, 1989: 94-103; Öztemel, 2003).

2.5.2. Delta Kuralı

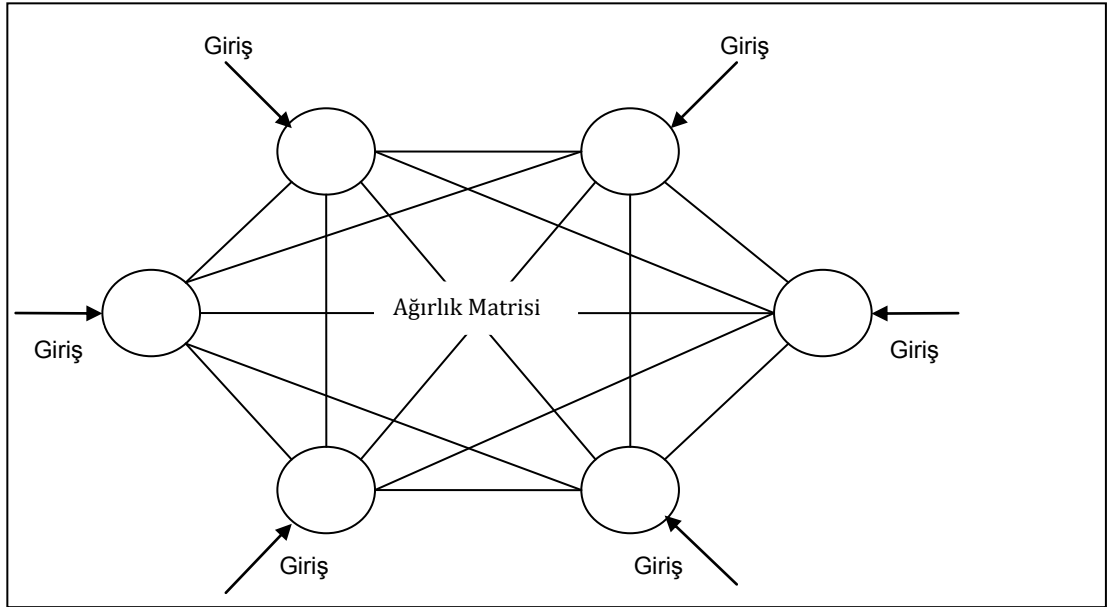
1949'da Hebb tarafından ortaya atılan bu öğrenme kuralı diğer öğrenme kurallarına yol göstermiştir. Bu kural Hebb kuralının biraz daha geliştirilmiş halidir. Örneğin iki sinir hücresi birbirine bağlanır ve aynı anda harekete başlarsa bu iki sinir hücresi arasındaki ilişki daha da güçlenir (Elmas, 2003: 50). Hebb'in ortaya koyduğu basit kurallar 1962'de ilk olarak Bernard Widrow ve Ted Hoff tarafından karmaşık bir şekilde ifade edilmiştir. Bu algoritma Widrow Hoff veya en küçük kareler yöntemi olarak da bilinmektedir. Delta kuralına göre belirli bir giriş vektörü için, çıkış vektörünün doğru cevabıyla karşılaştırır. Bunların farkı sıfırsa, herhangi bir öğrenme gerçekleşmez, aksi halinde ise ağırlıklar bu farkı azaltmak için ayarlanır. Aslında hedef çıktıyla gerçekleşen çıktı arasındaki farkı azaltmak için yapay sinir ağı bağlantılarının ağırlık değerlerini sürekli değiştirmesidir. Bu algoritma, hata karelerinin ortalamasını alarak, bu değerin en küçük olduğu

çözümü bulmaya amaçlar (Widrow ve Hoff, 1960: 96-104; Widrow ve Lehr, 1990: 1415-1442; Öztemel, 2003).

2.5.3. Hopfield Kuralı

1982’de John Hopfield tarafından geliştirilen bir danışmansız öğrenme ağıdır. Bu sinir ağında çok basit bir ağ sıralaması vardır. Bağlı nöronlar tam olarak tek bir tabaka oluşturur. Bu nöronların her biri birbirine bağlanmıştır. Hopfield sinir ağı, modelleri tanımak için eğitilmiş olabilir. Aşağıda bir Hopfield sinir ağı görebilirsiniz. Hopfield ağı tek katmanlı ve geri dönüşümlü bir ağıdır. Ayrıca, Şekil 2.10’da görüldüğü gibi bağlantılardan çift yönlü bilgi akmakta ve simetriktir. İşlem elemanlarının tamamı hem girdi hem çıktı elemanlarıdır. Her bağlantı için bir ağırlık değeri vardır (Hopfield, 1982: 2554-2558).

Hopfield, dinamik sinir ağ çalışmalarına öncülük yapmış ve son zamanlarda gecikme zamanı ile birçok alanda uygulanmaktadır. Bu özellikle Cohen-Grossberg ile 1983’te yapılan çalışmasıyla çok ilginç hale gelmiştir. Potansiyel uygulamaları şekil tanıma, paralel dağıtım, bellek ilişkileri ve optimizasyon yöntemlerinde kullanılmaktadır. Böyle uygulamalar yoğun şekilde dinamik davranışlara bağlıdır (Gan, 2012: 3040-3049; Wang, Lin ve Wang, 2012: 1353-1361).



Şekil 2.10. Bir Hopfield Ağı Örneği

Kaynak: Elmas, 2003: 75

2.5.4. Kohonen (Kompetitif) Kuralı

Bu yönteme göre ağı'n hücreleri, ağırlıkları değiştirmek için birbirleriyle yarışır. En büyük çıktıyı üreten ağı kazanan hücre ve bağlantı ağırlıkları değiştirilmektedir. Bunun anlamı, o hücrenin yanındaki hücelere karşı daha güçlü hale gelmesi demektir (Öztemel, 2003: 27).

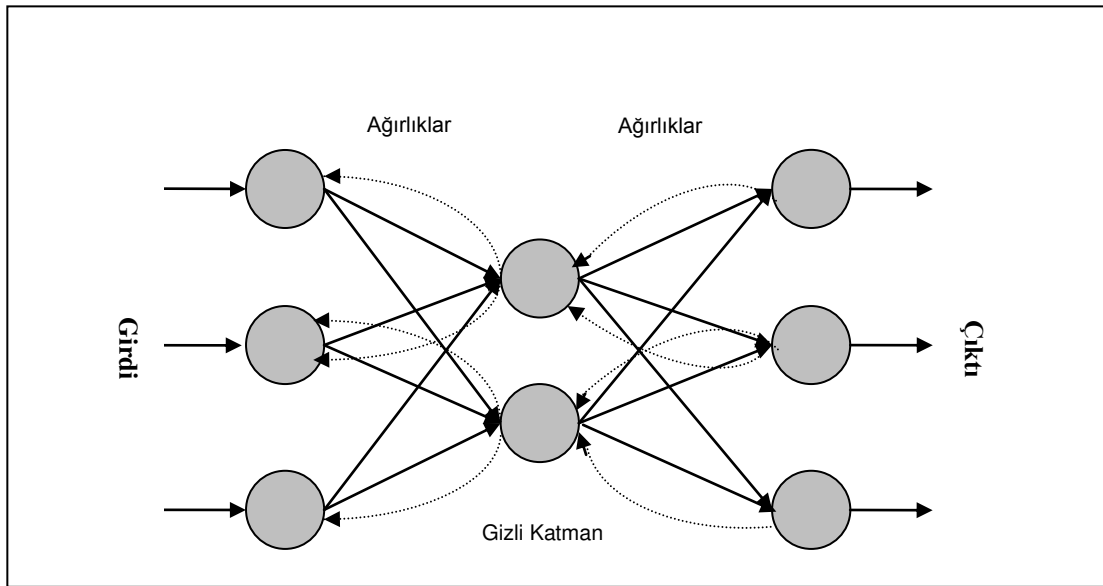
Bu yöntemde bir hedef olmasına gerek yoktur. Bu nedenle kendi kendine yani danışmansız olarak eğitimini tamamlar. Bilgi işleme elemanları bize bir birimin yakın komşularının belirlememizi sağlar. Bilgi işleme birimleri komşularının öğrenme sırasında ağırlıklarını güncelleştirilir, çünkü bu çok önemlidir. Böyle bir öğrenme yaklaşımının hedefi komşu birimlerinden ilgili sinyalleri tepki olarak öğrenme gerçekleştirmesidir (Rojas, 1996).

2.5.5. Geri Yayılım Ağı

Tek katmanlı sinir ağı için pek çok kısıtlama var. Bu ağlar oldukça sınırlı görevleri başarabilir. Minsky ve Papert iki katmanlı ileri beslemeli ağların pek çok kısıtlamanın üstesinden gelebildiğini gösterdi ama "gizli katmanlar için giriş ağırlıkları nasıl ayarlanacağı ?" soruna bir çözüm veremedi. Bu sorunun cevabını 1986 yılında Rumelhart, Hinton ve Williams tarafından sunuldu. Bu çözümün ardında yatan temel fikir hata geri yayılımının gizli katman birimleri ile çıkış katmanı birimlerinin hataları tespit etmesidir. Aslında bu yöntem tekrar tekrar ağına gönderilen bilgilerin gerçek çıkış vektörüyle arzu edilen çıkış vektörü arasındaki farkın ölçüsünü en aza indirecek şekilde ağ bağlantılarının ağırlıklarını ayarlar. Bu yöntem geri yayımlı öğrenme kuralı denir (Rumelhart, Hinton ve Williams, 1986: 533-536).

Geri yayılım ağı, 1970'lerin başında geliştirilmiş çok popüler ve karmaşık, tanımlanamamış doğrusal olmayan problemlere çözümler getirebilen bir ağ çeşididir. Tipik bir geri yayılım ağının bir girdi katmanı, çıktı katmanı ve en az bir gizli katmanı vardır (Elmas, 2003: 69-70).

Algoritma, hataları çıkış noktasından geriye doğru azaltmaya çalışmasından dolayı geri yayılım ismini almıştır (Sağıroğlu, Beşdok ve Erler, 2003: 85). Geri yayılım ağı, yapay sinir ağları içerisinde çoğu probleme rahatlıkla uygulanabilir bir yapıyı oluşturmaktadır. Geri yayılım ağı, geri beslemeli bir öğrenme sistemi kullanılır. Burada ağın yapısı ileri beslemeli olmasına karşın, hatanın geriye doğru yayılmasından dolayı bir öğrenme olduğu için geri besleme söz konusudur. Aktivasyon fonksiyonu için türevi alınabilecek bir fonksiyonu işleme sokulmaktadır. Çoğunlukla sigmoid fonksiyonunu kullanan geri yayılım ağı, öğrenme fonksiyonu olarak da Delta kuralını kullanır (Endar, 2010: 31).



Şekil 2.11. Geri Yayılım Ağı

2.5.6. Levenberg – Marquardt Öğrenme Algoritması

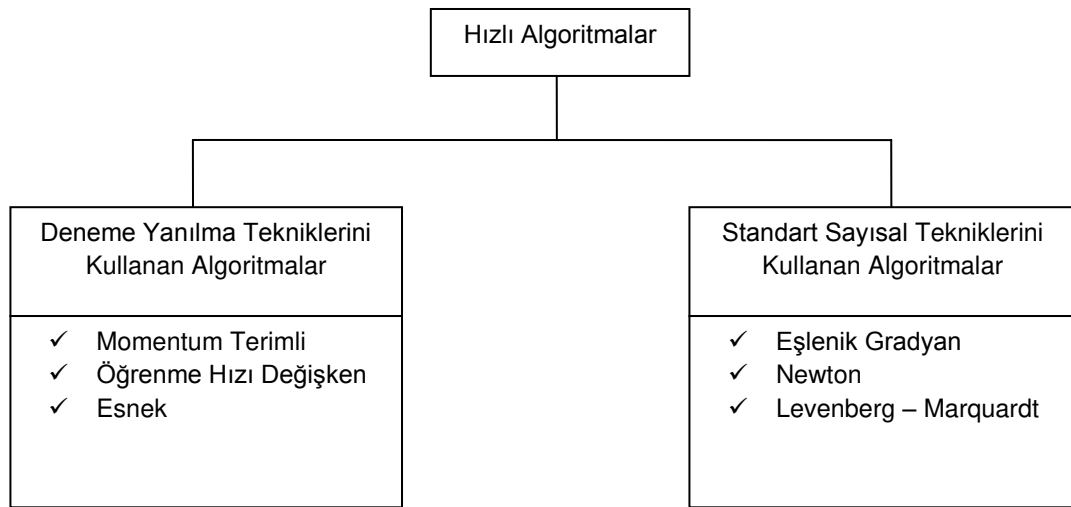
Doğrusal olmayan parametrelerin en küçük kareler kestirimi için iki yaklaşım vardır. Bir yandan, Taylor model serisi ve yerel doğrusallık varsayımıyla her adımda hesaplanan çeşitli parametrelerle düzeltmeler yapılır. Diğer taraftan, çeşitli modifikasyonlarda adım azaltmalı yöntemi kullanılır. Öğrenmede kullanılan geri yayılım algoritmasının farklı öğrenme tekniklerinden birisi olup, optimizasyon yöntemlerinde oldukça başarılı bir algoritmadır. Levenberg-Marquardt algoritması, maksimum komşuluk fikri üzerine kurulmuş “Taylor Serisi” ile “Gradient” bir hesaplama metodu olup en küçük kareler hesaplama yöntemidir (Marquardt, 1963: 431-441).

Gauss-Newton ve adım azaltmalı algoritmalarının en iyi özelliklerin birleşimi olup, birinci dereceden türev (Hessian) yaklaşımı üzerine dayanan oldukça hassas bir tekniktir. Bu öğrenme algoritması hızlı öğrenme ve iyi yakınsayabilme özelliğine sahiptir (Sağıroğlu, Beşdok ve Erler, 2003: 23-50).

Levenberg-Marquardt algoritması son yıllarda çok kullanılan ve diğer algoritmalara alternatif olarak geliştirilmiştir. Şekil 2.12’de hızlı ve yüksek

performanslı algoritmaların 2'ye ayrıldığı görülmektedir (Doğan, 2010: 61). Bunlar;

1. Deneme yanılma tekniklerini kullanan algoritmalar,
2. Standart sayısal optimizasyon yöntemlerini kullanan algoritmalar.



Şekil 2.12. Hızlı ve Yüksek Performanslı Algoritmalar

Kaynak: Hanbay, 2007

Ağırlıkların yenilenebilmesi için yapılması gereken ilk adım Hessian matrisini elde etmektir. Hessian matrisi, performans ağırlıklara göre ikinci dereceden türevlerinin alınmasıyla oluşturulur.

$$h = \begin{bmatrix} \frac{\partial^2 P(m)}{\partial w_1^2} & \frac{\partial^2 P(m)}{\partial w_1 \partial w_2} & \dots & \frac{\partial^2 P(m)}{\partial w_1 \partial w_n} \\ \frac{\partial^2 P(m)}{\partial w_2 \partial w_1} & \frac{\partial^2 P(m)}{\partial w_2^2} & \dots & \frac{\partial^2 P(m)}{\partial w_2 \partial w_n} \\ \vdots & \vdots & \dots & \vdots \\ \frac{\partial^2 P(m)}{\partial w_n \partial w_1} & \frac{\partial^2 P(m)}{\partial w_n \partial w_2} & \dots & \frac{\partial^2 P(m)}{\partial w_n^2} \end{bmatrix} \quad (13)$$

h= Hessian matrisi

p= Performans Fonksiyonu

w =Ağın sinaptik ağırlığı

$$h(m) = \frac{\partial^2 P(m)}{\partial w^2(m-1)}$$

Yapay sinir ağıları için Hessian matrisinin hesaplanması zordur. Bu nedenle Levenberg Marquardt algoritması Gauss-Newton matrisin yaklaşık değerini kullanmaktadır. Levenberg-Marquardt algoritması için Hessian matrisinin yaklaşık değeri aşağıdaki şekildeki eşitlik ile bulunur (Kaftan, 2011: 22-27).

$$h(m) \approx J^T(m) J(m) + \mu I \quad (14)$$

$J(m)$ = Jakobiye matrisi

$J^T(m)$ = Jakobiye matrisin tersi

I = Birim matris

m = Adım sayısı

μ = Marquardt parametresi

Jakobiye matrisi, hesaplamada Hessian matrisinden daha kolay olduğu için tercih edilir. Ağ hataları ağırlıklara göre birinci türevlerinden oluşmaktadır. Jakobiye matrisi,

$$J = \begin{bmatrix} \frac{\partial e_1}{\partial w_1} & \frac{\partial e_1}{\partial w_2} & \dots & \frac{\partial e_1}{\partial w_n} \\ \frac{\partial e_2}{\partial w_1} & \frac{\partial e_2}{\partial w_2} & \dots & \frac{\partial e_2}{\partial w_n} \\ \vdots & \vdots & \dots & \vdots \\ \frac{\partial e_i}{\partial w_1} & \frac{\partial e_i}{\partial w_2} & \dots & \frac{\partial e_i}{\partial w_n} \end{bmatrix} \quad (15)$$

Ağın gradyanı ise Eşitlik 16'deki bağıntıyla elde edilir;

$$G(m) = J^T(m) E(m) \quad (16)$$

Levenberg-Marquardt algoritması için ağırlık değişim bağıntısı;

$$a(m+1) = a(m) - [J^T(m)J(m) + \mu I]^{-1} J^T(m) E(m) \quad (17)$$

Bu yöntemin

- 1.Kural: μ sıfır çıkarsa Hessian matrisini kullanan Newton algoritması,
- 2.Kural: μ büyük bir sayı ise küçük adımlı Gradyen azalması haline gelir.

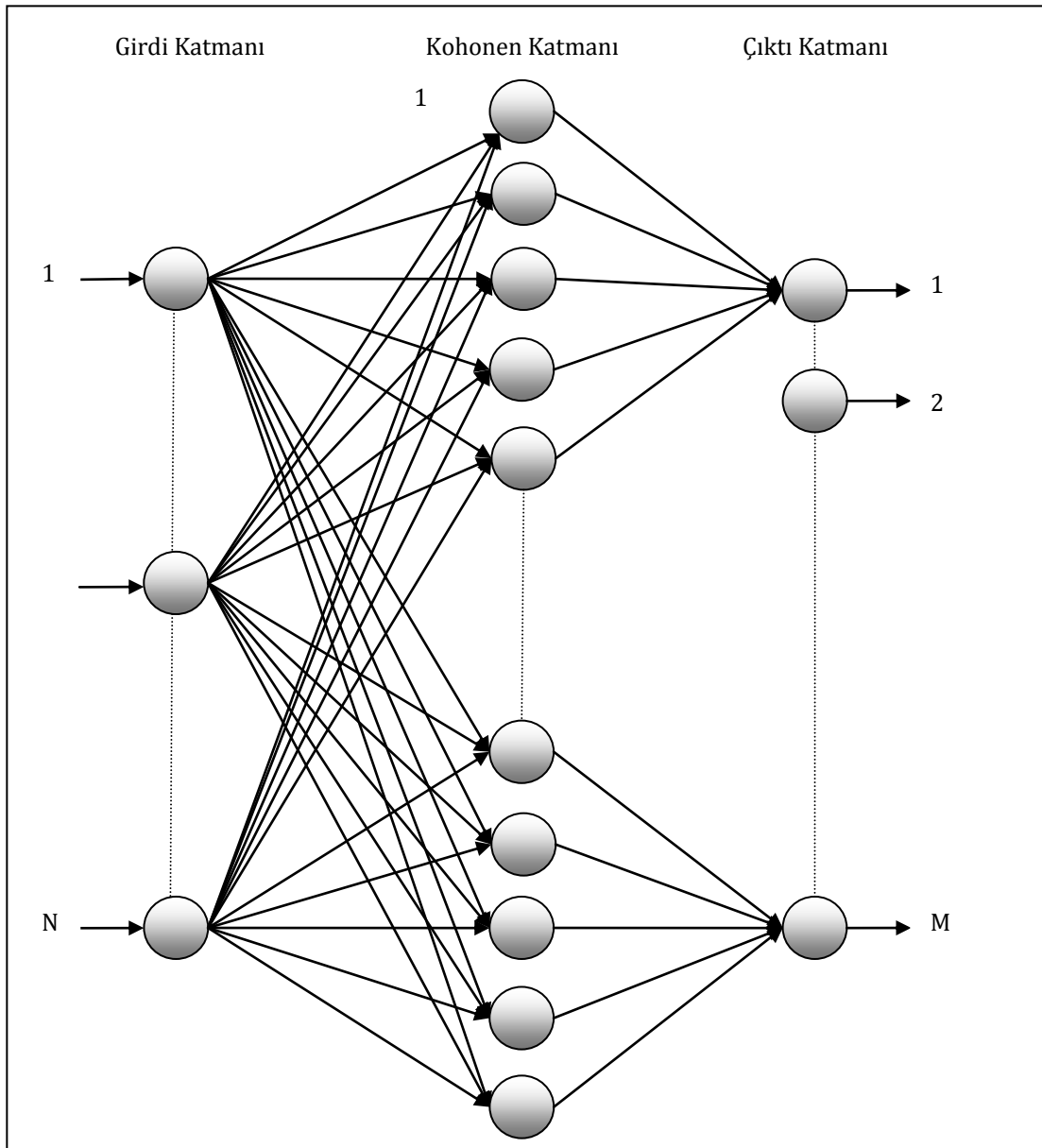
2.5.7. LVQ (Öğrenmeli Vektör Kuantalama)

LVQ ağı 1984 yılında Tuevo Kohonen tarafından geliştirilmiştir. Bu modele aynı zamanda Kohonen ağı olarakta bilinmektedir (Kohonen, 1984). Kohonen katmanı n boyutlu bir vektörü bir vektörler setine haritalamaktır.

2.5.7.1. LVQ Ağının Yapısı

LVQ ağları da 3 katmandan oluşmaktadır. İlk katman girdi katmanı, ikinci katmanı Kohonen katmanı ve diğer bir ifadeyle ara katmandır. Üçüncü katman çıktı katmanı olup girdinin ait olduğu sınıfı belirler.

LVQ ağları, girdi ve Kohonen katmanları arasında tam bağlantılı, Kohonen ve çıktı katmanları arasındaysa kısmi bağlantı olmaktadır. Aslında girdi katmanındaki her işlemci eleman Kohonen katmanındaki tüm işlemci elemanlarla bağlantılıdır. Kohonen katmanındaki elemanlar ise çıktı katmanındaki bir tek elemanla bağlantılıdır. Kohonen katmanı ile çıktı katmanı arasındaki ağırlıklar (α) sabit olup 1'e eşittir ve bu ağırlıklar değişmez. Sadece Kohonen katmanı ile girdi katmanı arasındaki ağırlıklar değiştirilir (Öztemel, 2003).



Şekil 2.13. LVQ Ağının Topolojik Yapısı

2.5.8. Hızlı Yayılım (Quick Propagation)

Hızlı yayılım (QP) algoritması, Falhman tarafından geliştirilmiştir. Geri yayılım ağı, her bir ağırlık bakımından toplam hatanın kısmi birinci türevini hesaplayarak çalışır. Bu bilgiler ele alındığında ağırlık uzayında azalan gradyan yapabilir. Hızlı yayılım algoritması ve sezgisel bir algoritmadır. (Fahlman, 1988). Bazı problem çözümlerinde Geri Yayılımdan daha hızlı

olabilir. Ancak bazen eğitim sırasında çözüm bir noktaya takılabilir (Bilen, 2011)

Sezgisel algoritmalar, tecrübeye ve eğitilmiş tahmine dayalı kurallar içerir. Bu algoritmalar optimum çözüme yakın çözüm bulmayı garanti etmektedir. Sezgisel metotların bazı üstünlükleri;

- Karar verici mekanizması için sadeleştirici olması,
- Bir yöntemin parçası olarak öğrenme amacıyla kullanılabilmeleri (Sağıroğlu, Beşdok ve Erler, 2003),

2.5.9. Eşleştirmeli Eğilim (Conjugate Gradient) Algoritmaları

Eğim azatlım doğrultusunda hızlı yakınsayan eşleştirme doğrultusunda bir arama yapar. Performans fonksiyonunu o doğrultuda minimize edebileceği boyutu belirler. Eşleştirmeli eğim yöntemi her iterasyonda güncellenir. Bu gibi tekrarlamalı yöntemler seyrek matrislerin kullanımı için uygundur. Bu sistemler, sonlu farklar ve kısmi diferansiyel denklemler, yapısal analiz, devre analizi çözümü için sonlu elemanlar yöntemleri gibi birçok önemli alanlarda ortaya çıkar (Shewchuk, 1994).

2.5.10. Kuasi-Newton (Quasi Newton) Öğrenme Algoritması

Kuasi-Newton, eşleştirmeli eğim yöntemine alternatif olarak hızlı optimizasyon için geliştirilmiştir. Hesian matrisinin hesaplanması zor ve zaman alıcı olduğundan dolayı bu matrisin hesaplanmasına gerek olmayan yöntemler geliştirilmiş ve bunlara Kuasi-Newton ismi verilmiştir (Sağıroğlu, Beşdok ve Erler, 2003). Gradyan kökenli Newton optimizasyon yöntemleri kullanılarak, eğitim ile hata en aza indirilebilir. Bu yöntem Newton optimizasyonu için gerekli yaklaşık Hessian matrisi türev bilgilerinden yararlanabilir (Likas ve Stafylopatis, 2000).

2.5.11. Online Geri Yayılım (Online Back Propagation)

Bu algoritma eğitimlerde kullanılan bilindik bir yöntemdir. Öğretici sinaps tarafından gradyan yöntemine göre, geriye taşıma mekanizmasıyla “bias ve sinaps” ağırlıklarını ayarlar. Bias ve ağırlıkları ayarladığı için “On-Line” denilmektedir Algoritmanın çalışmasında iki parametre kullanılır: bunlar öğrenme ve momentum katsayılarıdır. Bu parametreler öğrenme öncesi ayarlanmış olmalı ve bu katsayıların en uygun değerlerini birkaç deneme yaptıktan sonra bulabilir (Ahmad, Ansari ve Mohsin, 2008).

2.5.12. Grup Geri Yayılım (Batch Back Propagation) Algoritması

Bu yöntem on-line algoritmasının bir türevidir. Bu yaklaşımı diğerlerinden ayıran özellik, her girişte ağırlıklarının değiştirilmesi yerine veri setinin tüm ağa uygulanmasından elde edilen sonuca göre ağırlıklar değiştirilir. Bu algoritma, on-line çeşidiyle aynı parametreler kullanır. Ayrıca başka bir parametre ise “batch size” adındadır. Ağırlık/Bias ayarlamaları her epok (epoch)’un sonunda uygulanır (Ahmad, Ansari ve Mohsin, 2008).

2.6. YAPAY SİNİR AĞLARIN UYGULAMA ALANLARI

Yapay sinir ağlarıyla değişik alanlarda binlerce uygulamanın yapıldığı ve başarılı sonuçların elde edildiği söylenebilir. İlk yapay sinir ağı çalışması yaklaşık altmış yıl önce başlamış olmasına karşılık, bu yöntemin etkin olarak kullanımı ve gelişimi Rumelhart ve arkadaşlarının 1982 ve 1984 yılında yayınlanan çalışmalarıyla sinir ağların gelişimine önemli katkıda bulunmuştur. Oysaki diğer iyileştirme ve matematiksel yöntemlerin gelişimi çok uzun yıllar sürmüştür. YSA yöntemi; yüksek eğitim, tıp, borsa, araştırma, endüstri ve teknolojik gelişmeler için, Bulanık Mantık (BM) ve Genetik Algoritmalar (GA) yöntemleriyle beraber çok başarılı bir temel teşkil etmektedir. Uygulama alanına göre kullanılan yapay sinir ağları aşağıda ifade edilmiştir (Karahana, 2011: 61-62; Elmas, 2003:178).

Endüstriyel Uygulamalar: Sinir ağları sanayideki kalite denetiminde, kalitenin geliştirilmesinde yaygın olarak kullanılmaktadır.

- Gezgin satıcı problemi,
- Cep telefonlarında ses ile çalışabilme,
- Optimizasyon uygulamalarında (üretim planlama ve kontrol çalışmaları),
- İmalatta, ürün tasarımında, süreç ve makinelerin bakımında,
- Kimyasal süreçlerin dinamik modellenmesi (Öztemel, 2003: 205),

Ulaştırma ve Havacılık Uygulamaları: Birçok otomobil imalatçısı, otomobillerde ve otomobil ürünlerinde sinir ağı kullanımını daha yaygın kullanabilmek için birçok araştırma yapmaktadırlar. Fren denetimi, mühendislik hataları ve uygulamaları gibi bazı alanlarda umut verici sonuçlar elde etmişler. Uçak motorlarında ve NASA'da, uzay mekiği yolculuklarında manevra denetimi için yapay sinir ağları çalışmaları uygulanmaktadır (Elmas, 2003:177).

Borsa ve Para Uygulamaları: Yapay sinir ağlarının kullanılabileceği diğer uygulama alanıysa borsa ve para yönetimi işlemleridir (Efe ve Kaynak, 2000: 110). Yapay sinir ağları Citibank ve World Bank'ın gibi kuruluşlar ve gelişmiş firmaların büyük bölümünde mali tahmin için kullanılmaktadır. American Express, Mellon Bank, First USA Bank gibi bazı bankalar ve kredi kartı şirketleri, sahtekarlıkları önlemek amacıyla sürekli sinir ağı kullanmaktadırlar. Son zamanlarda sinir ağları el yazısı ve imza tanımlanması için kullanılmaktadır (Elmas, 2003: 177).

Askeri ve Savunma Uygulamaları: Yapay sinir ağları askeri alandaki uygulamaları da son zamanlarda dikkat çekmektedir. Aşağıdakiler örnek olarak gösterilebilir:

- Silah otomasyonu,
- Hedef seçme/izleme,
- Radar ve izleme,
- Sensör sonar sistemleri-nesneleri/görüntüleri ayırma ve tanıma, sinyal işleme, görüntü işleme vb. gibi,
- Mayın detektörleri,
- Algılayıcı tasarımı ve gürültü önleme,

Sağlık ve İlaç Sanayi Uygulamaları: Yapay sinir ağları, tıbbi sinyallerin analizi, kanserli hücrelerin belirlenmesi, protez tasarımı, transplantasyon zamanlarının optimizasyonu ve hastane giderlerin optimizasyonu gibi konularda uygulanmaktadır.

- Hastalıkların teşhisi,
- EEG analizleri,
- Hastaların ilaçlara karşı davranışlarının incelenmesinde,
- İlaç sanayisinde,

İletişim Sanayi Uygulamaları: YSA, telefon ve internet gibi iletişim sistemlerin güncellenmesinde kullanılabilmektedir.

Tablo 2.2. Yapay Sinir Ağları ve Uygulama Alanları

Uygulama	Ağ Türü
Öngörü	Geri-Yayılım Ağı Delta Bar Delta Genişletilmiş Delta Bar Delta Yüksek Seviyeli Ağlar Özörgütlemeli Harita Ağı Algılayıcılar
Sınıflandırma	Geri Yayılım Ağı Algılayıcılar Olasılıksal Sinir Ağları Kohonen Ağı Boltzmann Makinesi
Veri Birleştirilmesi	Hopfield Boltzmann Makinesi Hamming Ağı İki Yönlü Çağırışım Belleği
Veri Kavramlaştırılması	Uyarlanır Rezonans Ağı Özörgütlemeli Harita Ağı
Veri Süzülmesi	Yeniden Dolaşım
Resim veya Görüntü İşleme	Geri-Yayılım Ağı Algılayıcılar

Kaynak: Öztemel, 2003: 176

2.7.YAPAY SİNİR AĞLARINA YÖNELTİLEN ELEŞTİRİLER

Ağ yapısının belirlenmesinde kural olmaması: Sinir ağlarına ilişkin yapı belirlenirken belirli bir kural yoktur. Gizli katman sayısının rastgele belirlenmesi ve beklenen sonuca uygun bir şekilde modellenmesi, yapay sinir ağlarına karşı ortaya atılan önemli bir eleştiridir.

Ağın parametre değerlerinin belirlenmesinde kural olmaması: Bir yapay sinir ağının katsayı parametreleri keyfi olarak seçilmesi, bu katsayı değıştikçe farklı sonuçlar üretmesi ve çıktı sonuçlarının performansına bakılarak atanması eleştiri konusudur.

Ağ davranışlarının açıklanamaması: Eldeki veri setine aynı öğrenme algoritması ve mimari yapı kullanılmasına karşın örneğin test setinin seçiminin farklı oluşması nedeniyle, her eğitim sonrasında elde edilen performans farklı olabilmektedir. Aynı yapının farklı sonuçlar üretmesi, bu farklılığın tam olarak açıklanamaması diğer bir eleştiri konusudur (Özün, 2011: 59-62).

2.8.YAPAY SİNİR AĞLARI METODUNUN DİĞER METOTLARLA KARŞILAŞTIRILMASI

Yapay sinir ağlarında kullanılan terimler farklı terminolojiye sahiptir. Aşağıdaki tablolarda yapay sinir ağları ve istatistiksel yöntemlerin terminolojileri, geleneksel talep tahmin metotları ve uzman sistemlerle karşılaştırılması gösterilmiştir.

Tablo 2.3. YSA'ların İstatistiksel Yöntemler ile Terminolojik Benzeşimi

Yapay Sinir Ağları	İstatistiksel
Ağ	Model
Öğrenme	Tahmin
Danışmalı Öğrenme	Regresyon
Öğrenme Algoritması	Gözlem
Ağ Parametreleri	Parametre
Giriş verileri	Bağımsız Değişken
Çıkış verileri	Bağımlı Değişken

Kaynak: Sağiroğlu, Beşdok ve Erler, 2003: 23-50

Tablo 2.4. YSA ve Geleneksel Talep Tahmin Metotlarının Karşılaştırılması

	Geleneksel Metotlar	Yapay Sinir Ağları
Öğrenme süresi	---	Uzun
Cevap verme süresi	Uzun	Kısa
Bilgi temsili	Zor	Kolay
Öncelikli bilgi	Gerekli	Sadece örnekler gerekli
Rastgele seçilmiş problem	Zor	Kolay
Performans	Uygun	İyi
Bilgi işleme	Açık	Kapalı
Kaynak	Uzman Kişi	Örnekler

Kaynak: Toktaş ve Aktürk, 2004: 20

Tablo 2.5. Uzman Sistemler ve Yapay Sinir Ağları Arasındaki Farklılıklar

Parametre	Uzman Sistemler	Yapay Sinir Ağları
Kullanıcı arabirimi	Mevcuttur	Mevcut değildir
Açıklama yeteneği	Mevcuttur	Mevcut değildir
Uzman kişi	İhtiyacı var	İhtiyacı yoktur
Örnekler	İhtiyacı yoktur	İhtiyacı var
Problemin boyutu çoğaldığında bakım	Zorlaşır	Değişmemektedir
Oluşturulması	Kurallar, hiyerarşik yapılar	Eğitilmekle
Bilgi işleme	Seri ve karmaşık	Paralel
Bilgi temsili	Kesin ve açık/Sözlü ifade	Saklı/sözlü ve sayısal ifade
Tam değerlerin belli olması	Şarttır	Şart değildir

Kaynak: Toktas ve Aktürk, 2004: 16

2.9. YAPAY SİNİR AĞLARI UYGULAMA ÖRNEKLERİ

2.9.1. ENDÜSTRİYEL UYGULAMA

Çok Katmanlı Ağlar, mühendislik problemlerinin çoğuna çözüm üretebilme yeteneğine sahiptir. Bu sebeple imalat sektöründe bir uygulama örneği ele alınmıştır.

2.9.1.1. Problemin Tanımlanması

Öztemel (2003) kitabında piston üreten bir fabrikada piston kalitesinin artırılması için bir çalışma yapılmıştır. Piston üreten fabrikada üretim proses dökümden başlar ve son ürüne kadar devam eder. Döküm bölümünde piston yapılmak üzere uzun borular halinde tüplere dökülmektedir. Döküm savurma tekniğiyle yapılmaktadır. Döküm işlemleri esnasında tüplerin dış yüzeyleri ısı iletkenliği çok az olan bir boya ile boyanmaktadır. Döküm yapılıp söküldükten sonra tüplerin dış yüzeylerindeki boya temizlenmektedir. Ancak bilinmeyen nedenler ötürü dökülen tüpler üzerinde bir eğim meydana gelmektedir. Bazı tüplerin dış yüzeyleri bu nedenle yeterli oranda temizlenememektedir. Oluşan bu eğime sehime denmektedir. Sehimin önlenmesi için ona neden olan etkin olan faktörlerin kontrol altında tutulması gerekmektedir.

Yapılan araştırma ve incelemeler neticesinde tüplerde meydana gelen sehime etki eden faktörler belirlenmiş ve aşağıda verilmiştir. Tablo 2.6'da gösterildiği gibi her faktörün iki değerinin olması söz konusudur.

Tablo 2.6. Piston Üzerindeki Sehime Neden olan Faktörler ve Değerleri

Faktörler	Değer 1	Değer 2
X-Kalıpta Soğutma Süresi	60 sn	75 sn
Y-Tüpün Kalıptan Çekilmesi	Tam	Yarım
Z-Tırnakla Kavrama Ayarı	Ayarlı	Ayarsız
T-Sephanın Isı İletimi	Boyalı	Boyasız
W-Dökümhane Giriş Kapısı	Açık	Kapalı
Q-Kalıp Boyama Süresi	45 sn	30 sn
P-Eriyik Sıcaklığı	1250 derece	1190 derece

2.9.1.2. Öğrenme Setinin Oluşturulması

Çok Katmanlı Ağı temsil eden 8 örnek belirlenmiş ve Tablo 2.7’de gösterilmiştir. Girdi seti yedi faktörden ve çıktı seti bir faktörden oluşmaktadır.

Tablo 2.7. Öğrenme seti

Deney	X	Y	Z	T	W	Q	P	Ç
1	60	Tam	Ayarsız	Boyalı	Kapalı	45	1250	1,55
2	60	Tam	Ayarsız	Boyasız	Açık	30	1190	1,60
3	60	Yarım	Ayarlı	Boyasız	Açık	30	1190	1,40
4	60	Yarım	Ayarlı	Boyalı	Kapalı	45	1250	1,34
5	75	Tam	Ayarlı	Boyasız	Kapalı	45	1190	1,31
6	75	Tam	Ayarlı	Boyalı	Açık	30	1250	1,45
7	75	Yarım	Ayarsız	Boyasız	Kapalı	30	1250	1,28
8	75	Yarım	Ayarsız	Boyalı	Açık	45	1190	1,49

Yukarıdaki Tablo 2.7’de görüldüğü gibi deneylerde kullanılan bütün değerler nümerik (sayısal) değildir. Sinir ağlarına sunulan verilerin nümerik olması şarttır. Bu yüzden nümerik değerlere dönüştürülmesi gerekmektedir. Bu örnekte her faktörün iki değeri vardır (1. değer için 1, ikinci değer için 2 rakamı o faktörün temsili değeri olarak seçilmiştir).

Tablo 2.8. Örneklerin Nümerik Gösterimi

Deney	X	Y	Z	T	W	Q	P	Ç
1	1	1	2	1	2	1	1	1,55
2	1	1	2	2	1	2	2	1,60
3	1	2	1	2	1	2	2	1,40
4	1	2	1	1	2	1	1	1,34
5	2	1	1	2	2	1	2	1,31
6	2	1	1	1	1	2	1	1,45
7	2	2	2	2	2	2	1	1,28
8	2	2	2	1	1	1	2	1,49

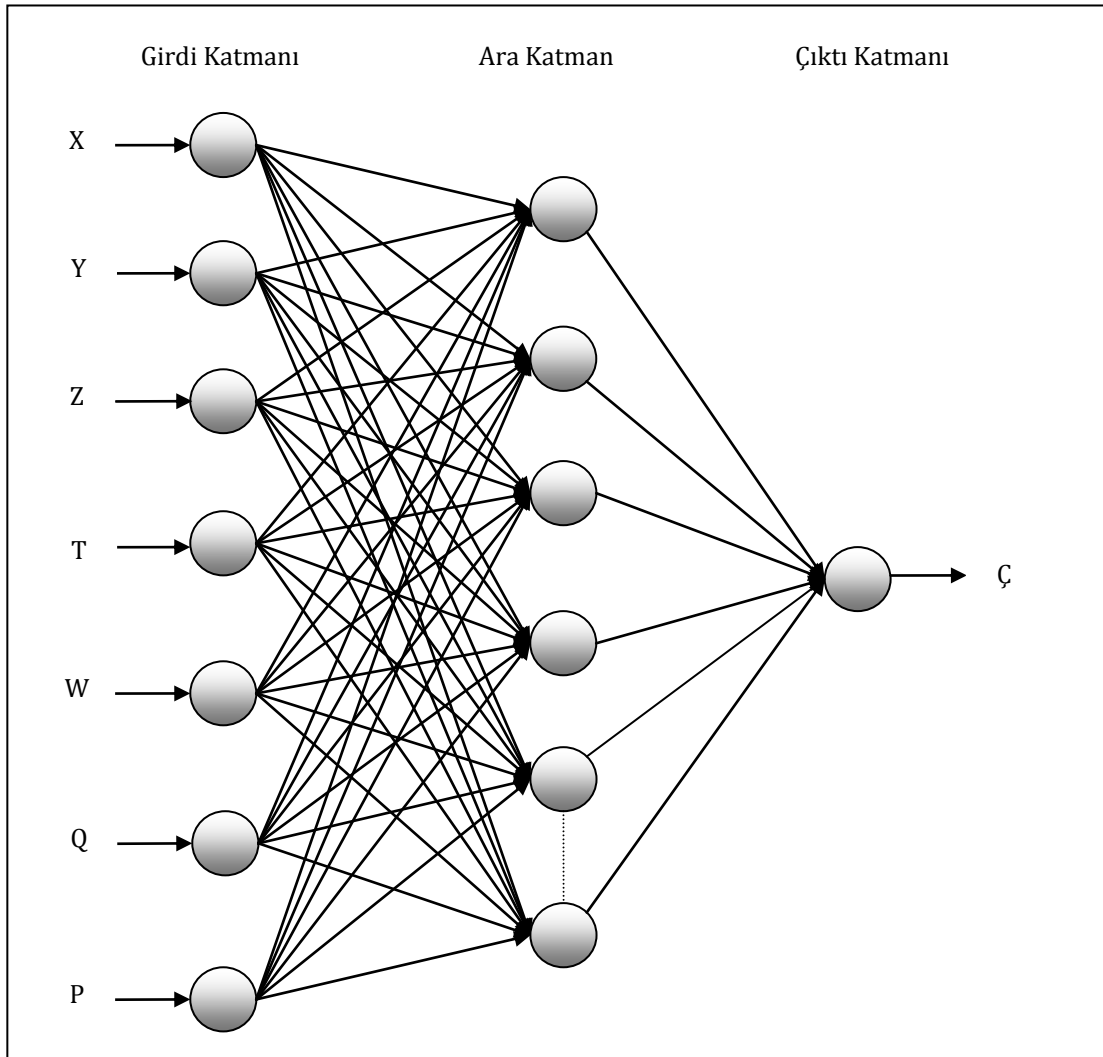
Problemin çözümünde aktivasyon fonksiyonu olarak sigmoid fonksiyonu kullanılacağından sonuçlar 0 ile 1 arasında çıkacaktır. Sinir ağı, çıktı olarak 1'den büyük bir değeri ve 0'dan küçük bir değeri üretmesi mümkün değildir. Bu nedenle girdiler ve çıktılar normalize (ölçeklendirilme) edilerek 0 ile 1 arasındaki değerlere dönüştürülür. Bütün değerler 20'ye bölünerek eğitim seti elde edilmiştir. Ağ eğitildikten sonra elde edilen çıktı değerleri 20 ile çarpılarak tahmin edilen sehim miktarı belirlenmiştir.

Tablo 2.9. Ölçeklendirilmiş Eğitim Seti

Deney	X	Y	Z	T	W	Q	P	Ç
1	0,05	0,05	0,1	0,05	0,1	0,05	0,05	0,0775
2	0,05	0,05	0,1	0,1	0,05	0,1	0,1	0,08
3	0,05	0,1	0,05	0,1	0,05	0,1	0,1	0,07
4	0,05	0,1	0,05	0,05	0,1	0,05	0,05	0,067
5	0,1	0,05	0,05	0,1	0,1	0,05	0,1	0,0655
6	0,1	0,05	0,05	0,05	0,05	0,1	0,05	0,0725
7	0,1	0,1	0,1	0,1	0,1	0,1	0,05	0,064
8	0,1	0,1	0,1	0,05	0,05	0,05	0,1	0,0745

2.9.1.3. Çok Katmanlı Ağın Oluşturulması

Sehimin oluşmasına neden olan faktörlerin sayısı 7 olduğu için 7 giriş ve bir çıktı ünitesi belirlenmiştir.



Şekil 2.14. Sehim Tahmini için ÇKA Modeli

2.9.2. İMALAT SEKTÖRÜNDE KALİTE KONTROL UYGULAMASI

2.9.2.1. LVQ-ÖRÜNTÜ TANIMA

Günümüzdeki işletmelerin en önemli sorunlarından birisi kalite kontroldür. İstatiksel süreç kontrolü yaklaşımıyla herhangi bir üretim hattı veya süreçteki değişimi ölçmek, analiz etmek ve kalite problemlerinin oluşmasını önlemek amaçlanmaktadır. Bunu gerçekleştirmek kolay değildir. Çünkü bu konuda süreçle ilgili yeterli bilgi ve tecrübe sahip olmak gerekir. Aynı zamanda üretim süreci hakkında otomatik bilgiler üretecek sistemlere ihtiyaç vardır.

2.9.2.1.1. Problemin Tanımlanması

İstatiksel kalite kontrolün temel amacı sistemde hatalar oluşmadan hataya sebep olacak etkenleri ortadan kaldırmaktır. Bunun için süreçleri sürekli izlemek gerekir. Deming'e göre değişkenlik iki tür nedenden kaynaklanmaktadır.

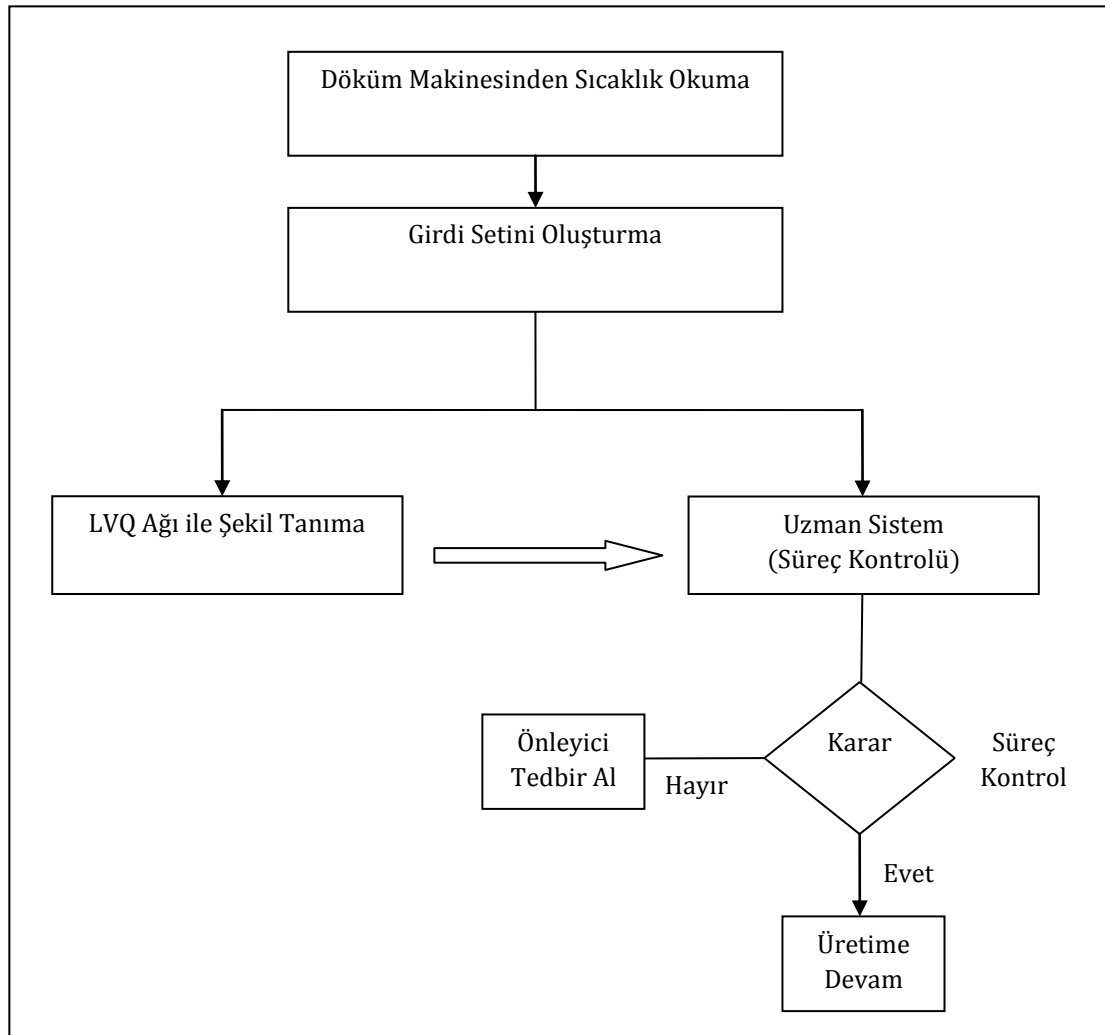
- Kontrol edilebilir faktörlerden kaynaklanan değişkenlik
- Kontrol edilemez faktörlerden kaynaklanan değişkenlik

İstatiksel kalite kontrolün temel felsefesinde, kontrol edilemeyen faktörler yer almamaktadır. Çünkü bu faktörleri değiştirmek hem zor hem de anlamsız olur. Önemli olan kontrol edilebilir faktörler üzerindeki değişkenliği kontrol altında tutmaktır. Kontrol şeması üzerinde değişik şekiller oluşabilmektedir. Örneğin her şeyin normal olması durumunda şema üzerinde normal dağılıma uygun şekil görülmektedir. Ölçülen değerlerin kalitesinde zamanla ortalamasında bir artma olursa o zaman şema üzerinde yükselen bir trend görülmektedir. Her şeklin oluşması durumunda alınacak tedbirler belirlidir. Önemli olan şeklin tanınmasıdır. Otomatik olarak şekilleri tanıyan bir sistem kurularak gerekli tedbirler sistem aracılığıyla operatörlere bildirilir ve gerekeni

yapmaları istenir. Bu yüzden şekillerin tanınması imalat endüstrisi için önemlidir.

Aşağıda çikolata üretimi yapan sıvı mamul dolum hattı sıcaklık kontrolünde, sıvı maddelerin kalıplara dökülmesi sırasında sıcaklığın kontrol edilmesine yönelik olarak geliştirilen bir kontrol şemalarının şekillerini tanıma sisteminden (örüntü tanıma sistemi) bahsedilecektir. Problem için geliştirilen bir uzman sistem YPC olarak adlandırılmıştır. Uzman sistem prosesin kısa dönemli davranışını kontrol edebilmektedir. LVQ ağı uzun dönemli davranışlarda, otomatik algılamalar için kullanılabilir. Bu ağ kontrol şemaları üzerinde oluşan şekilleri yorumlayarak uzun dönemli davranışları hakkında bilgi sahibi olmamızı sağlar. Klasik yöntemlerden oldukça farklı ve başarılı bir sistem önerilmiştir. Bu sistem makinelerde ölçülen sıcaklık değerlerini alarak sürecin kontrol altında olup olmadığını kontrol etmektedir. Eğer ölçülen sıcaklık değerleri normal dağılıma uyuyorsa bir sorun olmadığını aksi halde bir uzman sistem önerilmektedir.

Geliştirilen YPC sistemin yapısı Şekil 2.15' te gösterilmiştir. Sistemden okunan sıcaklık değerleri girdi vektörü olarak sisteme gönderilmektedir. Girdi vektörünün boyutu 30'dur. Bu durum sürekli ölçülen son 30 sıcaklık değerini almaktadır. Uzman sistemler daha çok sistem sürecinin kısa dönemli davranışlarını yorumlarken LVQ ise sistemin uzun dönemli davranışını incelemektedir (Öztemel, 2003).



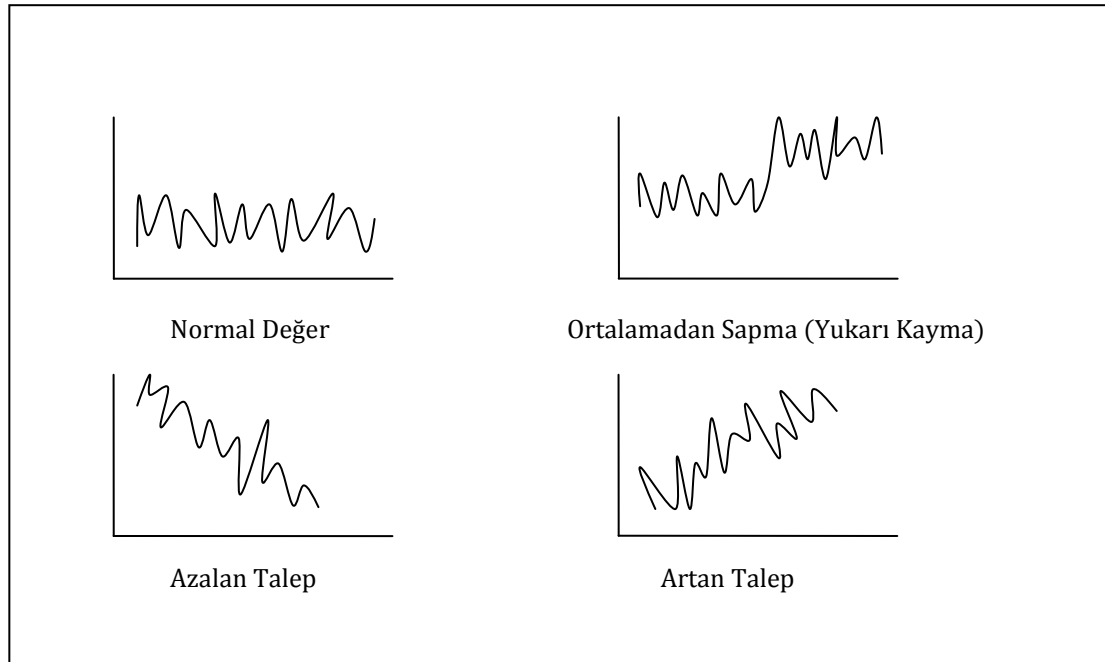
Şekil 2.15. İstatiksel Kalite Kontrol Sistemi

Kaynak: Öztemel, 2003: 115-135

YPC sisteminin LVQ ağı aşağıda Şekil 2.16' da gösterilen 4 adet şekli tanıyabilmektedir. Bu durum aslında ölçülen sıcaklık değerlerinin hangi sınıfa ait olduğunu belirlemektedir. Bu şekiller:

- **Trend:** Sıcaklık değerlerinin zaman içerisinde artması veya azalması durumudur.
- **Normal Değer:** Üretim sürecinde her şeyin normal olduğunu, üretimdeki değişikliğin kontrol edilemez faktörlerden kaynaklandığını gösterir.

- **Ortalamadan Sapma:** Üretilen ürünün kalite karakteristiğinde ortalama değerinde ani artış veya azalmanın olduğu durumdur.



Şekil 2.16. LVQ Ağının Tanıdığı Kontrol Şeması Üzerindeki Şekiller

2.9.2.1.2. Öğrenme Setinin Oluşturulması

Kontrol şemalarının örnekleri benzetim yoluyla üretilmiştir. Kontrol şemasındaki her şekli üreten matematiksel formülasyon belirlenmiştir. Pham ve Öztemel tarafından aşağıdaki gibi anlatılmıştır (Pham ve Öztemel, 1996).

2.9.2.1.2.1. Artan veya Azalan Trendin Üretilmesi

Artan veya Azalan Trend şekillerinin formülü Eşitlik 18'deki gibi ifade edilir.

$$\zeta(t) = \mu + k(t) * \sigma \pm m * t \quad (18)$$

$\zeta(t)$ = t. sıcaklık değerini,

μ = Beklenen ortalama sıcaklık değeri,

$k(t)$ = Normal dağılıma uygun üretilmiş rassal değer

m = Trendin eğiminin hangi aralıkta değişeceğini (örneğin 0,1 ile 0,4 arası)

σ = Ölçülen sıcaklık değerlerinin ortalamadan sapmalarını ifade etmesi

2.9.2.1.2.2. Normal Dağılım Şeklin Üretilmesi

Aşağıdaki Eşitlik 29'daki ifade kullanılarak normal dağılıma uygun şekil üretilir.

$$\zeta(t) = \mu + k(t) * \sigma \quad (19)$$

Bilgisayarların ürettiği değerleri normal dağılıma uygun hale getirmek için bir dönüşüm fonksiyonu kullanılmıştır. Bu fonksiyon;

$$\begin{aligned} r(t) \\ = \sqrt{-2\ln(r)} \cos 2\pi r \end{aligned} \quad (20)$$

u = Bilgisayar tarafından düzgün dağılıma uygun üretilmiş rassal değer

2.9.2.1.2.3. Yukarı veya Aşağı Doğru Ortalamadan Sapma

$$\zeta(t) = \mu + k(t) * \sigma \pm n * h \quad (21)$$

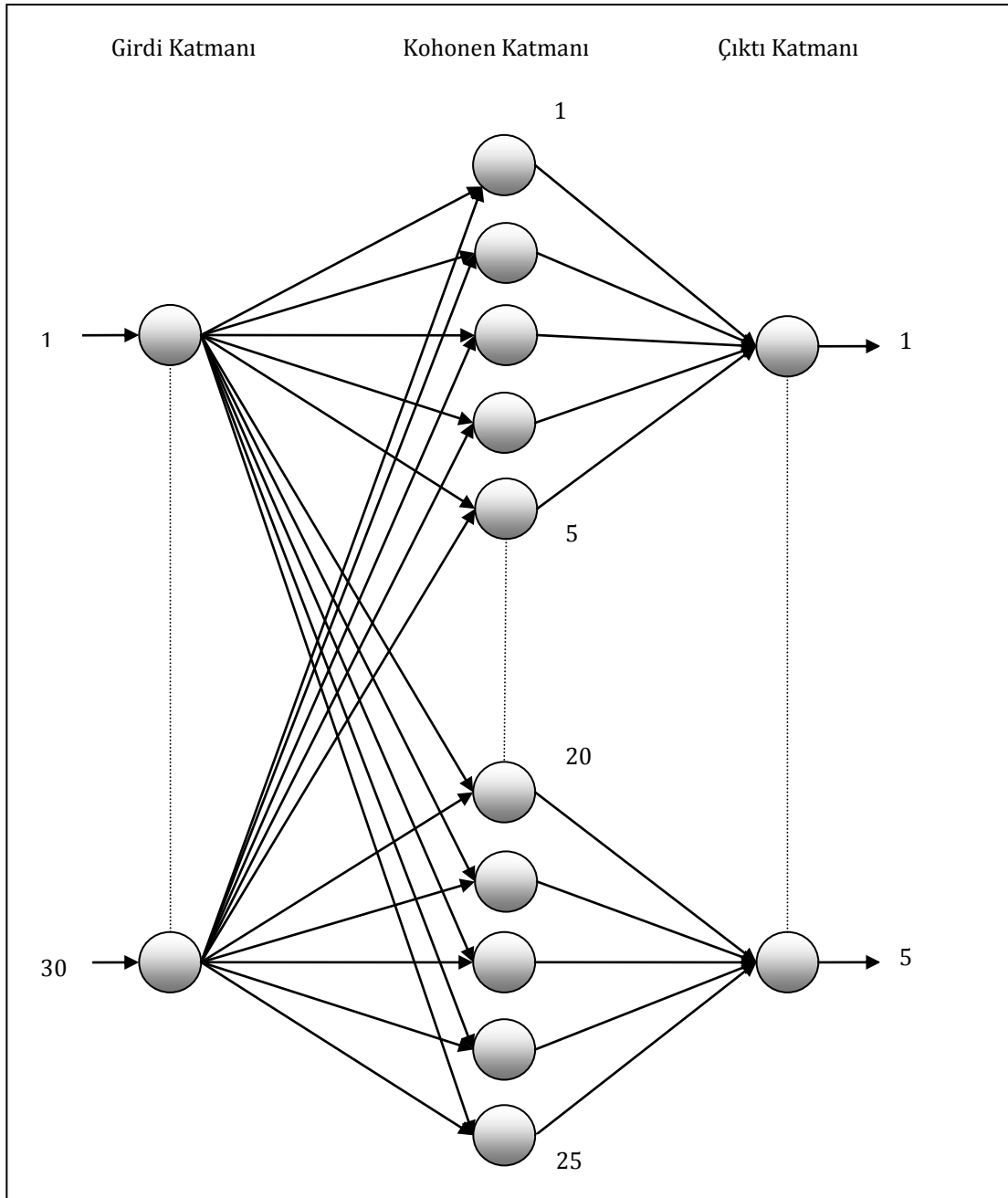
n = Sapma pozisyonu

h = Sapma miktarı

2.9.2.1.3. LVQ Ağıının Oluşturulması

LVQ ağı, kontrol şeması üzerindeki 5 şekli tanımasını istediği için 5 çıktı ünitesi belirlenmiştir. Her bir örüntü 30 sıcaklık değerinin ölçülmesi için

oluşturulduğundan 30 girdi vardır. Şekil 2.17’de gösterilen bir topolojide ağ kurulmuştur.



Şekil 2.17. LVQ Ağ Topolojisi

Yukarıdaki Şekil 2.17’ de görüldüğü gibi girdi katmanında 30 işlem elemanın hepsi de Kohonen katmanındaki elemanlara bağlıdır. Kohonen

katmanındaki her 5 ünite 1 çıktı katmanı elamanına bağlıdır. Oluşturulan ağıın parametreleri Tablo 2.10'da gösterildiği gibi belirlenmiştir.

Tablo 2.10. LVQ Ağıının Parametreleri

Parametre	Değeri
Girdi Sayısı	30
Çıktı Sayısı	5
Her Şekil için Kohonen Katman Sayısı	5
Kohonen Katmanının Eleman Sayısı	25
Ölçeklendirilme Aralığı (Sigmoid)	0-1
Başlangıç Değerleri	-0,1 ile 0,1 aralığında değerler
Öğrenme Katsayısı	0,04

2.9.3. BANKA'DA MÜŞTERİ BEKLEME SÜRESİ ÖNGÖRÜSÜ

YSA ile bir banka şubesine gelen müşterinin bekleyeceği süre tahmin edilebilmektedir. Bankaya gelen müşterinin gün, bölüm (bireysel, gişe), bekleyen kişi sayısı, kart (kartlı=1, kartsız=2), çalışan gişe sayısı değişkenlerini göz önünde bulundurarak bekleme süresini öngörmeye anlatılmaya çalışılmıştır. Bu amaçla;

Tablo 2.11. Bekleme Süresini Etkileyen Faktörler

Bekleme Süresini Etkileyen Faktörler	Değerler
Gün	Pazartesi (1), Salı (2), Çarşamba (3), Perşembe (4), Cuma (5)
Bölüm	Gişe İşlemleri (1), Bireysel (2)
Kart	Kart var ise (1), Kart yok ise (2)
Sırada Bekleyen Kişi Sayısı	Her müşterinin önünde bekleyen kişi sayısı
Çalışan Gişe Sayısı	O gün çalışan gişe sayısı

Bilet türü bireysel ve gişe işlemleri olarak 2'ye ayrılmıştır. Banka sisteminde bireysel ve gişe işlemleri kendi aralarında ayrı ayrı değerlendirilmekte ve her ikisi için ayrı bir numara üretmektedir. Ayrıca müşteriler kartlı ve kartsız olarak da ayrılmaktadır. Sisteme dâhil olan müşteri eğer kartıyla sıra almışsa sistem otomatik olarak bu müşteriye öncelik tanımaktadır. Sırada bekleyen kişi sayısı; müşteri fişi aldığı sırada önünde bekleyen kişi sayısını göstermektedir. Çıktı katmanında ise bekleme süresi vardır.

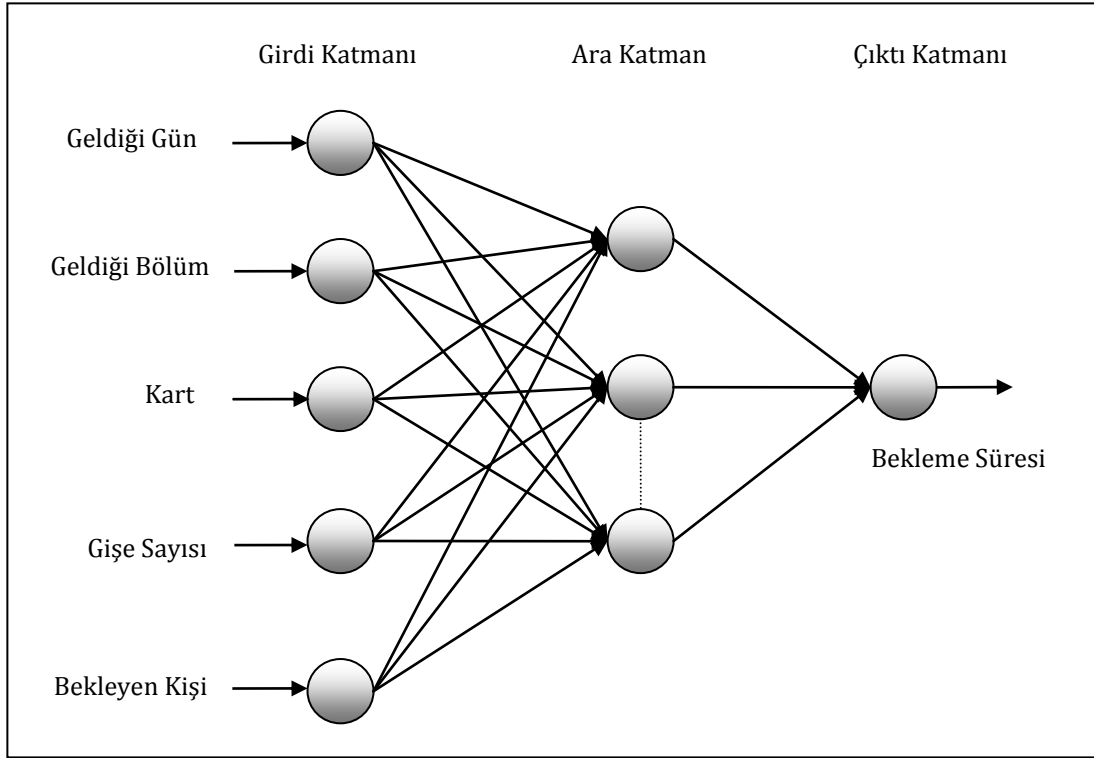
Tablo 2.12. Eğitim Seti

Gün	Bölüm	Kart	Önünde Bekleyen Kişi Sayısı	Gişe Sayısı	Bekleme Süresi (dk)
Pazartesi	Bireysel	Var	3	3	10
Pazartesi	Gişe	Var	7	3	6
Pazartesi	Gişe	Yok	13	3	12
Salı	Bireysel	Yok	22	2	45
Salı	Gişe	Yok	10	3	9
Çarşamba	Gişe	Var	9	2	14
Çarşamba	Gişe	Yok	15	2	24
Çarşamba	Gişe	Var	4	2	6
Perşembe	Bireysel	Var	0	2	1
Perşembe	Gişe	Var	2	3	3
Perşembe	Gişe	Yok	11	3	7
Perşembe	Bireysel	Yok	8	2	16
Cuma	Bireysel	Var	5	1	19
Cuma	Bireysel	Yok	15	1	44
Cuma	Gişe	Var	12	2	17
Cuma	Bireysel	Var	1	1	2
Cuma	Gişe	Var	6	2	8

Sinir ağırları nümerik bilgiyle çalışabilmektedir. Bu yüzden girdi ve çıktı seti sayısal hale dönüştürülür. Aşağıdaki Tablo 2.13’de verilerin dönüştürülmüş hali gösterilmiştir.

Tablo 2.13. Eğitim Setinin Nümerik Gösterimi

Gün	Bölüm	Kart	Önünde Bekleyen Kişi Sayısı	Gişe Sayısı	Bekleme Süresi (dk)
1	2	1	3	3	10
1	1	1	7	3	6
1	1	2	13	3	12
2	2	2	22	2	45
2	1	2	10	3	9
3	1	1	9	2	14
3	1	2	15	2	24
3	1	1	4	2	6
4	2	1	0	2	1
4	1	1	2	3	3
4	1	2	11	3	7
4	2	2	8	2	16
5	2	1	5	1	19
5	2	2	15	1	44
5	1	1	12	2	17
5	2	1	1	1	2
5	1	1	6	2	8



Şekil 2.18. Kuyruk Modeli için Çok Katmanlı Ağın Gösterimi

2.9.4. GÜNLÜK SICAKLIK TAHMİNİ

Bu çalışmada kullanılan veriler Ankara Meteoroloji Müdürlüğünden Adana iline ait günlük veriler alınmıştır. Yöntemin uygulanması “Matlab 7.10” bilgisayar programının yapay sinir ağları modülü ile gerçekleştirilmiştir. MATLAB üzerinde kurulan birçok farklı geri yayılım modeli üzerinde yapılan çok sayıda denemeler sonucunda en elverişli model seçilmiştir.

Çalışmada kullanılan veri setleri için değişik gizli katman sayıları (1-10 arasında) seçilmiş, değişik nöron sayıları (1-20 arasında) ile modeller kurulmuş ve iterasyon sayısı 1000 seçilerek eğitim gerçekleştirilmiştir. Daha sonra test için ayrılan verilerle kurulan tüm modeller test edilmiştir. Test işlemi sonucunda bulunan tahmin değerleri, gerçek değerlerle karşılaştırılarak, değişik mimarilere sahip yapay sinir ağı modellerinin tahmin doğrulukları değerlendirilmiştir.

Tablo 2.14. YSA Girdi Parametreleri

Parametreler	Özellikleri
Ağ Tipi	İleri Beslemeli Geri Yayılım
Eğitim Fonksiyonu	Trainlm
Öğrenme Fonksiyonu	Learngdm
Performans Fonksiyonu	MSE, SSE
Tabaka Sayısı	1 ile 10 arasında
Nöron Sayısı	1 ile 20 arasında
Transfer veya Aktivasyon Fonksiyonu	Tansig, Logsig

Sinir ağında kullanılacak verilerin özellikleri aşağıdaki Tablo 2.15’de verilmiştir.

Tablo 2.15. Modelde Kullanılan Parametreler ve Özellikleri

Parametre	Birim	Değişken Tipi	Ortalama	Standart Sapma	Maks.	Min.
Günlük Ortalama Rüzgar	m\sec	Girdi	1,2109157	0,7448406	6	0
Günlük Ortalama Nem	%	Girdi	67,856337	13,801702	97,7	13
Günlük Toplam Güneşlenme Süresi	Saat	Girdi	7,2931174	3,5543929	13,3	0
Günlük Açık Yüzey Buharlaşması	mm	Girdi	4,4760215	2,8187908	42,9	0
Toplam Global Güneş Radyasyonu	Cal cm ²	Girdi	368,41947	158,34234	726,15	6,45
Günlük Maks. Sıcaklık	°C	Girdi	25,509801	7,812715	43,8	6,2
Günlük Min. Sıcaklık	°C	Girdi	14,500712	7,3052991	28,7	-3,2
Basınç	hPa	Girdi	3,6424812	2,892875	10	0
Günlük Ortalama Bulutluluk	-	Girdi	1009,9814	6,0923453	1031,9	989
Günlük Ortalama Sıcaklık	°C	Çıktı	19,388614	7,4752638	35	2

Aşağıdaki Tablo 2.16’da Meteoroloji Genel Müdürlüğünden alınan günlük verinin bir kısmı verilmiştir.

Tablo 2.16. Hava Sıcaklığı Eğitim Verileri

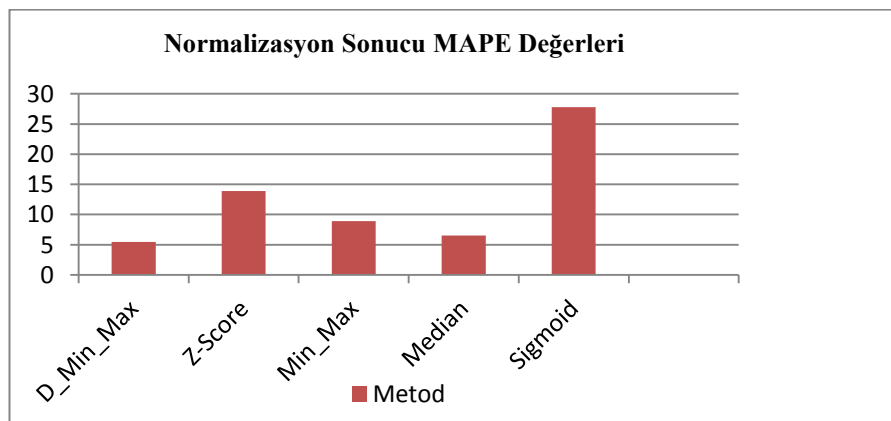
Günlük Ortalama Rüzgâr	Günlük Ortalama Nem	Günlük Toplam Güneşlenme Süresi	Günlük Açık Yüzey Buharlaştırma	Toplam Global Güneş Radyasyonu	Günlük Maksimum Sıcaklık	Günlük Minimum Sıcaklık	Günlük Ortalama Bulutluluk	Basınç	Günlük Ortalama Sıcaklık
67,3	11,5	5	497,75	28	17,9	3	1014,3	21,6	1,5
63,7	6,9	6	565,65	30	18,6	6,3	1012	23	1,9
64,7	11,8	7	353,8	30	18,7	2	1009,6	23,2	1
69,3	1,4	8,2	436,35	27	19,3	10	1009	22	1,9
70,7	5,7	5	295,2	28,2	18,1	6	1007,9	22,6	1,7
79,7	1,5	5	506,4	24,3	18,7	7,3	1003,5	20,2	1,2
63,7	8,4	1	577,4	26	17,2	4,7	1002,8	20,6	3,8
61	10,5	4,8	623,1	26,7	17	2,7	1011,7	20,4	2,3
61,7	11,2	6	619,2	27,8	15,2	3	1013,7	20,4	1,7
61,7	11,7	7	614,4	30,4	15,3	2,7	1011,2	22,4	1,4
62	12,1	7,6	552,3	28,8	17,1	1	1008,8	22,6	1,8
33,7	9,3	6,6	619,8	26,8	17	6	1008,5	22,2	2,4
32,7	12	9	603,2	28	16	3,3	1011,4	21,6	2
47,7	11,4	8,2	565,4	28	15,6	1,3	1011,4	20,5	1,6
55	9,8	7,3	339,9	28,5	14	4	1011,7	21,7	1,1
48	2,9	6,8	602,1	28,2	14,6	5,3	1010,2	21	1,1
66	11,9	6	480,3	27,4	15,3	1,7	1009,2	21,2	2,7
59,7	9,8	8	518,7	25,3	16	3,3	1008,2	19,2	2,5
63,3	10,4	6,1	334,8	26,2	14,4	5	1008,2	19,2	2,7
58,3	2,8	7,6	429,75	23	14	7,3	1007,3	17,6	3,2
61	8	4,9	524,4	23,5	12,4	4,3	1010,9	17,2	0,9
57,3	9,9	3,6	562	25,2	13,9	5	1012,2	19,2	2,4
60	11,1	6	562,8	26	14,5	4,3	1012,1	20,2	2,1
55,3	11,3	6,5	595,8	28,8	14,4	2,7	1012,9	21,7	2,2

Bu çalışmada amacımız ağın farklı normalizasyon teknikleriyle eğitilerek sistem performansı ve doğruluk derecesini tartışmaktır. Yapay sinir ağlarının geri yayımlı algoritması uygulanmıştır. Aşağıda yapay sinir ağları çalıştırılarak elde edilen ve çeşitli modellerin MAPE ve MSE sonuçları Tablo 2.17’de verilmiştir. Ağ eğitimi farklı aktivasyon fonksiyonu, nöron sayısı ve gizli katman sayıları değiştirilerek performansı ölçülmüştür.

Tablo 2.17. En İyi İstatiksel Sonuçlar

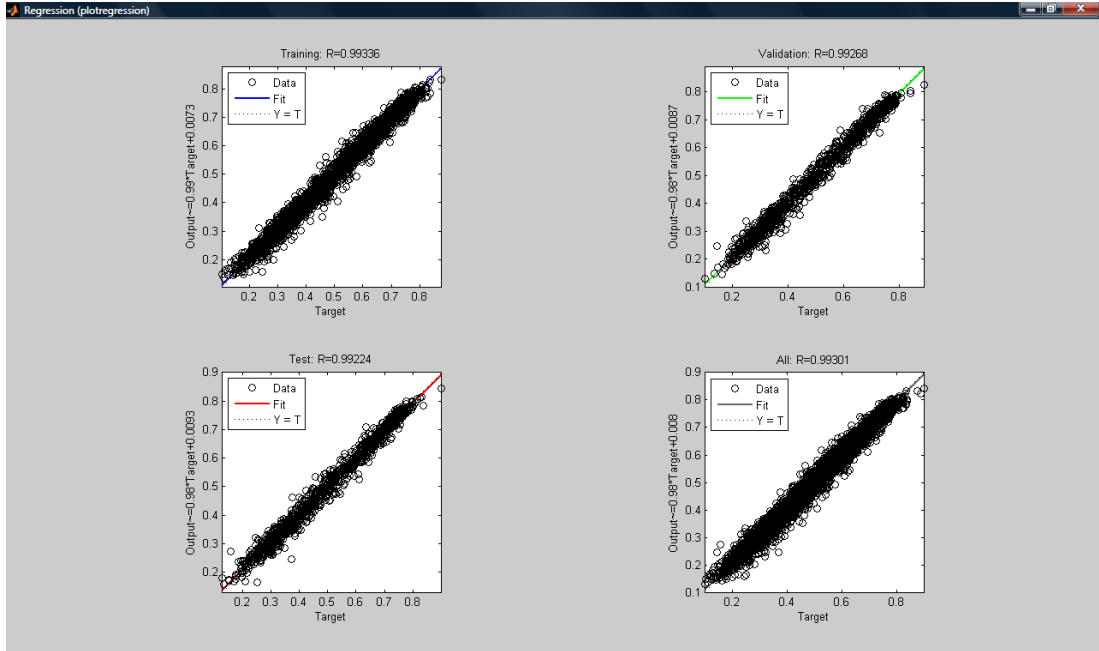
İstatistikî Sonuçlar	TANSİG(10,1,9)		TANSİG(10,1,2)		LOGSİG(10,1,9)		LOGSİG(10,1,2)	
	MAPE(%)	MSE	MAPE	MSE	MAPE	MSE	MAPE	MSE
D-Min-Maks	5,48	0,001	5,48	0,001	5,48	0,001	5,54	0,001
Z-Değeri	13,92	0,027	16,56	0,028	17,5	0,029	16,67	0,026
Min-Maks	9,07	0,002	8,92	0,001	8,93	0,001	9,06	0,001
Median	6,51	0,004	6,55	0,004	6,54	0,004	6,55	0,004
Sigmoid	28,63	0,194	28,38	0,189	28,05	0,187	27,78	0,184

Yapılan denemeler sonucunda hava tahmini için en iyi performansı veren D-Min-Maks normalizasyon yöntemidir. Şekil 2.19 de farklı normalizasyon yöntemlerinin en iyi MAPE değerleri verilmiştir.



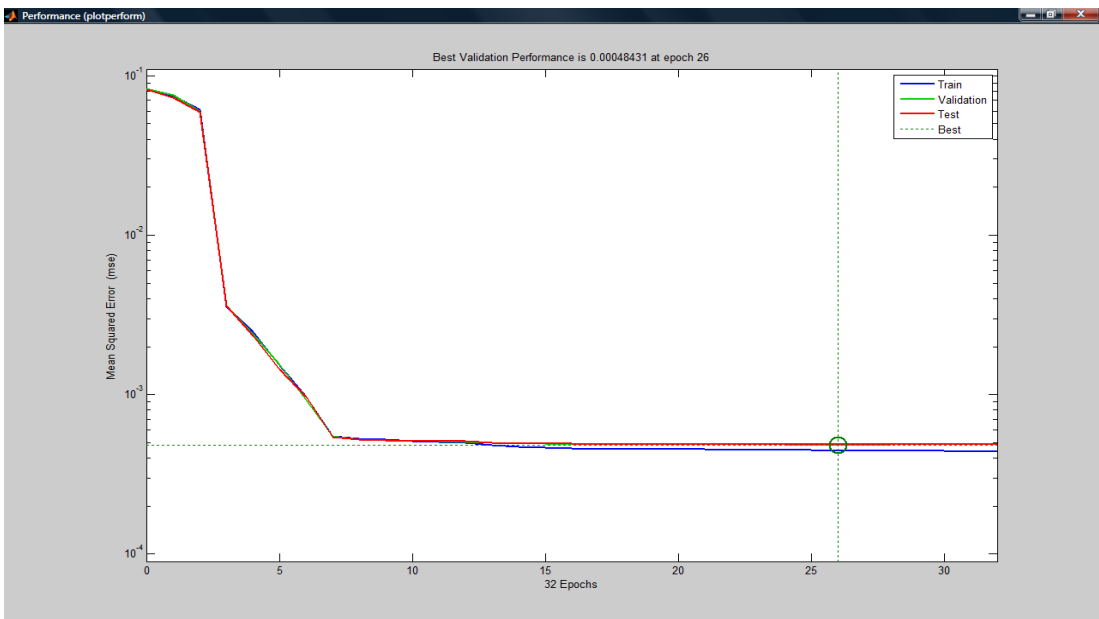
Şekil 2.19. Yöntem Sonuçlarının MAPE(%) Karşılaştırmaları

Şekil 2.20’de YSA modelinin regresyon analizi verilmiş olup, sonuç oldukça başarılıdır.

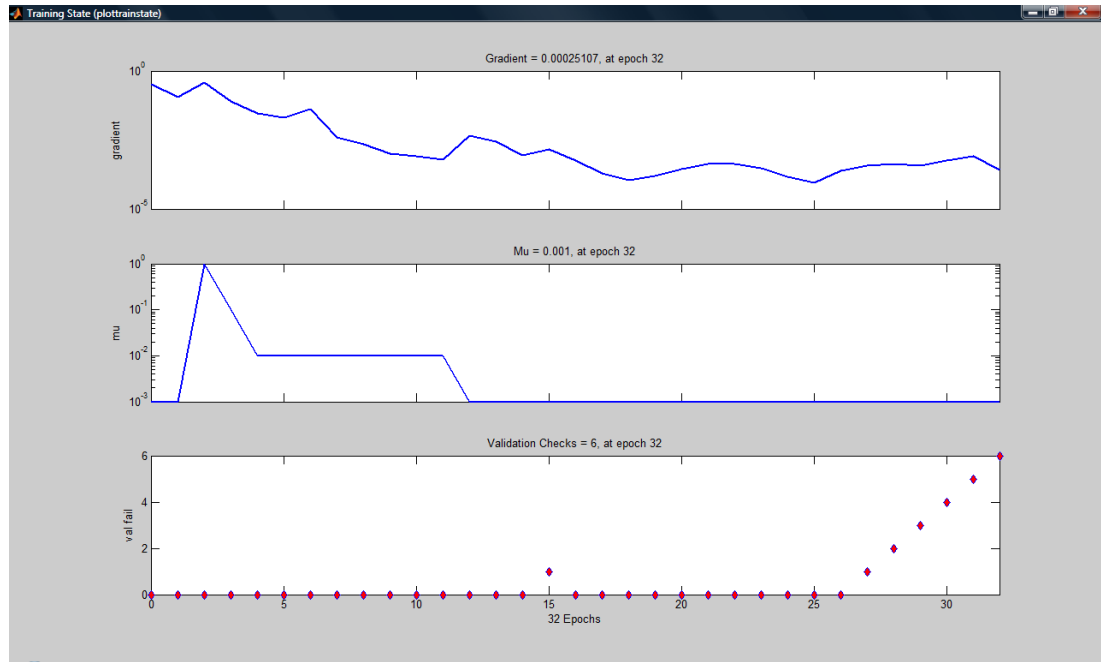


Şekil 2.20. D-Min-Maks Normalizasyon Yöntemi için Regresyon Analizi

MATLAB programıyla gerçekleştirilen eğitim, geçerlilik ve test sonuçlarının performans grafiği Şekil 2.21’de gösterilmiştir.



Şekil 2.21. D-Min-Maks Normalizasyon Yöntemi için Performans Eğrisi



Şekil 2.22. D-Min-Maks Normalizasyon Yöntemi için Eğitim Durum

ÜÇÜNCÜ BÖLÜM

HASTALARIN KUYRUKTA BEKLEME SÜRESİ TAHMİNİ

3.1. PROBLEMİN TANIMLANMASI

Uygulama Türkiye’de bir Devlet Hastanesinde hastaların kuyrukta bekleme süresini, yapay sinir ağı modelleriyle doğru ve etkin bir şekilde tahmin yapılması amaçlanmıştır. Bekleme süresini tam olarak tahmin etmek planlama açısından çok önemlidir. Örneğin hastane yönetiminin bu bekleme sürelerine göre hangi gün ve saatlerde kaç doktor çalıştıracağını belirlemesine yardımcı olur. Dolayısıyla literatürde öngöründe bulunmak için kullanılan birçok metot vardır. Bu çalışmada yapay sinir ağların öğrenme, ilişkilendirme ve genelleme yapabilme, hızlı sonuç sağlama ve kolay uygulanabilme gibi özellikleri kullanılarak bekleme süresi tahmini yapılmaya çalışılmıştır. Burada hastanın geldiği gün, bölüm, doktor sayısı, saat ile bekleme süresi arasında bir YSA modeli oluşturulmaya çalışılmıştır.

Bu çalışmada bekleme süresi tahmini için veri setine farklı normalizasyon (ölçeklendirme) tekniği, ÇKA yapısı ve bu ağı eğitmek için farklı öğrenme algoritmaları kullanılmıştır. Bu farklı normalizasyon teknikleri ve öğrenme algoritmasının kullanılmasındaki amaç, eğitim zamanını kısaltmak ve performansı arttırmak hedeflenmiştir.

Bu farklı normalizasyon teknikleri ve öğrenme algoritmaları, Çok Katmanlı Algılayıcılar ile eğitme işlemi yani girdi ve çıktı setinin ağa uygulanmasından elde edilen bekleme süresiyle gerçek bekleme süresi değeri arasındaki farkın sıfıra düşürülmeye çalışılmasıdır. Elde edilen hata değerine göre ağların ağırlıkları, farklı öğrenme algoritmaları ve ölçeklendirilmesiyle düşürülmeye çalışılmıştır.

Yapılan araştırma ve incelemeler neticesinde bekleme süresine etki eden faktörler belirlenmiş ve aşağıda Tablo 3.1’de verilmiştir.

Tablo 3.1. Bekleme Süresini Etkileyen Faktörler

Bekleme Süresini Etkileyen Faktörler	Değerler
Gün	Pazartesi (1), Salı (2), Çarşamba (3), Perşembe (4), Cuma (5)
Bölüm	Çocuk Doktoru (1), Dahiliye (2), Diş (3), Göz (4),
Randevu	Randevulu ise (1), Randevusuz ise (2)
Hastanın Geldiği Saat	Hastanın fiş aldığı saat
Çalışan Doktor Sayısı	O gün çalışan doktor sayısı
Bekleme Süresi	Hastanın bekleme süresi

Tablo 3.2. Modelde Kullanılan Parametreler

Parametre	Değişken Tipi
Gün	Girdi
Bölüm	Girdi
Randevu	Girdi
Geliş Saati	Girdi
Doktor Sayısı	Girdi
Bekleme Süresi	Çıktı

3.2. ARAŞTIRMANIN AMACI VE YÖNTEMİ

Bu çalışmada amacımız veri setine farklı öğrenme algoritmaları ve normalizasyon işlemi uygulayarak sistem performansı ve doğruluk derecesini tartışmaktır. Farklı normalizasyon teknikleri kullanarak ağ eğitilmiş ve bunların sonuçları karşılaştırılmıştır. Beş farklı normalizasyon işlemi sonucunda ağa gönderilen eğitim ve test setinin üzerinde farklı sonuçlar

bulunmuştur. Çalışmada kullandığımız her bir veri setinin birimleri farklıdır. Normalizasyon işlemleri bir nevi veri setini aynı birimlere getirmek için kullanılıp ve eğitim, geçerlilik ve test seti üzerinde sistem performansını iyileştirebilir. Alyuda NeuroIntelligence 2.2 (577) üzerinde kurulan birçok farklı model denenmiş ve bunun sonucunda en başarılı model seçilmiştir.

3.2.1. Çalışmanın Yöntemi

Çalışmada öncelikle, verilerin yapısına uygun yöntemlerin belirlenebilmesi amacıyla verilerin zaman serisi özellikleri analiz edilerek seriyi etkileyen temel bileşenler incelenmiştir. Veri bileşenlerinin incelenmesinden sonra, eğitim verileriyle test verileri yapay sinir ağ modelinde farklı normalizasyon teknikleriyle bekleme süresine yönelik tahminler yapılmıştır. Farklı normalizasyon teknikleriyle modelin ürettiği tahmin değerleri, gerçekleşmiş olan bekleme süresiyle karşılaştırılarak hangi normalizasyon tekniğinin daha gerçekçi sonuçlar verdiği belirlenmiştir. Uygulanan yöntemlerin tahmin doğruluklarını ve başarısını test etmek için Ortalama Mutlak Yüzde Hata (MAPE) ve Hata Kareler Ortalaması (MSE) istatistiği yardımıyla değerlendirilmiştir. MAPE istatistiğinin tahmin hatalarını yüzde olarak ifade etmesi nedeniyle tek başına bir anlamının olması, diğer kriterlere göre üstünlüğü olarak kabul edilmektedir (Çuhadar, Güngör ve Göksu, 2009).

Tahmin hatasıyla tahmin doğruluğu ters orantılıdır, yani tahmin hatası küçüldükçe modelin doğruluk derecesi artmaktadır. Bu nedenle, farklı tahmin modellerini karşılaştırırken en küçük hata değerine sahip model en iyi model olarak seçilmektedir (Asilkan ve Irmak 2009). MAPE ve MSE istatistiğinin matematiksel ifadesi aşağıda verilmiştir.

$$MAPE = \frac{\sum_{i=1}^n \frac{|y_i - \hat{y}_i|}{y_i}}{n} 100(\%) \quad (32)$$

$$MSE = \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n} \quad (33)$$

Bu formüllerde;

$\hat{y}_i$ = i. Dönem için hesaplanan tahmini değer,

y_i = i. Dönemde gerçekleşen değer .

n = Veri sayısı,

3.2.2. Girdi ve Çıktıların Nümerik Gösterimi

Ağın girdi setini anlayabilmesi için girdilerin nümerik gösterilmesi gerekmektedir. Bu ise her zaman kolay olmamakta ve problem tasarımcısını zor durumda bırakmaktadır. Girdi ve çıktılar nümerik hale dönüştürülmezse, çıktı değerleriyle beklenen değerler arasındaki hatayı bulmak mümkün olmaz. Aşağıdaki Tablo 3.3'te kullanılan verilerin bir kısmı örnek olarak verilmiştir.

Tablo 3.3. Bekleme Süresine Etki Eden Faktörler ve Değerleri

Gün	Bölüm	Randevu	Doktor Sayısı	Geliş Saati	Bekleme Süresi
Pazartesi	Çocuk Bölümü	Hayır	2	10:55:23	01:57:25
Çarşamba	Çocuk Bölümü	Hayır	2	11:35:18	00:06:43
Çarşamba	Çocuk Bölümü	Evet	2	13:20:08	00:02:46
Çarşamba	Çocuk Bölümü	Hayır	2	13:57:36	00:06:46
Çarşamba	Çocuk Bölümü	Hayır	2	14:23:31	00:17:45
Çarşamba	Çocuk Bölümü	Hayır	2	09:10:29	00:14:25
Çarşamba	Çocuk Bölümü	Hayır	2	09:27:06	00:10:43
Çarşamba	Çocuk Bölümü	Hayır	2	09:35:53	00:05:07
Çarşamba	Çocuk Bölümü	Hayır	2	09:12:32	00:17:13
Çarşamba	Çocuk Bölümü	Hayır	2	09:25:47	00:07:49
Çarşamba	Çocuk Bölümü	Hayır	2	09:26:08	00:06:37
Çarşamba	Çocuk Bölümü	Hayır	2	09:44:29	00:03:57
Perşembe	Çocuk Bölümü	Hayır	2	09:52:14	00:03:35
Perşembe	Çocuk Bölümü	Hayır	2	10:05:48	00:02:43
Salı	Dahiliye Bölümü	Evet	1	09:31:38	00:33:00
Salı	Dahiliye Bölümü	Hayır	1	09:36:30	00:05:58
Salı	Dahiliye Bölümü	Hayır	1	09:44:04	00:02:12
Salı	Dahiliye Bölümü	Evet	1	09:52:14	00:44:51
Salı	Dahiliye Bölümü	Hayır	1	09:56:34	00:31:19
Salı	Dahiliye Bölümü	Evet	1	09:58:13	00:05:30
Çarşamba	Dahiliye Bölümü	Hayır	1	13:05:53	00:22:05
Çarşamba	Dahiliye Bölümü	Evet	1	13:19:30	00:10:39
Çarşamba	Diş Bölümü	Evet	2	10:43:20	00:25:29
Çarşamba	Diş Bölümü	Hayır	2	13:05:15	00:33:08
Çarşamba	Diş Bölümü	Hayır	2	13:22:29	00:32:58
Çarşamba	Diş Bölümü	Evet	2	13:23:49	00:42:06
Çarşamba	Diş Bölümü	Hayır	2	14:40:59	00:09:18
Çarşamba	Diş Bölümü	Hayır	2	08:25:13	01:59:52
Çarşamba	Diş Bölümü	Hayır	2	08:27:58	01:57:42
Çarşamba	Diş Bölümü	Hayır	2	08:32:33	01:37:42
Çarşamba	Diş Bölümü	Hayır	2	08:34:22	01:53:57
Cuma	Göz Bölümü	Evet	2	13:51:04	00:07:19
Cuma	Göz Bölümü	Hayır	2	07:49:17	02:22:57
Cuma	Göz Bölümü	Hayır	2	08:13:35	01:05:39
Cuma	Göz Bölümü	Hayır	2	08:15:00	01:23:54
Cuma	Göz Bölümü	Hayır	2	08:15:51	01:23:48
Cuma	Göz Bölümü	Hayır	2	08:18:15	02:10:23
Cuma	Göz Bölümü	Hayır	2	08:46:36	01:43:01
Çarşamba	Göz Bölümü	Evet	2	08:45:17	01:46:03

Görüldüğü gibi bazı değerler tamamen alfanümerik (rakam ve harf içeren) değerlerdir. Ağı eğitmek için bu değerleri nümerik hale dönüştürmek zorundayız. Bu değişik şekillerde yapılabilir. Örneğin evet için 0, hayır için 1

değeri gösterilebilir. Ya da evet için 1, hayır için 2 değeri kullanılabilir. Seçilecek yöntem başarıyı etkileyecektir. O zaman veri setinin nümerik gösterimi Tablo 3.4'te verilmiştir.

Tablo 3.4. Verilerin Nümerik Değer ile Gösterimi

Gün	Bölüm	Randevu	Doktor Sayısı	Geliş Saati	Bekleme Süresi
1	1	2	2	10,92305555	0,035000000
3	1	2	2	11,58833333	0,111944444
3	1	1	2	13,33555556	0,046111111
3	1	2	2	13,96	0,112777778
3	1	2	2	14,39194444	0,295833333
3	1	2	2	9,174722222	0,240277778
3	1	2	2	9,451666667	0,178611111
3	1	2	2	9,598055556	0,085277778
3	1	2	2	10,3625	0,065277778
3	1	2	2	9,429722222	0,130277778
3	1	2	2	9,435555556	0,110277778
3	1	2	2	9,741388889	0,065833333
3	1	2	2	10,25638889	0,110833333
4	1	2	2	9,870555556	0,059722222
4	1	2	2	10,09666667	0,045277778
2	2	1	1	9,527222222	0,55
2	2	2	1	9,608333333	0,099444444
2	2	2	1	9,734444444	0,036666667
2	2	1	1	9,870555556	0,7475
2	2	2	1	9,942777778	0,521944444
2	2	1	1	9,970277778	0,091666667
3	2	1	1	13,325	0,1775
3	2	1	1	13,33972222	0,216388889
3	3	2	2	13,0875	0,552222222
3	3	2	2	13,37472222	0,549444444
3	3	1	2	13,39694444	0,701666667
3	3	2	2	14,68305556	0,155
3	3	2	2	8,420277778	1,997777778
3	3	2	2	8,466111111	1,961666667
3	3	2	2	8,5425	1,628333333
3	3	2	2	8,572777778	1,899166667
3	3	2	2	9,486666667	1,007777778
5	4	1	2	13,85111111	0,121944444
5	4	2	2	7,821388889	2,3825
5	4	2	2	8,226388889	1,094166667
5	4	2	2	8,25	1,398333333
5	4	2	2	8,264166667	1,396666667
5	4	2	2	8,304166667	2,173055556
5	4	2	2	8,776666667	1,716944444
3	4	1	2	8,754722222	1,7675

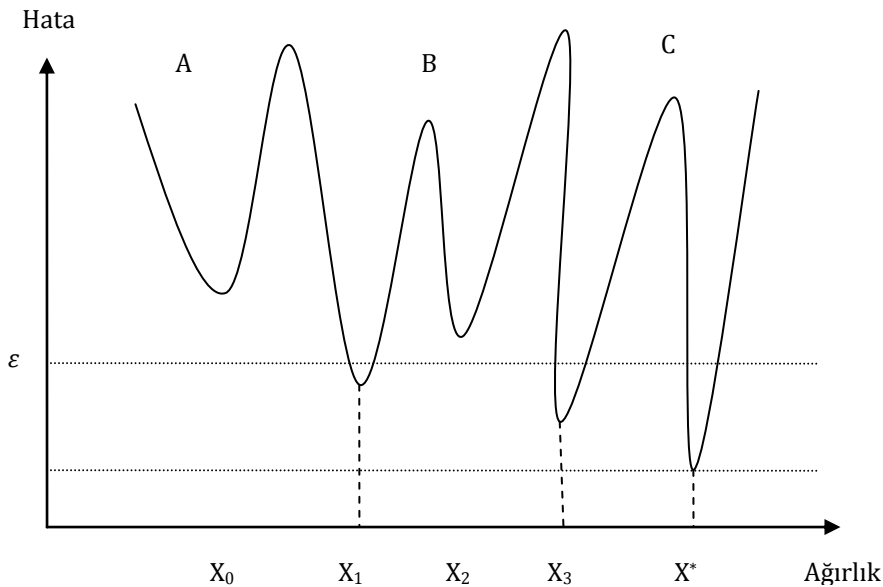
Yukarıdaki Tablo 4.4’de görüldüğü gibi günler için 1, 2, 3, 4 ve 5 rakamı, bölümler için 1, 2, 3 ve 4 rakamı, randevu için 1 ve 2, geliş saati ve bekleme süresi 24’ lük dilim üzerinden ifade edilmiştir.

3.2.3. Yapay Sinir Ağı Yapısının Seçimi

Uygulanan probleme bağlı olarak uygun YSA mimarisi seçilmelidir. Çok katmanlı ağların performansları yüksektir. Bizim çalışmamızda da ÇKA yapısını kullanacağız.

3.2.4. Başlangıç Değerlerinin Atanması

Ağın performansını yakından ilgilendirmektedir. Başlangıç değerleri belirli aralıkta atanmaktadır. Bu değerlerin büyük veya küçük olması ağın performansını olumlu veya olumsuz etkilemektedir.



Şekil 3.1. Başlangıç Değerlerinin Çok Katmanlı Ağlara Etkisi

Şekil 3.1’de görüldüğü gibi sinir ağı öğrenmeye A noktasından başlarsa yerel bir çözüme takılabilme olasılığı var iken C noktasından başlarsa en iyi çözümü (X^*) bulabilme olasılığı vardır.

3.2.5. Öğrenme Katsayısı ve Momentum Katsayılarının Belirlenmesi

Öğrenme ve momentum katsayılarının belirlenmesi başlangıç değerlerinin belirlenmesi kadar önemlidir. Öğrenme katsayısı ağırlıkların değişim miktarını belirlemektedir. Momentum katsayısıysa bir önceki iterasyondaki değişimin belirli bir oranını yeni değişim miktarına eklenmesi durumudur. Ağın yerel çözümlere takılmasını engellemek için kullanılır (Öztemel, 2003).

3.2.6. Normalizasyon (Ölçeklendirme) İşlemi

Sinir ağların doğrusal olmama özelliğini anlamlı kılan yaklaşım, verilerin normalizasyon işlemine tabi tutulmasıdır. Veri setine göre seçilen normalizasyon tekniklerinden biri, ağın performansını doğrudan etkileyecektir. Çünkü normalizasyon, giriş verilerinin transfer edilirken fonksiyonun aktif olan bölgesinden aktarılmasını sağlar. Böylelikle aşırı değerlendirilmiş kümülatif toplamalarının oluşturacağı olumsuzlukların engellenmesi sağlanır. Genellikle veriler $[0, 1]$ veya $[-1, +1]$ aralıklarından birine normalize edilmesi önerilir. Ölçekleme, verinin geçerli eksen sisteminde sıkıştırılması anlamı taşıdığından, aşırı salınımlar içeren problemlerin sinir ağı modellerini olumsuz yönde etkileyebilir. Bu olumsuzluk, kullanılacak öğrenme fonksiyonunu da başarısız kılabilir (Sağıroğlu, Beşdok ve Erler, 2003).

Yapay sinir ağlarında, ağ giriş ve çıkışlarına belirli ön işlem adımları uygulayarak yapay sinir ağına sunulan verilerin eğitimi daha verimli hale getirilebilir. Ağ giriş işlem fonksiyonları, ağ kullanımını daha iyi bir forma dönüştürür. Normalleştirme işlemi ham verilere uygulanarak ve bu verilerin eğitim için uygun veri setinin hazırlanmasına etkisi vardır. Yapay sinir ağların eğitimi normalizasyonsuz çok yavaş olabilir. Normalleştirme işlemlerinde farklı teknikler kullanılır. Literatürde birçok istatistiksel normalizasyon çeşidi

vardır. Bunlar; Minimum-Maksimum kuralı, Medyan, Sigmoid ve Z-Değeri gibi kurallar olabilmektedir (Jayalakshmi ve Santhakumaran, 2011).

ÇKA modelin girdi ve çıktıların normalizasyonu, ağı performansı yakından etkilemektedir. Çünkü normalleştirme, veri setindeki değerlerin dağılımını düzenli hale getirmektedir. YSA girdileri arasında aşırı büyük veya küçük değerler görülebilir. Bunlar yanlışlıkla girdi setine girmiş olabilir. Net girdiler hesaplanırken bu değerler aşırı büyük veya küçük değerlerin doğmasına neden olarak ağı yanlış yönlendirebilirler. Bütün girdilerin belirli aralıkta (çoğunlukla 0-1 aralığında) normalize edilmesi hem farklı ortamlardan gelen bilgilerin aynı ölçek üzerine indirgenmesine hem de yanlış girilen çok büyük ve küçük şekildeki değerlerin etkisinin ortadan kalkmasına neden olur. Bazı araştırmacılar kendi problemlerine özgü normalizasyon yöntemleri geliştirmektedir. Her problem için farklı bir normalizasyon yöntemi kullanılabilir. ÇKA ağı tasarımcıları ellerindeki verileri normalize edecek bir yaklaşımı kendileri belirleyebilir. Bu konuda bir standart koymak doğru olmaz (Öztemel, 2003).

Aşağıda bu normalizasyon teknikleri incelenmiş olup, veri setlerin bir kısmı eğitim, geçerlilik ve bir kısmı test serisi olarak yapay sinir ağlarında farklı öğrenme aktivasyonları, gizli katman ve nöron sayıları değiştirilerek incelenmiştir. Test sonucunda elde edilen verilerle gerçek değer karşılaştırılmıştır. Sonuçlar MAPE ve MSE yardımıyla değerlendirilmiştir.

3.2.6.1. İstatiksel veya Z-Değeri Normalizasyonu

Aynı veri seti üzerindeki bazı değerlerin 0'dan küçük bazılarının ise daha büyük değerlere sahip olduğu, veriler arasındaki bu uzaklıkların özellikle uç verilerin sonuçlar üzerinde daha etkin olacağını göstermektedir. Verilerin normalleştirme işlemiyle eğitim giriş setindeki her bir parametrenin modelin tahmin işlemine eşit ölçüde katkıda bulunması sağlanır. Her bir değişkene karşı gelen sütunların aritmetik ortalaması ve standart sapması

bulunmuştur. Sonra da, Eşitlik 34'de belirtilen formül ile veriler normalleştirilerek veriler arasındaki uzaklıklar ortadan kaldırılır ve verilerdeki uç noktaları azaltılır (Yüksek, Bircan, Zontul ve Kaynar, 2007).

$$x' = \frac{x_i - \mu_i}{\sigma_i} \quad (34)$$

x' = Normalize edilmiş veri ,

x_i = Girdi değeri ,

μ_i = Girdi setinin ortalaması ,

σ_i = Girdi setinin standart sapması ,

3.2.6.2. Min-Maks Normalizasyonu

Verileri doğrusal olarak normalize eder. Minimum bir verinin alabileceği en düşük değerkken, maksimum verinin alabileceği en yüksek değeri ifade eder. Bir veriyi Min-Maks yöntemi ile 0 ile 1 aralığına indirmek için Eşitlik 35'teki formül kullanılır.

$$x' = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}} \quad (35)$$

x' = Normalize edilmiş veri ,

x_i = Girdi değeri ,

$x_{\min}$ = Girdi seti içerisinde yer alan en küçük sayı ,

$x_{\max}$ = Girdi seti içerisinde yer alan en büyük sayıyı ifade etmekte ,

3.2.6.3. Medyan Normalizasyonu

Bu yöntem her girdinin medyan değerini alarak her örneklem için medyan normalize yöntemini kullanır. Medyan aşırı sapmalardan etkilenmez (Jayalakshmi ve Santhakumaran, 2011).

$$x' = \frac{x_i}{\text{Median}(a_i)} \quad (36)$$

x' = Normalize edilmiş veri ,

x_i = Girdi değeri ,

a_i = Girdi setinin medyanı ,

3.2.6.4. Sigmoid Normalizasyonu

Sigmoid ölçeklendirme fonksiyonu, verileri 0 ile 1 veya -1 ile 1 arasında sınıflandırır. Bir kaç tane doğrusal olmayan sigmoid fonksiyon çeşidi vardır. Tanjant sigmoid fonksiyonu süreçleri hızlandırmada iyi bir seçim olabilir (Jayalakshmi ve Santhakumaran, 2011).

$$x' = \frac{e^{x_i} - e^{-x_i}}{e^{x_i} + e^{-x_i}} \quad (37)$$

x' = Normalize edilmiş veri ve x_i = Girdi değeri ,

3.2.6.5. D-Min-Maks Normalizasyonu

Öncelikle modelin uygulanması için aşağıda verilen denklem ile tüm veriler 0,1 ile 0,9 arasında normalize edilmiştir. Normalizasyon yapılarak veriler boyutsuz hale getirilmiş olur (Doğan, Işık ve Sandalcı, 2007).

$$x' = 0.8 * \frac{x_i - x_{\min}}{x_{\max} - x_{\min}} + 0.1 \quad (38)$$

x' = Normalize edilmiş veri ,

x_i = Girdi değeri ,

$x_{\min}$ = Girdi seti içerisinde yer alan en küçük sayı ,

$x_{\max}$ = Girdi seti içerisinde yer alan en büyük sayıyı ifade etmekte ,

3.3. ALYUDA NEUROINTELLIGENCE İLE YSA BEKLEME SÜRESİ TAHMİNİ

Yapay sinir ağları sisteminde ne kadar çok veri ağı girilirse o kadar doğru sonuç alınacağından 1966 adet veri sunulmuştur.

Tablo 3.5. Veri Setinin Sınıflandırılması

Sınıflandırma	Yüzde(%)	Veri Seti
Eğitim Seti	68	1338
Geçerlilik Seti	16	314
Test Seti	16	314
Göz ardı	0	0
Toplam	100	1966

Yapay sinir ağlarının çalışma sistemine bağlı olarak, 1338 veri eğitim, 314 geçerlilik, 314 veri test ve toplamda 1966 veri kullanılmıştır. Seçilen parametrelerin değiştirilmesiyle elde edilen tahmin değerleri için performans hesabı yapılmıştır. Veriler ve bunlara karşılık gelen sonuçlar tanıtıldıktan sonra ağı eğitilmiştir. Bu aşamada sistem verilerle sonuçlar arasında birçok bağ kurarak, verilere farklı ağırlıklar atayarak elde edilen sonuçlara yaklaşılmaya çalışmaktadır. Diğer bir deyişle çok değişkenli bu problemi öğrenmektedir. Bu çalışmada iterasyon sayısı 500 veya 1000 olarak kullanılmıştır.

Modelde kullanılan parametreler normalizasyon, katman sayısı (K.S), öğrenme katsayısı (Ö.K), momentum katsayısı (M.K) ve aktivasyon fonksiyonu (A.F) kullanılmaktadır. Aşağıdaki Tablo 3.6, 3.7 ve 3.8’de farklı öğrenme algoritmaları, normalizasyon, katman sayısı, aktivasyon fonksiyonu, momentum ve öğrenme katsayıları kullanılarak, bekleme süresi için en iyi performansı veren yöntem belirlenmiştir.

Tablo 3.6. Öğrenme Algoritmasına Göre Sistem Çıktıları 1

Öğrenme Algoritması	Normalizasyon	K.S	Ö.K	M.K	A.F	MAPE(%)	MSE
Hızlı Yayılım	D-Min-Maks	45	0,1	0,1	L	25,04	0,0036
Hızlı Yayılım	D-Min-Maks	23	0,1	0,1	L	24,16	0,0039
Hızlı Yayılım	D-Min-Maks	7	0,1	0,1	L	25,28	0,0041
Hızlı Yayılım	D-Min-Maks	23	0,1	0,1	L ve T	24,37	0,0036
Hızlı Yayılım	D-Min-Maks	23	0,1	0,1	T	24,31	0,0037
Kuasi-Newton	D-Min-Maks	20	0,1	0,1	L	29,83	0,0044
Kuasi-Newton	D-Min-Maks	23	0,1	0,1	T	25,86	0,0041
Kuasi-Newton	D-Min-Maks	16	0,1	0,1	T	29,93	0,0044
Online Geri Yayılım	D-Min-Maks	45	0,1	0,1	T	29,1	0,0061
Online Geri Yayılım	D-Min-Maks	23	0,1	0,1	T	28,44	0,0054
Online Geri Yayılım	D-Min-Maks	16	0,1	0,1	T	30,18	0,0058
Eşleştirmeli Eğilim	D-Min-Maks	19	0,1	0,1	L	28,67	0,0051
Eşleştirmeli Eğilim	D-Min-Maks	45	0,1	0,1	T	26,58	0,0045
Eşleştirmeli Eğilim	D-Min-Maks	23	0,1	0,1	T	25,86	0,0042
Eşleştirmeli Eğilim	D-Min-Maks	16	0,1	0,1	T	26,4	0,0044
Grup Geri Yayılım	D-Min-Maks	45	0,1	0,1	T	28,07	0,0055
Grup Geri Yayılım	D-Min-Maks	23	0,1	0,1	T	28,09	0,0055
Grup Geri Yayılım	D-Min-Maks	16	0,1	0,1	T	28,36	0,0056
Hızlı Yayılım	Median	45	0,1	0,1	T	308,76	0,9662
Hızlı Yayılım	Median	31	0,1	0,1	T	283,34	1,0032
Hızlı Yayılım	Median	23	0,1	0,1	T	287,59	0,9132
Hızlı Yayılım	Median	16	0,1	0,1	T	285,19	0,9353
Kuasi-Newton	Median	45	0,1	0,1	T	325,15	1,1517
Kuasi-Newton	Median	31	0,1	0,1	T	314,38	1,1532
Kuasi-Newton	Median	23	0,1	0,1	T	310,81	1,1578
Kuasi-Newton	Median	16	0,1	0,1	T	320,78	1,1635
Online Geri Yayılım	Median	45	0,1	0,1	T	288,69	1,2644
Online Geri Yayılım	Median	31	0,1	0,1	T	354,58	1,2518
Online Geri Yayılım	Median	23	0,1	0,1	T	326,88	1,2537
Online Geri Yayılım	Median	16	0,1	0,1	L	315,29	1,275
Grup Geri Yayılım	Median	45	2	0,8	L	322,71	1,1314
Grup Geri Yayılım	Median	31	2	0,8	L	324,05	1,1365
Grup Geri Yayılım	Median	23	2	0,8	L	326,89	1,1363
Grup Geri Yayılım	Median	16	0,1	0,1	L	400,43	1,6525
Grup Geri Yayılım	Median	16	2	0,8	L	324,82	1,1322

(L: Logistik Sigmoid ve H: Hiperbolik Tanjant)

Tablo 3.7. Öğrenme Algoritmasına Göre Sistem Çıktıları 2

Algoritma	Normalizasyon	K.S	Ö.K	M.K	A.F	MAPE(%)	MSE
Hızlı Yayılım	Min-Maks	45	0,1	0,1	L	473,18	0,0083
Hızlı Yayılım	Min-Maks	31	0,1	0,1	L	487,99	0,0085
Hızlı Yayılım	Min-Maks	16	0,1	0,1	L	479,27	0,0078
Hızlı Yayılım	Min-Maks	23	0,1	0,1	T	500,29	0,0079
Kuasi-Newton	Min-Maks	45	0,1	0,1	L	443,03	0,0086
Kuasi-Newton	Min-Maks	31	0,1	0,1	L	445,05	0,0089
Kuasi-Newton	Min-Maks	23	0,1	0,1	L	343,62	0,0089
Kuasi-Newton	Min-Maks	16	0,1	0,1	L	365,37	0,0089
Online Geri Yayılım	Min-Maks	45	0,1	0,1	L	423,85	0,0087
Online Geri Yayılım	Min-Maks	16	0,1	0,1	L	404,98	0,0086
Online Geri Yayılım	Min-Maks	31	0,1	0,1	L	429,74	0,0086
Online Geri Yayılım	Min-Maks	23	0,1	0,1	L	421,41	0,0086
Grup Geri Yayılım	Min-Maks	45	2	0,8	L	436,53	0,0085
Grup Geri Yayılım	Min-Maks	31	2	0,8	L	438,34	0,0085
Grup Geri Yayılım	Min-Maks	23	2	0,8	L	439,55	0,0085
Grup Geri Yayılım	Min-Maks	16	2	0,8	L	438,66	0,0085
Hızlı Yayılım	Z-Değeri	23	0,1	0,1	L	463,4	0,3981
Hızlı Yayılım	Z-Değeri	45	0,1	0,1	T	393,28	0,4071
Hızlı Yayılım	Z-Değeri	42	0,1	0,1	T	369,83	0,4033
Hızlı Yayılım	Z-Değeri	23	0,1	0,1	T	342,85	0,4084
Hızlı Yayılım	Z-Değeri	16	2	0,2	T	472,79	0,4029
Kuasi-Newton	Z-Değeri	42	2	0,2	L	692,08	0,449
Kuasi-Newton	Z-Değeri	16	2	0,2	L	815,86	0,4498
Online Geri Yayılım	Z-Değeri	45	0,1	0,1	L	528,25	0,4207
Online Geri Yayılım	Z-Değeri	42	0,1	0,1	L	451,58	0,411
Online Geri Yayılım	Z-Değeri	23	0,1	0,1	L	477,87	0,3951
Online Geri Yayılım	Z-Değeri	16	0,1	0,1	L	469,75	0,4079
Online Geri Yayılım	Z-Değeri	42	0,1	0,1	T	1321,28	0,4526
Grup Geri Yayılım	Z-Değeri	45	2	0,2	L	736,15	0,4501
Grup Geri Yayılım	Z-Değeri	42	2	0,2	L	771,45	0,4516
Grup Geri Yayılım	Z-Değeri	23	2	0,2	L	774,86	0,453
Grup Geri Yayılım	Z-Değeri	16	4	0,1	L	840,98	0,4403
Hızlı Yayılım	Sigmoid	45	0,1	0,1	L	194,5	0,0993
Hızlı Yayılım	Sigmoid	32	0,1	0,1	L	187,28	0,1003
Kuasi-Newton	Sigmoid	45	3	0,8	L	191,18	0,1036
Kuasi-Newton	Sigmoid	32	3	0,8	L	190,06	0,1026
Online Geri Yayılım	Sigmoid	45	0,1	0,1	L	188,35	0,0996
Online Geri Yayılım	Sigmoid	32	0,1	0,1	L	188,06	0,0996
Grup Geri Yayılım	Sigmoid	45	2	0,9	L	188,69	0,1007
Grup Geri Yayılım	Sigmoid	32	3	0,8	L	189,16	0,1012

Tablo 3.8. Öğrenme Algoritmasına Göre Sistem Çıktıları 3

Algoritma	Normalizasyon	K.S	Ö.K	M.K	A.F	MAPE(%)	MSE
Hızlı Yayılım	D-Min-Maks	7	0,1	0,1	L	7,79	0,0227
Hızlı Yayılım	D-Min-Maks	34	0,1	0,1	L	7,77	0,0226
Hızlı Yayılım	D-Min-Maks	34	1	0,1	L	7,86	0,0238
Hızlı Yayılım	D-Min-Maks	34	2	0,1	L	7,74	0,0234

(L: Logistik ve H: HiperbolikTanjant)

Tablo 3.6 ve 3.7’de eğitim, geçerlilik ve test setleri için dikkate alınan girdi-çıkı verileri rastgele seçilmiştir. Burada ağırlıkları ayarlamak için farklı momentum yöntemleri kullanılmıştır. Yapay sinir ağı yönteminde, yakınsama hızını arttırmak için öğrenme oranı kullanılmıştır. En uygun YSA model yapısını bulabilmek için Alyuda NeuralIntelligence programı kullanılarak, farklı YSA mimarileri denenmiştir. Bunun sonucunda en iyi MAPE değeri D-Min-Maks yöntemiyle 24,16 bulunmuştur. Sonucu dahada iyileştirmek adına veriler ağı eğitim ve test seti olarak ikiye bölünmüştür. Tablo 3.8’de ağı gönderilen 1966 verinin %85’i eğitim ve %15 test için kullanılmıştır. Bunun sonucunda çarpıcı şekilde farklı MAPE ve MSE sonuçları bulunmuştur. Yine D-Min-Maks yöntemi sonucunda bulunan MAPE değeri 7,74’tür. Bulunan sonuç farklı öğrenme ve momentum katsayısı, nöron, normalizasyon yöntemi ve katmanlar ile dahada iyileştirilebilir.

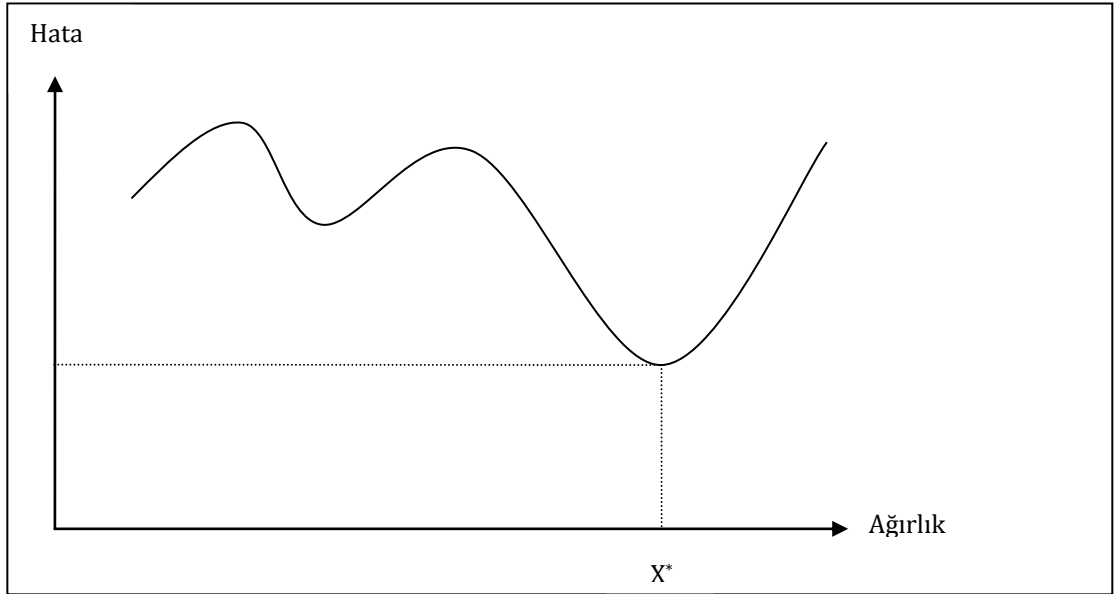
İstatiksel normalizasyon tekniği ile YSA farklı öğrenme algoritmalarıyla uygulanmıştır. Yukarıda çeşitli modellerin MAPE(%) ve MSE sonuçları tabloda gösterilmiştir. En küçük MAPE(%) ve MSE değeri seçilerek, hangi yöntemin daha iyi olduğu belirlenmeye çalışılmıştır.

Tablo 3.9. En iyi MAPE ve MSE Sonucu

Algoritma	Normalizasyon	K.S	Ö.K	M.K	A.F	MAPE(%)	MSE
Quick Propagation	D-Min-Maks	34	2	0,1	L	7,74	0,0234

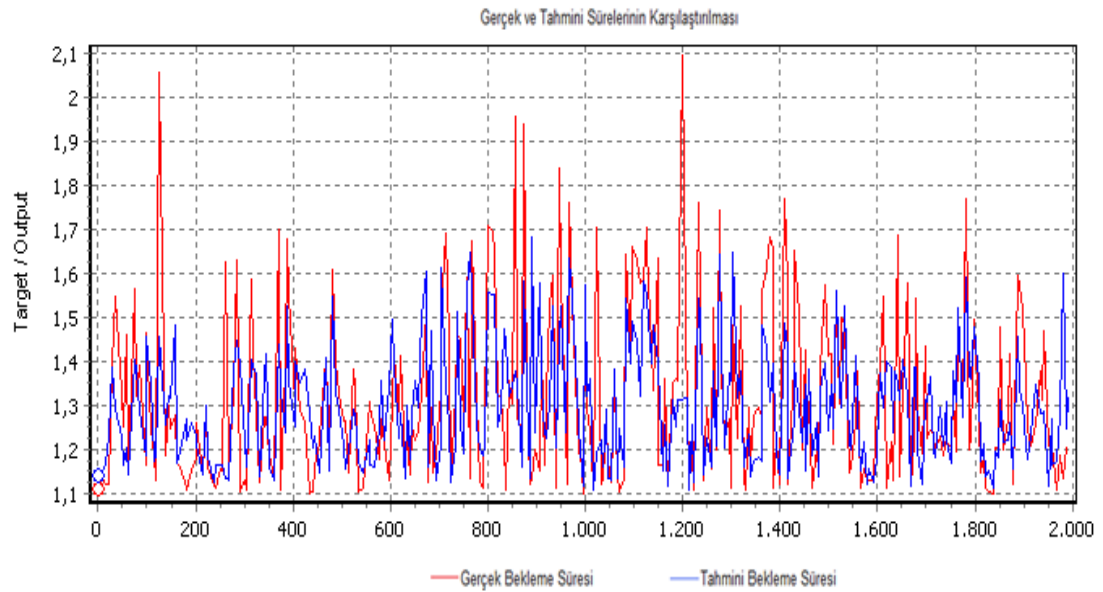
Aşağıdaki Şekil 3.2’deki X^* , hatanın en az olduğu ağırlık vektörünü göstermektedir. Problemi çözerken hatanın X^* değerine ulaşmasını isteriz.

Çoğu zaman bu değeri yakalamak mümkün değildir. Bu durum problemin en iyi çözüm noktasıdır.



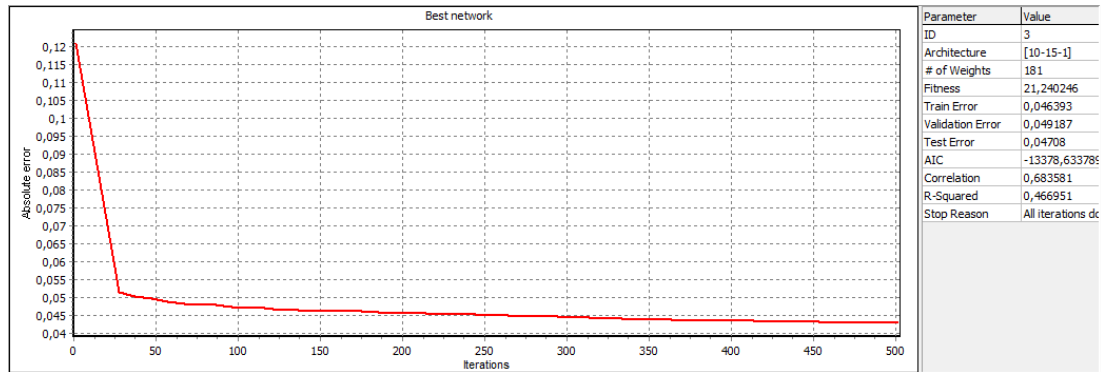
Şekil 3.2. Öğrenmenin Hata Uzayındaki Gösterimi

Ağ farklı algoritmalar ile eğitilmiş ve en iyi X^* değeri bulunmaya çalışılmıştır.

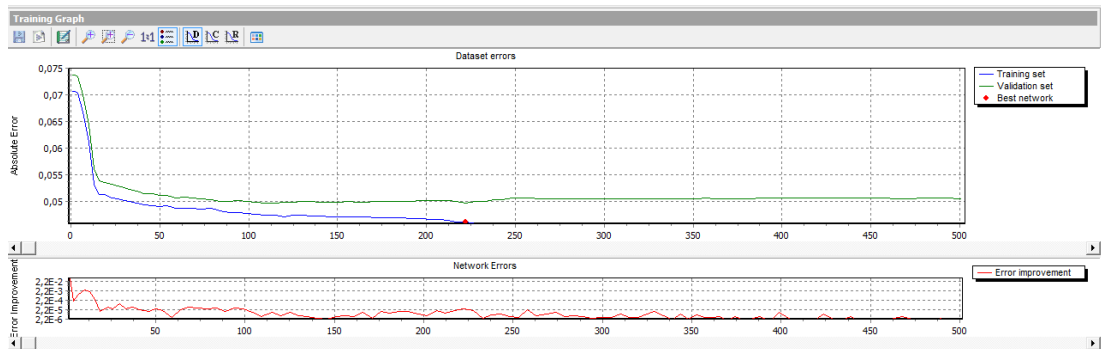


Şekil 3.3. Gerçek Bekleme Süresi ile Tahmini Sürelerin Karşılaştırılması

Şekil 3.4'te Hızlı Yayılım öğrenme algoritmasıyla sistem eğitildikten sonra Alyuda NeuralIntelligence tarafından verilen eğitim raporu ile görülmektedir.



Şekil 3.4. Ağ Eğitim Raporu



Şekil 3.5. Eğitim Grafiği

Eğitim sonucu oluşan mutlak hata, sistemin yapısı, eğitim algoritması ve iterasyon sayısı sistem çıktısı olarak Şekil 3.6'da görülmektedir.

Parameters		
	Training	Validation
Absolute error:	0,044266	0,050568
Network error:	0,016789	0
Error improvement:	0,000001	
Iteration:	501	
Training speed, iter/sec:	23,857134	
Architecture:	[10-34-1]	
Training algorithm:	Quick Propagation	
Training stop reason:	All iterations done	

Şekil 3.6. Eğitim Grafiği Parametreleri

SONUÇLAR ve ÖNERİLER

YSA'da, başarılı sonuçlara ulaşabilmek için, girdilerin doğru tanımlanması gerekmektedir. Modelde kullanılacak olan girdilerin geniş çaplı incelenmesi, kullanılacak olan modelin güvenilir sonuç vermesi için öncelikli şarttır.

Bu çalışmada; YSA öğrenme yöntemlerinden Hızlı Yayılım, Eşleştirmeli Eğilim, Kuasi-Newton, Online Geri Yayılım ve Grup Geri Yayılım yöntemleriyle farklı mimarilere, ağırlık, katsayı, katman, normalizasyon ve aktivasyon fonksiyonuna sahip modellerinin tahmin doğruluklarını karşılaştırarak, en yüksek doğruluğu sağlayan modelin belirlenmesi ve belirlenen model yardımıyla hastaların kuyrukta bekleme süresinin tahmini yapılması amaçlanmıştır. Farklı normalizasyon tekniklerinin yapay sinir ağı performansına etkisi incelenmiştir. Beş farklı teknik, farklı yöntemlerle çalıştırılarak en iyi sonuç bulunmaya çalışılmıştır. Uygulanan yöntemlerden elde edilen tahmin sonuçlarının değerlendirilmesi neticesinde, en yüksek tahmin doğruluğunu sağlayan ve gerçek değerlere en yakın sonuçları veren yöntemin, "Hızlı Yayılım" ve normalizasyon yöntemindeyse "D-Min-Maks" olduğu görülmüştür. YSA, veriler arasındaki doğrusal olmayan ilişkileri öğrenip genelleme yapabilmekte ve bu sayede daha önce hiç karşılaşmadığı sorulara kabul edilebilir bir hatayla cevap bulabilmektedirler. Bu özellikleri nedeniyle yapay sinir ağları, tahminlerde etkili bir yöntem olarak kullanılmaktadır.

Bekleme süresi tahmini için istatistiksel normalizasyon metodu, yapay sinir ağlarını kullanarak daha iyi performans sergilerler. Sonuç olarak, uygun ağ yapısı ve yeterli sayıda veri kullanıldığında, zaman serilerinin tahmininde yapay sinir ağları istatistiksel yöntemlere alternatif bir yöntem olarak kullanılabilir. Bu eldeki bekleme süresi verileri için uygulanan yöntemlerden elde edilen tahmin sonuçlarının değerlendirilmesi neticesinde, en yüksek tahmin doğruluğunu sağlayan ve gerçek değerlere en yakın sonuçları veren

yöntemin, “D-Min-Maks” normalizasyon olduğu görülmüştür. Jayalakshmi ve Santhakumaran yaptığı bir çalışmada diyabet hastalarını tahmin etmede kullanılan verilerde ise farklı bir normalizasyon tekniği daha iyi sonuç vermiştir. Her problem için farklı bir normalizasyon yöntemi kullanılabilir. Tasarımcılar ellerindeki verileri normalize edecek bir yaklaşımı kendileri belirleyebilir (Öztemel, 2003).

Ayrıca yapay sinir ağları ile zaman serisi tahmin yöntemlerinin birleştirildiği melez modellerin tahmin performansları araştırılabilir. Ülkemizde gerek yapay sinir ağları, gerekse melez yaklaşımlarla kuyrukta bekleme sürelerinin serilerinin modellenmesi ve tahminiyle ilgili çalışmaların sınırlı sayıda olduğu dikkate alınır, önerilen çalışmaların literatüre ve hastane sektöründeki uygulamacılar ile karar verme konumunda olan yöneticilerin geleceğe yönelik planlama çalışmalarına önemli katkılar sağlayacağı söylenebilir.

KAYNAKÇA

AHMAD, İftikhar, ANSARİ, M, A., MOHSİN, Sajjad; “Performance Comparison between Backpropagation Algorithms Applied to Intrusion Detection in Computer Network Systems”, **9th WSEAS International Conference on NEURAL NETWORKS (NN’08)**, Bulgaria, 2008.

ALAVALA, Chennakesava, R.; **Fuzzy Logic And Neural Networks: Basic Concepts & Applications**, New Age International Publishers, 1.Edition, 2008.

ASILKAN, Özcan, IRMAK, Sezgin; “İkinci El Otomobillerin Gelecekteki Fiyatlarının Yapay Sinir Ağları ile Tahmin Edilmesi”, **Süleyman Demirel Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi**, Cilt 14, Sayı 2, 2009, s.375-391.

ATAŞ, İsa; “Açıklık Kuplajlı Mikroşerit Yama Antenin Rezonans Frekansının Yapay Sinir Ağları İle Belirlenmesi”, Dicle Üniversitesi, Fen Bilimleri Enstitüsü, Elektrik-Elektronik Mühendisliği Anabilim Dalı, Yüksek Lisans Tezi, Diyarbakır, 2011.

BİLEN, Gamze, Hatice; “Termoelektrik Modüllerden Oluşan Bir Sistemin Yapay Sinir Ağları İle Modellenmesi”, Gazi Üniversitesi, Bilişim Enstitüsü, Elektronik Ve Bilgisayar Eğitimi, Yüksek Lisans Tezi, Ankara, 2011.

BOSE, Nirmal, K., LIANG, Ping; **Neural Network Fundamentals with Graphs, Algorithms and Applications**, United States, McGraw-Hill, 1996.

BROOMHEAD, Dave, S., LOWE, David; “Radial Basis–Functions, Multi-Variable Functional Interpolation and Adaptive Networks”, **Royal Signals and Radar Establishment Memorandum**, 1988.

CARPENTER, Gail, A., GROSSBERG, Stephen; “ART2: Self-Organization of Stable Category Recognition Codes for Analog Input Patterns”, **Applied Optics**, Vol.26, No.3, 1987, pp.4919-4930.

ÇELEBİ, Arif; “Pnömatik Bir Sistemin Parametrik ve Yapay Sinir Ağları İle Tanılanması”, İstanbul Teknik Üniversitesi, Fen Bilimler Enstitüsü, Makine Mühendisliği Anabilim Dalı, Yüksek Lisans Tezi, İstanbul, 2010.

ÇUHADAR, Murat, GÜNGÖR, İbrahim, GÖKSU, Ali; “Turizm Talebinin Yapay Sinir Ağları ile Tahmini ve Zaman Serisi Yöntemleri ile Karşılaştırmalı Analizi: Antalya İline Yönelik Bir Uygulama”, **Süleyman Demirel Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi**, Cilt 14, Sayı 1, 2009, s.99-114.

DOĞAN, Gülay; “Yapay Sinir Ağları Kullanılarak Türkiye'deki Özel Bir Sigorta Şirketinde Portföy Değerlendirmesi”, Hacettepe Üniversitesi, İstatistik Anabilim Dalı, Yüksek Lisans Tezi, Ankara, 2010.

DOĞAN, Emrah, IŞIK, Sabahattin, SANDALCI, Mehmet; “Günlük Buharlaşmanın Yapay Sinir Ağları Kullanarak Tahmin Edilmesi” **İMO Teknik Dergi**, Yazı 271, 2007, s.4119 -4131.

EBRAHİMZADEH, Ata, RANAEE, Vahid; “Control Chart Pattern Recognition Using an Optimized Neural Network and Efficient Features”, **ISA Transactions**, Vol.49, 2010, pp.387-393.

EĞRİOĞLU, Erol, ALADAĞ, Çağdaş, Hakan; “Yapay Sinir Ağları ve Arıma Modellerinin Melez Yaklaşımı İle Zaman Serilerinde Öngörü”, **16.İstatistik Araştırma Sempozyumu**, 2007.

EFE, Önder, KAYNAK, Okyay; **Yapay Sinir Ağları ve Uygulamaları**, İstanbul, Boğaziçi Üniversitesi Basım Evi, 1.Baskı, 2000.

ELMAS, Çetin; **Yapay Sinir Ağları**, Ankara, Seçkin Yayıncılık, 1.Baskı, 2003.

ENDAR, Burak; “Hvac Sistemlerinin Yapay Sinir Ağları İle Denetlenmesi”, Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Elektronik Mühendisliği Anabilim Dalı, Yüksek Lisans Tezi, İstanbul, 2010.

FAHLMAN, Scott E.; “An Empirical Study of Learning Speed in Back-Propagation Networks”, **Technical Report CMU-CS-88-162**, Carnegie Mellon University, 1988.

FUKUSHIMA, Kunihiko; “Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position”, **Biological Cybernetics**, Vol.16, 1980, pp.193-202.

FUKUSHIMA, Kunihiko; “Neocognitron: A New Algorithm For Pattern Recognition of Deformations And Shifts in Position”, **Pattern Recognition**, Vol.15, 1982, pp.455-469.

GAN, Qintao; “Adaptive Synchronization of Cohen–Grossberg Neural Networks with Unknown Parameters and Mixed Time-Varying Delays”, **Commun Nonlinear Sci Numer Simulat**, Vol.17, 2012, pp.3040–3049.

HANBAY, Davut; “Yapay Sinir Ağı Tabanlı Akıllı Yöntemlerle Karmaşık Sistemlerin Modellenmesi”, Fırat Üniversitesi, Fen Bilimleri Enstitüsü, Doktora Tezi, Elazığ, 2007.

HAYKIN, Simon; **Neural Networks: A Comprehensive Foundation**, New York, MacMillan College Publishing Company, 1.Edition, 1994.

HAYKIN, Simon; **Neural Networks: A Comprehensive Foundation**, New Jersey, Prentice-Hall, Inc., 2.Edition, 1999.

HEBB, Donald, O.; **The Organization of Behavior**, New York: Willey, Introduction and Chapter 4, "The First Stage of Perception: Growth of the Assembly", pp.60-78, 1949.

HOPPFIELD, Jhon, J.; "Neural Networks and Physical Sytems with Emergent Collective Computational Abilities", **Proc. of the Natural Academy of Science**, Vol.79, 1982, pp.2554-2558.

HWANG, Gene, J., DİNG, Adam, A.; " Prediction Intervals for Artifical Neural Networks", **Journal of the American Statistical Association**, Vol.92, No.438, 1997, pp.748-757.

JAMES, William; **Psychology (Briefer Course)**, New York: Holt, Chapter XVI, "Association", 1890, pp.253-279

JAYALAKSHMI, T., SANTHAKUMARAN, A.; "Statistical Normalization and Back Propagation for Classification", **International Journal of Computer Theory and Engineering**, Vol.3, No.1, 2011, pp.1793-8201.

KAFTAN, İlknur; "Batı Türkiye Gravite ve Deprem Katalog Verilerinin Yapay Sinir Ağları İle Değerlendirilmesi", Dokuz Eylül Üniversitesi, Fen Bilimleri Enstitüsü, Jeofizik Mühendisliği Bölümü, Jeofizik Mühendisliği Anabilim Dalı, Doktora Tezi, İzmir, 2010.

KARAHAN, Mehmet; "İstatistiksel Tahmin Yöntemleri: Yapay Sinir Ağları Metodu İle Ürün Talep Tahmini Uygulaması", Selçuk Üniversitesi, Sosyal Bilimler Enstitüsü, İşletme Anabilim Dalı, Doktora Tezi, Konya, 2011.

KAYA, İhsan, OKTAY, Selda, ENGİN, Orhan; "Kalite Kontrol Problemlerinin Çözümünde Yapay Sinir Ağlarının Kullanımı", **Erciyes Üniversitesi Fen Bilimleri Enstitüsü Dergisi**, Cilt 21, Sayı 1-2, 2005, s.92-107.

KAYA, Yıldırım; “Türk Telekomünikasyon A.Ş.’de Elektrik Enerjisi Tüketiminin Yapay Sinir Ağları İle Yerel Tahmini”, Yıldız Teknik Üniversitesi, Fen Bilimler Enstitüsü, Endüstri Mühendisliği Anabilim Dalı, Yüksek Lisans Tezi, İstanbul, 2010.

KOHONEN, Teuvo; **Self Organization and Associative Memory**, New York, Springer Verlag, 1984.

LIKAS, Aristidis, STAFYLOPATIS, Andreas; “Training the random neural network using quasi-Newton methods”, **European Journal of Operational Research**, Vol.126, 2000, pp.331-339.

MARQUARDT, Donald, W.; “An Algorithm for Least-Squares Estimation on Nonlinear Parameters”, **Journal of the Society for Industrial and Applied Mathematics**, Vol.11, No.2, 1963, pp.431-441.

MCCULLOCH, Warren, S., PITTS, Walter; “A Logical Calculus of Ideas Imminent in Nervous Activity”, **Buttein of Mathematics and. Biophysics**, Vol.5, 1943, pp.115-133.

ÖZTEMEL, Ercan; **Yapay Sinir Ağları**, İstanbul, Papatya Yayıncılık, 1.Baskı, 2003.

ÖZÜN, Alper; “Yapay Sinir Ağları ile Risk Öngörüsü”, Marmara Üniversitesi, Sosyal Bilimler Enstitüsü, Gazetecilik Anabilim Dalı, Bilişim Bilim Dalı, Doktora Tezi, İstanbul, 2011.

Özveren, Uğur; “Pem Yakıt Hücrelerinin Yapay Sinir Ağları İle Modellenmesi”, Yüksek Lisans Tezi, YTÜ Fen Bilimleri Enstitüsü, İstanbul, 2006.

PHAM, D, T., ÖZTEMEL, Ercan; Intelligent Quality Control, Springer Verlag, 1996.

ROJAS, Raul; **Neural Networks**, Berlin, Springer-Verlag, 1996

ROSENBLATT, Frank; “The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain”, **Psychological Review**, Vol.65, 1958, pp.386–408.

RUMELHART, David, E., HINTON, Geoffrey, E., WILLIAMS, Ronald, J.; “Learning Representations of Back-propagation Errors”, **Nature**, Vol.323, No.9, 1986, pp.533-536.

SAĞIROĞLU, Şeref, BEŞDOK, Erkan, ERLER, Mehmet; **Mühendislikte Yapay Zeka Uygulamaları-1: Yapay Sinir Ağları**, Kayseri, Ufuk Yayıncılık, 2003.

SEÇME, Gökhan; “Akış Tipi Çizelgeleme Problemlerinin Yapay Sinir Ağları İle Modellenmesi”, Erciyes Üniversitesi, Sosyal Bilimler Enstitüsü, İşletme Anabilim Dalı, Yüksek Lisans Tezi, Kayseri, 2006.

SHEWCHUK, Jonathan, Richard; **An Introduction to the Conjugate Gradient Method Without the Agonizing Pain**, Edition 11, August, 1994.

SEJNOWSKI, Terrence, J., TESAURO, Gerald; **The Hebb Rule for Synaptic Plasticity: Algorithms and Implementations**, In Byrne, J. And Berry, W. O. (Eds.) Neural Models of Plasticity, Academic Press, New York, pp.94-103, 1989.

SLAUGHTER, Gymama, E.; “ Artificial Neural Network for Temporal Impedance Recognition of Neurotoxins”, M.S. Thesis, Virginia Commonwealth University, Virginia, 2003, p.115.

SPECTH, Donald, F.; “Probabilistic neural networks for classification, mapping, or associative memory”, **IEEE Conference on Neural Networks**, Vol.1, 1998, pp.525-532.

SPECTH, Donald, F.; “A General Regression Neural Network”, **IEEE Transactions on Neural Networks**, Vol.2, No.6, 1991, pp.568-576.

SYMEONIDIS, Klimis; “Hand Gesture Recognition Using Neural Networks”, School of Electronic and Electrical Engineering, Master of Science in Multimedia Signal Processing Communications, 2000.

TEMİZEL, İlknur; “Melas Atık Sularının Arıtımında Yukarı Akışlı Havasız Çamur Yataklı Reaktör (Yahçyr) Sisteminden Elde Edilen Biyogaz ve Metan Üretimlerinin Yapay Sinir Ağları İle Modellenmesi”, Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Çevre Mühendisliği Programı, Çevre Mühendisliği Anabilim Dalı, Yüksek Lisans Tezi, İstanbul, 2012.

TOKTAŞ, İhsan, AKTÜRK, Nizami; **Makine Tasarım İşleminde Kullanılan Yapay Zeka Teknikleri ve Uygulama Alanları**, Makine Teknolojileri Elektronik Dergisi, Sayı 2, 2004, s.7-20.

YALÇIN, Nesibe; “Sezgisel Algoritma Öğrenmeli Yapay Sinir Ağları İle Epilepsi Hastalığının Teşhisi”, Selçuk Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı, Yüksek Lisans Tezi, Konya, 2012.

YAZICI, Ümmü, Habibe; “Yapay Sinir Ağları İle Banka Müşterisi Bekleme Süresi Tahmini”, Atatürk Üniversitesi, Fen Bilimleri Enstitüsü, Endüstri Mühendisliği Anabilim Dalı, Yüksek Lisans Tezi, Erzurum, 2010.

YURTOĞLU, Hasan; “Yapay Sinir Ağları Metodolojisi İle Öngörü Modellemesi: Bazı Makroekonomik Değişkenler İçin Türkiye Örneği”, Devlet Planlama Teşkilatı, Uzmanlık Tezi, Yayın No: DPT: 2683, Ankara, 2005.

YÜKSEK, Ahmet, Gürkan, BİRCAN, Hüdaverdi, ZONTUL, Metin, KAYNAR, Oğuz; “Sivas İlinde Yapay Sinir Ağları ile Hava Kalitesi Modelinin Oluşturulması Üzerine Bir Uygulama”, **Cumhuriyet Üniversitesi İktisadi ve İdari Bilimler Dergisi**, Cilt 8, Sayı 1, 2007.

ZURADA, Jacek, M.; **Introduction to Artificial Neural Networks**, New York, West Publishing Company, 1992.

WANG, Yangfan, LİN, Ping, WANG, Linshan; “Exponential Stability of Reaction–Diffusion High-Order Markovian Jump Hopfield Neural Networks With Time-Varying Delays”, **Nonlinear Analysis: Real World Applications**, Vol.13, 2012, pp.1353–1361.

WARNER, Brad, MISRA, Manavendra; “Understanding Neural Networks as Statistical Tools”, **The American Statistician**, Vol.50,1996, pp.284-293.

WIDROW, Bernard, HOFF, Marcian, E.; **Adaptive Switching Circuits**, Part 4, 1960

WIDROW, Bernard, LEHR, Michael, A.; “30 Years of Adaptive Neural Networks: Perceptron, Madaline, and Backpropagation”, **Proceedings of The IEEE**, Vol.78, No.9, 1990, pp.1415-1442.

WIENER, Norbert; **Cybernetics: Or Control and Communication in The Animal and The Machine**, Paris, Camb. Mass. (MIT Press), 2. Edition, 1948.

ÖZET

DEVECİ, Muhammet. Yapay Sinir Ağları ve Bekleme Süresinin Tahmininde Kullanılması, Yüksek Lisans Tezi, Ankara, 2012.

Son yıllarda, yapay sinir ağları pek çok alanda uygulanan ve üzerinde en çok araştırma yapılan yöntemlerden birisidir. Yapay Sinir Ağ teknikleri karmaşık ve doğrusal olmayan modellerde etkilidir. YSA araştırmaları; sınıflandırma, optimizasyon, tahmin, şekil tanıma, modelleme ve öğrenme gibi birçok alanda kullanılmaktadır.

Bu çalışmada yapay sinir ağlarının genel yapısı ve çalışma mimarisi tanıtılmıştır. Uygulama kısmında yapay sinir ağları ile hastaların bekleme süresinin tahmini yapılmıştır. Çeşitli istatistiksel normalizasyon teknikleri öğrenme yöntemleri ile birlikte kullanılmıştır. Sonuçlar gösterdi ki bekleme süresinin tahmin performansı, yapay sinir ağlarında kullanılan normalizasyon yöntemlerine bağlıdır. Sonuç olarak; yapay sinir ağlarının öğrenme ve normalizasyon teknikleri ile geleceği tahmin etmede başarıyla kullanılabileceği ortaya konulmuştur.

Anahtar Sözcükler:

- 1- Yapay Sinir Ağları
- 2- Tahmin
- 3- Normalizasyon
- 4- Performans
- 5- Bekleme Süresi

ABSTRACT

DEVECİ, Muhammet. Artificial Neural Networks and Used of Waiting Time Estimation, Graduate Thesis, Ankara, 2012.

In recent years, artificial neural networks have been applied in many areas and it is one of the most commonly researched methods. Artificial Neural Network (ANN) techniques are effective for the complex and non-linear models. ANN research techniques are applied to various fields such as classification, optimization, forecasting, recognition, modeling and learning.

This paper introduces basic structure and working architecture of artificial neural networks. Patient's waiting time has been estimated by the help of artificial neural networks for the application part. The various statistical normalization techniques used with by combination of learning techniques. The results showed that, the estimated performance of waiting time depends on normalization techniques, which is used in the neural networks. As a conclusion, it has been shown that artificial neural networks can be used successfully to forecast future by using normalization and learning techniques.

Key Words:

- 1- Artificial Neural Networks
- 2- Forecasting
- 3- Normalization
- 4- Performance
- 5- Waiting Time