

T.C.

İnönü Üniversitesi Sosyal Bilimler Enstitüsü

Ekonometri Bölümü Ekonometri Anabilim Dalı

LDL KOLESTEROLÜN YAPAY SİNİR AĞLARI İLE TAHMİNİ

Oktay KIZILKAYA

Yrd. Doç. Dr. Rabia Ece OMay

Yüksek Lisans Tezi

Malatya, 2013

LDL KOLESTEROLÜN YAPAY SİNİR AĞLARI İLE TAHMİNİ

Oktay KIZILKAYA

İnönü Üniversitesi Sosyal Bilimler Enstitüsü Ekonometri Ana Bilim Dalı

Yrd. Doç. Dr. Rabia Ece OMay

Yüksek Lisans Tezi

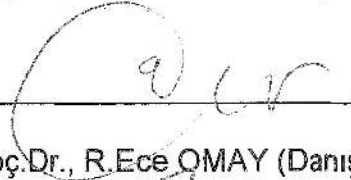
Malatya, 2013

KABUL VE ONAY

Oktay KIZILKAYA tarafından hazırlanan "Ldl Kolesterolün Yapay Sinir Ağları ile Tahmini" başlıklı bu çalışma, 16.08.2013 tarihinde yapılan savunma sınavı sonucunda başarılı bulunarak jürimiz tarafından Yüksek Lisans Tezi olarak kabul edilmiştir.



Yrd.Doç.Dr., Hasan SÖYLER (Başkan)



Yrd.Doç.Dr., R.Ece OMA Y (Danışman)



Yrd.Doç.Dr., Ahmet UĞUR

Yukarıdaki imzaların adı geçen öğretim üyelerine ait olduğunu onaylarım.

Prof.Dr.
Mehmet KARAGÖZ
Enstitü Müdürü

ONUR SÖZÜ

Yüksek lisans tezi olarak sunduğum “LDL Kolesterolün Yapay Sinir Ağları ile Tahmini” başlıklı bu çalışmanın, bilimsel ahlak ve geleneklere aykırı düşecek bir yardıma başvurulmaksızın tarafımdan yazıldığını ve yararlandığım bütün yapıtların hem metin içinde hem de kaynakça da yöntemine uygun biçimde gösterilenlerden oluştuğunu belirtir, bunu onurumla doğrularım.

16/08/2013

Oktay KIZILKAYA

BİLDİRİM

Hazırladığım tezin tamamen kendi çalışmam olduğunu ve her alıntıya kaynak gösterdiğimi taahhüt eder, tezimin kâğıt ve elektronik kopyalarının İnönü Üniversitesi Sosyal Bilimler Enstitüsü arşivlerinde aşağıda belirttiğim koşullarda saklanmasına izin verdiğimi onaylarım.

- ☐ Tezimin tamamı her yerden erişime açılabilir.
- ☐ Tezim sadece İnönü Üniversitesi yerleşkelerinden erişime açılabilir.
- ☐ Teziminsüreyle erişime açılmasını istiyorum.

16/08/2013

Oktay KIZILKAYA

ÖZET

Oktay Kızılkaya. LDL Kolesterolün Yapay Sinir Ağları ile Tahmini, Yüksek Lisans Tezi, Malatya, 2013.

Yapay sinir ağları (YSA), farklı disiplinlerdeki karmaşık problemlerin çözümlenmesinde kabul gören ve uygulamalarda sıklıkla yer alan modelleme araçları haline gelmiştir. Farklı YSA yapıları, tıp alanında karar destek sistemlerinin gelişmesinde kullanılmakta olan önemli modellerdendir. Bu çalışmada, üç farklı algoritma ile eğitilen çok katmanlı sinir ağları LDL Kolesterol'ün tahmininde kullanılmış ve en başarılı algoritma belirlenmiştir. Öğrenme oranı ve momentumlu geri yayılım algoritması, Levenberg-Marquardt geriyayılım algoritması ve Bayesyen düzeltmeye dayalı geri yayılım algoritması çalışılmış olan üç algoritmadır. Çok katmanlı sinir ağlarının eğitimi ve testi veri tabanında yer alan farklı kişilere ait kayıtlar ile yapılmıştır. Performans belirleyiciler ve istatistiksel ölçümler ile çok katmanlı sinir ağları değerlendirilmiş ve sonuçlar LDL Kolesterol'ün tahmininde Bayesyen düzeltmeye dayalı geri yayılım algoritmasının en başarılı eğitim algoritması olduğunu göstermiştir.

Anahtar Kelimeler: Çok katmanlı yapay sinir ağları, eğitim algoritmaları, LDL Kolesterolün tahmini.

ABSTRACT

Oktay Kızılkaya. Estimation of LDL Cholesterol by Artificial Neural Networks,

Master Thesis, Malatya, 2013.

Artificial neural networks (ANNs) have become modeling tools that have found extensive acceptance and they have frequently used in applications in many disciplines for solving complex problems. Different ANN structures are valuable models, which are used in the medical field for the development of decision support systems. In this study, three multilayer neural networks trained with different algorithms were used for estimation of LDL Cholesterol and the most efficient training algorithm was determined. Gradient descent with momentum and adaptive learning rate backpropagation, Levenberg-Marquardt backpropagation, Bayesian regulation backpropagation were the studied three training algorithms. The multilayer neural networks were trained and tested with subject records from the database. Performance indicators and statistical measures were used for evaluating the multilayer neural networks and the results demonstrated that the Bayesian regulation backpropagation algorithm was the most efficient training algorithm for estimation of LDL Cholesterol.

Key Words: Multilayer neural networks, training algorithms, estimation of LDL Cholesterol.

TEŞEKKÜR

Tez çalışmam süresince bilgi ve öneri anlamında hiçbir zaman desteğini esirgemeyen ve bu tezin oluşmasında büyük rol oynayan çok değerli ve saygıdeğer hocam Yrd. Doç. Dr. R. Ece OMAY'a,

Çalışmalarım sırasında bana yardımcı olan bütün bölüm hocalarım ve araştırma görevlisi arkadaşlarıma, tez çalışmamın uygulama aşamasında yardımını esirgemeyen Yrd. Doç. Dr. Çiğdem GİRİFTİNOĞLU'na,

Tezi hazırlama aşamasında bana destek olan eşim Fatma KIZILKAYA'ya,

Bana maddi ve manevi her türlü desteği veren aileme,

Sonsuz teşekkür ederim.

İÇİNDEKİLER

ONUR SÖZÜ	I
BİLDİRİM	II
ÖZET	III
ABSTRACT	IV
TEŞEKKÜR	V
ŞEKİLLER DİZİNİ	IX
TABLolar DİZİNİ	XI
KISALTMALAR LİSTESİ	XII
1. GİRİŞ	1
1.1 Tahmin	2
1.2 Tahmin Yöntemleri	2
1.3 Yapay Sinir Ağları ve Tahmin	5
2. YAPAY SİNİR AĞLARI'NA GENEL BAKIŞ	6
2.1 Yapay Zekâ	6
2.2 YSA'nın Tanımı ve Özellikleri	7
2.3 YSA'nın Tarihçesi	10
2.4 YSA'nın Uygulama Alanları	12
2.4.1 Arıza Analizi ve Tespiti	13
2.4.2 Sağlık	13
2.4.3 Savunma Sanayi	14
2.4.4 Haberleşme	14
2.4.5 Üretim	14
2.4.6 Otomasyon ve Kontrol	14
2.5 Sağlık Alanında YSA'nın Yeri	14
2.6 YSA'nın Üstünlükleri	16

2.6.1	Doğrusal Olmama	16
2.6.2	Paralellik	16
2.6.3	Gerçeklenme Kolaylığı	17
2.6.4	Yerel Bilgi İşleme	17
2.6.5	Hata Toleransı	17
2.6.6	Öğrenebilirlik	17
2.6.7	Genelleme	18
2.6.8	Uyarlanabilirlik	18
2.6.9	Donanım ve Hız	18
2.6.10	Analiz ve Tasarım Kolaylığı	18
2.7	Biyolojik Nöron Yapısı	19
2.8	Yapay Sinir Hücresi	21
2.8.1	Girdiler	21
2.8.2	Ağırlıklar	22
2.8.3	Toplama Fonksiyonu:	22
2.8.4	Aktivasyon Fonksiyonu	22
2.8.5	Çıktı	27
2.9	YSA'nın Yapısı	27
2.10	YSA'da Öğrenme	28
2.10.1	Öğretmensiz öğrenme	29
2.10.2	Öğretmenli öğrenme	30
2.10.3	Takviyeli öğrenme	30
2.11	YSA'nın Sınıflandırılması	30
2.11.1	YSA'nın Yapısındaki Katmanlara Göre Sınıflandırılması	31
2.11.2	YSA'nın Mimari Yapılarına Göre Sınıflandırılması	33
2.12	YSA Tasarımı	36
2.12.1	Ağ Yapısı	36
2.12.2	Başlangıç Değerlerini Belirleme	37
2.12.3	Eğitim Kümesinin Büyüklüğü	38
2.12.4	Öğrenme Parametreleri	38
2.12.5	Aktivasyon Fonksiyonu	39
2.12.6	Veri Normalleştirme	39
2.12.7	Öğrenme Algoritması	40

2.12.8	Veri Kümesinin Düzenlenmesi	41
2.12.9	Performans fonksiyonu seçimi	41
3.	GERİ YAYILIM(BACKPROPAGATION) YAPAY SİNİR AĞLARI	43
3.1	Giriş.....	43
3.2	Öğrenme Oranı, Momentum Katsayısı ve Bias	44
3.3	Standart Geri Yayılım Algoritması	44
3.4	Eğitim Algoritması.....	47
3.5	Kesin Yerel Geri Yayılım (Strictly Local Backpropagation).....	50
3.6	Levenberg-Marquardt ve Bayesian Regülasyon (Regulation) Metodu.....	53
4.	UYGULAMA	56
4.1	Uygulamada Kullanılacak Değişkenler Hakkında Genel Bilgiler	56
4.2	Bulgular.....	59
4.3	Sonuç ve Tartışma.....	64
	KAYNAKÇA	66

ŞEKİLLER DİZİNİ

	Sayfa No
Şekil 2.1 Nöron Yapısı	19
Şekil 2.2 Yapay Sinir Hücresinin Yapısı	21
Şekil 2.3 Tek Kutuplu Eşik Fonksiyonu	23
Şekil 2.4 Çift Kutuplu Eşik Fonksiyonu	24
Şekil 2.5 Doğrusal Aktivasyon Fonksiyonu	24
Şekil 2.6 Sigmoid Fonksiyonu	25
Şekil 2.7 Hiperbolik Tanjant Fonksiyonu	26
Şekil 2.8 Sinüs Fonksiyonu	26
Şekil 2.9 Yapay Sinir Ağının Yapısı	28
Şekil 2.10 İki Girdi ve Bir Çıktıdan Oluşan En Basit TKA Modeli	31
Şekil 2.11 İki Gizli Katmana Sahip ÇKA	33
Şekil 2.12 İleri Beslemeli Ağ için Blok Diyagramı	34
Şekil 2.13 İleri Beslemeli 3 Katmanlı YSA	34
Şekil 2.14 Geri Beslemeli Ağ için Blok Diyagram	35
Şekil 3.1 Bir Gizli Tabakalı Geri Yayılımlı Ağ	45
Şekil 3.2 Kesin Yerel Geri Yayılımlı Ağ	52
Şekil 4.1 Gizli Nöron Sayısı 14 Olan Levenberg-Marquardt Öğrenme Algoritmasının Eğitim Performansı	60
Şekil 4.2 Gizli Nöron Sayısı 14 Olan Levenberg-Marquardt Öğrenme Algoritmasının Test Performansı	61
Şekil 4.3 Gizli Nöron Sayısı 14 Olan Bayesian Regulasyon (Regulation) Öğrenme Algoritmasının Eğitim Performansı	62

Şekil 4.4 Gizli Nöron Sayısı 14 Olan Bayesian Regulasyon (Regulation) Öğrenme Algoritmasının Test Performansı	62
Şekil 4.5 Gizli Nöron Sayısı 14 Olan Öğrenme Oranı ve Momentumlu Geri Yayılım Öğrenme Algoritmasının Eğitim Performansı	63
Şekil 4.6 Gizli Nöron Sayısı 14 Olan Öğrenme Oranı ve Momentumlu Geri Yayılım Öğrenme Algoritmasının Test Performansı	64

TABLÖLER DİZİNİ

	Sayfa No
Tablo 2.1 Sağlık Alanında YSA ile Yapılan Bazı Çalışmalar	15
Tablo 3.1 Standart Geri Yayılım ve Kesin Yerel Geri Yayılımın Karşılaştırılması	53
Tablo 4.1 Levenberg-Marquardt Öğrenme Algoritmasının Farklı Gizli Nöron Sayılarına Göre Performansı	60
Tablo 4.2 Bayesian Regulasyon (Regulation) Öğrenme Algoritmasının Farklı Gizli Nöron Sayılarına Göre Performansı	61
Tablo 4.3 Öğrenme Oranı ve Momentumlu Geri Yayılım Öğrenme Algoritmasının Farklı Gizli Nöron Sayılarına Göre Performansı	63

KISALTMALAR LİSTESİ

YSA	Yapay Sinir Ağları
GYA	Geri Yayılım Öğrenme Algoritması
TKA	Tek Katmanlı Yapay Sinir Ağları
ÇKA	Çok Katmanlı Yapay Sinir Ağları
İBYSA	İleri Beslemeli Yapay Sinir Ağları
GBYSA	Geri Beslemeli Yapay Sinir Ağları
LDL	Düşük Yoğunluklu Lipoprotein
HDL	Yüksek Yoğunluklu Lipoprotein
MSE	Hata Kareler Ortalaması

BÖLÜM 1

1. GİRİŞ

Yapay sinir ağları (YSA), doğrusal olmayan yapıları ve süreklilikleri sayesinde tahmin, fonksiyon yaklaştırma, desen sınıflandırma, veri ilişkilendirme, kümeleme, veri filtreleme, optimizasyon ve kontrol amacıyla, finansal alanlardan, tıp alanına, savunma sanayinden, otomasyon ve kontrol alanlarına kadar birçok alanda sıkça kullanılmışlardır. YSA, geleneksel sistemlerden farklı çalışma yöntemleri ile doğrusal olmayan birçok problemim çözümünde başarı ile kullanılmaktadır. YSA'nın öğrenme özelliği sayesinde herhangi bir olay hakkında girdi ve çıktılar arasındaki ilişkiyi, doğrusal olsun veya olmasın, elde bulunan mevcut örneklerden öğrenerek daha önce hiç görülmemiş olayları, önceki örneklerden çağrışım yaparak ilgili olaya çözümler üretebilme özelliği YSA'lardaki zeki davranışın da temelini teşkil eder.

Bu çalışma dört bölümden oluşmakta ve çalışmanın birinci bölümünde tahmin ve tahmin yöntemlerine kısaca değinilmiştir. İkinci bölümünde yapay sinir ağlarından genel olarak bahsedilmektedir. Bu bağlamda öncelikle yapay zekâ kavramı tanıtılmış, YSA'nın gelişimi incelenmiş, YSA'nın yapısı ve uygulama alanları hakkında bilgiler verilmiştir. Daha sonra YSA'nın öğrenmesi ve sınıflandırılması anlatılmış, yapay sinir ağı tasarımında önemli olan noktalara değinilmiştir. Üçüncü bölümde geri yayılım (backpropagation) yapay sinir ağları ayrıntılı bir şekilde anlatılmış, uygulama kısmında kullanılacak olan eğitim algoritmalarına değinilmiştir. Dördüncü ve son bölümde öncelikle uygulamada kullanılan veri setlerinin tanıtımı yapılmıştır. Daha sonra uygulamada kullanılacak olan üç farklı eğitim algoritması ile kurulan YSA eğitilmiş, performansları incelenmiş ve bu performanslar karşılaştırılmıştır.

Bu çalışmanın amacı, bir tahmin tekniği olarak YSA'yı tanıtmak, eğitiminde kullanılan üç farklı algoritmanın performanslarını elde etmek ve bu performansları karşılaştırmaktır.

1.1 Tahmin

Geleceği öngörü veya tahmin kelimeleri ile belirteceğimiz kavram, bir değişkenin gelecekte alabileceği değerlerin önceden yaklaşık olarak belirlenmesi olarak tanımlanabilir. Tahmin, işletme, finans, yönetim bilimi, siyaset bilimi, planlama ve daha birçok alanda karşılaşılan bir kavramdır. Daha doğrusu gelecekle ilgili karar vermeyi gerektiren durumlarda, tahmin, karar vericinin yardımcısı olacaktır. Tahminin amacı karar vermedeki riski azaltmaktır. İyi bir yönetim sisteminin önemli bir karakteristiği, belirsizlik altında en iyi performansa ulaşabilme kabiliyetidir. Tahmin doğruluğunun iyileştirilmesi belirsizliği azaltacaktır.

Herhangi bir değişkenin gelecekte alacağı değerlerin tahmini, değişkenin geçmiş dönemlerdeki gözlem değerleri kullanılarak oluşturulan modeller aracılığıyla sağlanır. Geleceği tahmin amaçlı kurulan bir tahmin sistemini iki aşamada incelemek mümkündür. Bunlardan ilki model kurma, ikinciside tahmin aşamasıdır. Yapılacak tahminler, kurulan modele göre gerçekleştirileceğinden dolayı model kurma aşaması oldukça önemlidir. Doğru model, gerçeğe daha yakın tahminlerin üretilmesini sağlayacaktır.

Model kurma aşamasının başlangıcında, ilgili probleme ait veriler sağlıklı bir şekilde toplanır ve toplanan verilere uygun model belirlenir. Belirlenen modelin matematiksel biçimi tayin edilerek, mevcut veriler yardımı ile modelin parametreleri belirlenmeye çalışılır. Model parametrelerinin ve modelin uygunluğu hata testleri ile sağlanabilir. Eğer kurulan model uygun bir model ise, gelecek için tahminler yapılabilir (Hamzaçebi, 2011).

1.2 Tahmin Yöntemleri

Tahmin yöntemleri, literatürde farklı şekillerde sınıflandırılmış olmakla beraber temel olarak iki grupta ele alınmaktadır:

1. Nitel (Kalitatif) Yöntemler
2. Nicel (Kantitatif) Yöntemler

Genel olarak nicel yaklaşımların girdisi, çeşitli zaman aralıklarında toplanmış olan verilerdir. Verilerin iyi bir şekilde analiz edilmesi, bu yöntemlerin temelini oluşturmaktadır. Buna karşılık nitel yaklaşımlar, konu ile ilgili uzmanların bilgi ve deneyimlerinden yararlanarak bu alandaki gelişmelerin ne yönde olacağı, ne tür ihtiyaçlar ortaya çıkaracağı gibi konularda yoğunlaşmaktadır (Archer, 1980). Uygulamada ileriye yönelik tahminler için nicel ve nitel yöntemlerin birlikte kullanıldığı da görülmektedir. Nicel yöntemlerle elde edilen tahmin sonuçları tarafsız ve deneyimli uzmanların görüşleri ışığında değerlendirilerek kullanılabilir.

Nitel tahmin yöntemleri, çalışma alanı konusunda uzman kabul edilen bireylerin yargılarına ve deneyimlerine dayanmaktadır. Bu yöntemlerde bilgi işleme süreci uzmanlar veya jüri üyeleri tarafından gerçekleştirilir. Beklentileri ifade etmeleri ve dolayısıyla sübjektif yargılara bağlı olmaları nedeniyle, nicel yöntemler gibi tekrarlanamayan, sonuçları tartışmaya açık yöntemler olsalar da, nitel yöntemlerin kullanılmasını zorunlu kılan bir takım nedenler bulunmaktadır. Örneğin pazara yeni sürülecek bir ürüne olan talebin nasıl gerçekleşeceğinin tahmini, pazar hakkında yetkin bilgisi olan bir karar vericinin kişisel görüşlerine dayanmak zorunda olabilir.

Nicel yöntemler, geçmiş dönem gözlem değerlerine dayalı analizler yapan tahmin modellerini kapsamaktadır. Kullanılan yöntemler; incelenen değişkende gözlenen gelişmelerin analiz edilmesi, veri serisinin dinamik özelliklerinin belirlenmesi ve bu özelliklerin matematiksel bir fonksiyon ile ifade edilerek geleceğe ilişkin öngörülerin türetilmesini içermektedir. Nicel tahmin yöntemleri genel olarak iki gruba ayrılmaktadır (De Lurgio, 1998).

1. İlişkiye Dayalı (Nedensel) Yöntemler
2. Zaman Serisi Yöntemleri

1.1 İlişkiye Dayalı Yöntemler

Nedensel ya da diğer adıyla ilişkiye dayalı tahmin yöntemlerinde; tahmini değeri bulunacak değişkenin çeşitli faktörlerden etkilendiği düşünülerek, bağımlı

değişkenin etkilendiği ya da bağlı olabileceği diğer değişkenlerin tespitine çalışılmakta ve bu etkilerin formüle edilmesi amaçlanmaktadır (Orhunbilge, 2002). Nedensel yöntemlerde bir değişkenin gelecekteki değerlerini tahmin etmekten çok iki ya da daha fazla değişken arasındaki ilişkinin açıklanmasına çalışılmaktadır. Bağımlı değişken ile ilişkisi olan değişken veya değişkenlerin belirlenmesi ve bu ilişkiye uygun bir matematiksel ifadenin bulunması, nedensel yöntemlerin temel amacını oluşturmaktadır. İlişkiye dayalı tahmin yöntemlerinin temelini regresyon analizi oluşturmaktadır. Regresyon analizi, herhangi bir değişkenin (bağımlı değişken) bir veya birden fazla değişkenle (bağımsız veya açıklayıcı değişken) arasındaki ilişkinin matematik bir fonksiyon şeklinde yazılmasıdır. Bu fonksiyona regresyon denklemi adı verilmektedir. Regresyon denklemi yardımıyla bağımlı değişken ile bağımsız değişken (ler) arasındaki ilişkiyi kuran parametrelerin değerleri tahmin edilir. Bağımlı değişkeni etkileyen bağımsız değişkenlerin tahmin edilmesi, bu değişken üzerinde geliştirilecek plan ve politikalarda hangi değişkenlerin önem kazandığının belirlenmesine yardımcı olmaktadır. Bu teknik sayesinde, hangi faktörlerde nasıl bir değişiklik yapılarak ilgilenilen değişkende artış veya azalış meydana geleceği ortaya çıkarılabilmektedir.

1.2 Zaman Serisi Yöntemleri

Zaman serisi yöntemleri, bir olaya ait geçmişteki verilerin incelenmesi ve belirli eğilimlerin ortaya çıkarılarak ileriye yönelik tahminlerin yapılması temeline dayanmaktadır. Bu yöntemlerin amacı, geçmiş gözlem değerlerindeki veri kalıplarını kullanarak istatistiksel modeller oluşturmak ve bu modellerle geleceği tahmin etmektir. Diğer bir anlatımla, çeşitli faktörlerin etkilerinin bir sonucu olarak ortaya çıkan zaman serilerinin bu faktörlerden nasıl etkilendiği incelenerek gelecekte alacağı değerler belirlenmeye çalışılmaktadır (Bülbül, 1994). Ekonomi ve iş dünyasındaki belirsizlikler nedeniyle, ekonomik zaman serilerinin gelecekte göstereceği seyri tahmin etmek, planlama ve karar alma açısından önemlidir. Zaman serisi yöntemlerinde geleceğin tahmini yanında geçmiş dönemlerin incelenmiş olması; geçmişteki olumlu ve olumsuz gelişmelerin tespit edilmesine, nedenlerinin araştırılmasına ve yapılan yanlışların tekrarlanmaması için gerekli tedbirlerin alınmasına da imkân sağlamaktadır. Sayılan bu özellikleri nedeniyle bilhassa orta ve

kısa dönem tahminlerine ihtiyaç duyulan her alanda yaygın olarak kullanılmaktadırlar (Akgül, 1994).

1.3 Yapay Sinir Ağları ve Tahmin

1980'li yılların sonlarından itibaren tahmin amaçlı kullanılan tekniklerden biri de YSA tekniğidir. YSA, girdi ve çıktı değişkenleri arasındaki herhangi bir ön bilgiye ihtiyaç duymadan doğrusal ve doğrusal olmayan modellemeyi sağlayabilmektedir. Bu sebeple YSA, tahmin aracı olarak diğer yöntemlere göre daha genel ve esnektir (Zhang, Patuwo, Hu, 1998).

YSA'nın tahmin problemlerinde kullanılmasına yönelik literatürde binlerce çalışma vardır. İlk uygulama 1964 yılında olmuştur. Hu (1964), tezinde ADALINE' ı hava durumu tahmini için kullanmıştır. Ancak çok katmanlı ağları eğitecek algoritmanın olmayışından dolayı bu çalışma oldukça kısıtlıdır (Zhang, Patuwo, Hu, 1998). 1986 yılında geri yayılım algoritmasının tanıtılmasından sonra, YSA'nın tahmin amaçlı kullanılmasında birçok gelişme olmuştur. Farklı disiplinlerden araştırmacıların yaptığı çalışmaların birçoğunda, geleneksel metodlar ile YSA teknikleri karşılaştırılmış ve hangi yöntemin daha iyi sonuç verdiği araştırılmıştır. Yapay sinir ağları, özellikle günümüze kadar çözümü güç ve karmaşık olan ya da ekonomik olmayan çok farklı alanlardaki problemlerin çözümüne uygulanmış ve genellikle başarılı sonuçlar elde edilmiştir.

BÖLÜM 2

2. YAPAY SİNİR AĞLARI'NA GENEL BAKIŞ

2.1 Yapay Zekâ

Yapay zekâ; bir bilgisayarın ya da bilgisayar denetimli bir makinenin, genellikle insana özgü nitelikler olduğu varsayılan akıl yürütme, anlam çıkartma, genelleme ve geçmiş deneyimlerden öğrenme gibi yüksek zihinsel süreçlere ilişkin görevleri yerine getirme yeteneği olarak tanımlanmaktadır. 1950'li yılların başında ortaya çıkan “yapay zekâ” çalışmaları gelişen bilgisayarlar tarafından matematiksel formülize edilemeyen problemlerin çözümünde kullanılmaya başlanmıştır. Bilgisayar teknolojisinin hızla gelişimi ile yapay zekâ sistemleri, insan karar verme sürecine oldukça benzer hareket edebilmekte ve daha karmaşık problemlerde sonuçlar üretebilmektedir. Özellikle belirli bir algoritma ve formülasyon kullanılarak çözülemeyen problemlerin çözülmesi için yapay zekâ sistemleri geliştirilmekte, problemin çözümü için geliştirilmiş bir algoritma mevcut ise geleneksel bilgisayar yöntemleri ile problem çözülebilmektedir.

Yapay zekâ sistemleri, literatürde zeki sistemler olarak adlandırılmaktadır. Burada en dikkat çekici nokta, bilgisayarların öğrenme yetileridir. Zeki sistemler için öğrenme “zaman içinde yeni bilgilerin keşfedilmesi yoluyla davranışların iyileştirme süreci” olarak tanımlanmaktadır. Zeki sistemlerde öğrenme şekilleri şu şekilde sıralandırılabilir (Öztemel, 2008);

- Alışkanlık yolu ile öğrenme,
- Görerek öğrenme,
- Talimatlardan öğrenme,
- Örneklerden öğrenme,
- Analoji yolu ile öğrenme,

- Açıklamalardan öğrenme,
- Deney yolu ile öğrenme,
- Keşfetmek yolu ile öğrenme.

Beynin nasıl çalıştığını henüz kimse anlamış değildir. Fakat burada beyni oluşturan milyarlarca nöronun önemli rol oynadığı bilinmektedir. Son yıllarda, yeniden araştırmacıların ilgisini beynin işlevinin modellenmesi, gerçek bir taklidin tasarlanması çekmektedir. Burada “yeniden” kelimesinin kullanılması, çalışmaların ilk olarak 1943 senesinde Mc Colloch-Pitts modeli ile başladıktan sonra 80’lerde teknolojinin gelişmesi ile yeniden hız kazanmasıdır. Bu modeller, canlı organizmalarda bulunan biyolojik sinir yapısından esinlenerek yapılmıştır (Nabiyev, 2005).

2.2 YSA’nın Tanımı ve Özellikleri

“Yapay sinir ağları, insan beyninin sinir hücresinin çalışma şeklini taklit ederek, sistemlere öğrenme, genelleme, hatırlama özelliklerinin kazandırılması” (Saraç, 2004) şeklinde ifade edilir. Bir diğer YSA tanımı, ilk ticari yapay sinir ağının geliştiricisi olan Dr. Robert Hecht-Nielsen'e aittir: "Yapay sinir ağı, dışarıdan gelen girdilere dinamik olarak yanıt oluşturma yoluyla bilgi işleyen, birbiriyle bağlantılı basit elemanlardan oluşan bilgi işlem sistemidir" (Caudill, 1987). Bu tanıma yakın bir tanım da YSA yazınında çok tanınan Teuvo Kohonen'e ait bir tanımdır: "Yapay sinir ağları, paralel bağlı çok sayıdaki basit elemanın, gerçek dünyanın nesneleriyle biyolojik sinir sisteminin benzeri yolla etkileşim kuran hiyerarşik bir organizasyondur" (Kohonen, 1987).

Yapay zekânın bir dalı olan YSA, biyolojik beyinden daha iyisini yapmaya çalışan büyük paralel hesaplama mekanizmalarıdır. Bunlar zekâyı birbiriyle bağlantılı ağırlıklarında depolarlar. Bu değişken ağırlıklar, nöronları paralel ve ardışık bir biçimde birleştirirler. Tüm mekanizma hiyerarşik olarak girdi vektörünü, nöronlar ve ağırlıklar doğrultusunda işleyerek çıktı vektörüne ulaştırır (Smith, 1994).

YSA bir programcının geleneksel yeteneklerini gerektirmeyen, kendi kendine öğrenme düzenekleridir. Bu ağlar öğrenmenin yanı sıra, ezberleme ve bilgiler arasında ilişkiler oluşturma yeteneğine de sahiptir (Elmas, 2003).

Yapay sinir ağları aşağıdaki varsayımlar üzerine kurulmuştur (L. Fausett, 1994).

- Bilgi işleme nöron olarak isimlendiren basit elemanlarda gerçekleştirilir
- İşaretler nöronlar arasındaki bağlantılardan geçer
- Her bağlantı birçok işareti taşıyan bir ağırlığa sahiptir
- Her nöron kendi giriş değerine çıkış işaretini belirlemek için aktivasyonfonksiyonu uygular.

YSA'nın bütün modelleri için geçerli olan genel karakteristik özellikleri vardır. Bunlar aşağıdaki gibi sıralanabilir (Öztemel, 2006):

- YSA makine öğrenmesi gerçekleştirirler. YSA'nın temel işlevi bilgisayarların öğrenmesini sağlamaktır. Olayları öğrenerek benzer olaylar karşısında benzer kararlar vermeye çalışırlar.

- Programları çalışma stili bilinen programlama yöntemlerine benzememektedir. Geleneksel programlama ve yapay zekâ yöntemlerinin uygulandığı bilgi işleme yöntemlerinden tamamen farklı bir bilgi işleme yöntemi vardır.

- YSA'da bilgi, ağın bağlantılarının değerleri ile ölçülmekte ve bağlantılarda saklanmaktadır. Diğer programlarda olduğu gibi veriler bir veri tabanında veya programın içinde gömülü değildir. Bilgiler ağın üzerinde saklı olup ortaya çıkartılması ve yorumlanması zordur.

- YSA örnekleri kullanarak öğrenirler. YSA'nın olayları öğrenebilmesi için o olay ile ilgili örneklerin belirlenmesi gerekmektedir. Örnekleri kullanarak ilgili

olay hakkında genelleme yapabilecek yeteneğe kavuşturulurlar (adaptif öğrenme). Örnek bulunamıyorsa ve yok ise yapay sinir ağının eğitilmesi mümkün değildir.

- YSA'nın güvenle çalıştırılabilmesi için önce eğitilmeleri ve performanslarının test edilmesi gerekmektedir. YSA'nın eğitilmesi demek, mevcut örneklerin tek tek ağa gösterilmesi ve ağın kendi mekanizmalarını çalıştırarak örnekteki olaylar arasındaki ilişkileri belirlemesidir.

- Görülmemiş örnekler hakkında bilgi üretebilirler. Ağ kendisine gösterilen örneklerden genellemeler yaparak görmediği örnekler hakkında bilgiler üretebilir.

- Algılamaya yönelik olaylarda kullanılabilirler.

- Şekil (örüntü) ilişkilendirme ve sınıflandırma yapabilirler. Genel olarak ağların çoğunun amacı kendisine örnekler halinde verilen örüntülerin, kendisi veya diğerleri ile ilişkilendirilmesidir. Diğer bir amaç ise sınıflandırma yapmaktır. Verilen örneklerin kümelenendirilmesi ve belirli sınıflara ayrıştırılarak daha sonra gelen bir örneğin hangi sınıfa gireceğine karar vermesi hedeflenmektedir.

- Örüntü tamamlama gerçekleştirebilirler. Bazı durumlarda ağa eksik bilgileri içeren bir örüntü veya bir şekil verilir. Ağın bu eksik bilgileri bulması istenir.

- Kendi kendini organize etme ve öğrenebilme yetenekleri vardır.

- Eksik bilgi ile çalışabilmektedirler. YSA kendileri eğitildikten sonra eksik bilgiler ile çalışabilir ve gelen yeni örneklerde eksik bilgi olmasına rağmen sonuç üretebilirler.

- Hata toleransına sahiptirler. YSA'nın eksik bilgilerle çalışabilme yetenekleri hatalara karşı toleranslı olmalarını sağlamaktadır. Ağın bazı hücrelerinin bozulması ve çalışmaz duruma düşmesi halinde ağ çalışmaya devam eder. Ağın bozuk olan hücrelerinin sorumluluklarının önemine göre ağın performansında düşmeler görülebilir.

- Belirsiz, tam olmayan bilgileri işleyebilirler. YSA'nın belirsiz bilgileri işleyebilme yetenekleri vardır. Olayları öğrendikten sonra belirsizlikler altında ağırlar öğrendikleri olaylarla ilgili ilişkileri kurarak kararlar verebilirler.

- Dereceli bozulma (Graceful degradation) gösterirler. YSA'nın hatalara karşı toleranslı olmaları bozulmalarının da dereceli (göreceli) olmasına neden olmaktadır.

- Dağıtık belleğe sahiptirler. YSA'da bilgi ağı yayılmış durumdadır. Hücrelerin birbirleri ile bağlantılarının değerleri ağı bilgisi gösterir. Tek bir bağlantının bir anlamı yoktur.

- Sadece nümerik bilgiler ile çalışabilmektedirler. Sembolik ifadeler ile gösterilen bilgilerin nümerik gösterime çevrilmeleri gerekmektedir.

2.3 YSA'nın Tarihçesi

İnsan beyni hakkındaki ilk çalışmalar çok eski zamanlara dayanmaktadır. Modern elektroniğin gelişmesi ile birlikte, bu düşünce işlemini kullanmaya çalışmak doğal hale gelmiştir. İlk YSA modeli 1943 yılında Warren McCulloch ve Walter Pitts tarafından gerçekleştirilmiştir. Bir nörobiyolojist olan Warren McCulloch ve bir istatistikçi olan Walter Pitts, "Sinir Aktivitesindeki Düşüncelere Ait Bir Mantıksal Hesap" başlıklı bir makale ile ilk dijital bilgisayarlara ışık tutmuştur. McCulloch ve Pitts, insan beyninin hesaplama yeteneğinden esinlenerek, elektrik devreleriyle basit bir sinir ağı modellemişlerdir. Yapay sinir hücreleri ile her türlü mantıksal ifadeyi formülize etmenin mümkün olduğunu göstermişlerdir. Hücrelerin birbiri ile paralel çalışması gerektiği fikrini ortaya atarak öğrenme kurallarını belirlemeye başlamışlardır.

1948 yılında Wiener "Cybernetics" isimli kitabında, sinirlerin çalışmasını ve davranış özelliklerini ele almıştır. 1949 yılında ise Donald Hebb "Organization of Behavior" isimli kitabında, öğrenme ile ilgili temel teoriyi ele almıştır.

Donald Hebb (1949) ilk olarak, sinir hücrelerinin bağlantılarını güncellemek için “Hebb öğrenme kuralı” olarak adlandırılan bir öğrenme kuralını tasarlamıştır. Hebb bilgilerin bağlantılarda depolandığını varsaydı. Bu düşünce gelecekteki gelişmeler üzerinde çok büyük bir etki yaratmıştı ve Hebb öğrenme kuralı YSA teorisinde önemli katkıda bulunmuştur (Zurada, 1995).

1950’li yıllardan sonra birçok araştırmacı Hebb kuralından esinlenerek YSA’nın hesaplama gücünü arttırıcı yönde çalışmalar yapmıştır. IBM araştırma laboratuvarında yapılan bir sinir benzetimi çalışmaları başarısızlıkla sonuçlanmasına karşın sonraki girişimlerde başarı sağlanmıştır. 1957 yılında Frank Rosentblatt’ın Perceptron’u geliştirmesinden sonra yapay sinir ağı alanındaki gelişmeler hızlanmıştır. Perceptron, beyin işlevlerini modelleyebilmek amacıyla yapılan çalışmalar neticesinde ortaya çıkan tek katmanlı eğitilebilen ve tek çıkışa sahip olan yapay sinir ağıdır.

Frank Rosentblatt (1958), bir sineğin göz işlemleri ile ilgilenmiştir. Bir sineğe kaçmasını söyleyen işlemlerin çoğu sineğin beyni yerine gözünün içinde yapılması, kavrama düğümü diye adlandırılan ağ yapısının kurulmasını sağlamıştır. Tek katmanlı bir kavrama düğümünde, giriş değerleri ağırlıklı toplamı hesaplanarak, bir eşik değeri çıkarılmakta ve iki olası değerden biri sonuç olarak alınmaktadır. Kavrama düğümü bugün kullanımda olan en eski sinir ağıdır.

1959 yılında, Bernard Widrow ve Marcian Hoff (Stanford Üniversitesinde) ADALINE ve MADALINE diye adlandırdıkları ağ modellerini geliştirdiler. MADALINE, telefon hatlarında oluşan yankıları yok eden uygulanabilir bir süzgeç olarak kullanılmış, gerçek dünya sorunlarına uygulanmış olan iki sinir ağıdır ve hala kullanımda bulunmaktadır (Elmas, 2003).

YSA konusunda ilginin artması ancak 1960’lı yıllarda bazı yeni çalışmalar yapılması ile oldu. İlk yöntemlerin karmaşık problemleri çözemeyecek kadar zayıf olması bu konudaki çalışmaların ilerlemesini engelledi. Bu durgunluk, 1970’li yılların başından 1980’li yıllara kadar devam etti.

İlk kırırdamalar Hopfield (1982) tarafından eğrisel ağların geliştirilmesiyle başladı. Bununla birlikte Kohonen (1982) ve Anderson (1983) tarafından yapılan çalışmalar sonucunda eğiticişiz öğrenen ağların geliştirilmesiyle çalışmalar yeniden ivme kazanmıştır (Şen, 2004).

1985 yılına kadar, Amerikan Ulusal Fizik Akademisi, YSA ile ilgili gelişmeleri izlemiş ve desteklemiştir.

1986 da Rumelhart ve McClelland karmaşık ve çok katmanlı ağlar için geriye yayılmalı öğrenme algoritması ortaya koymuştur.

1987 yılında Elektrik Elektronik Mühendisliği Enstitüsü (IEEE) tarafından sinir ağlarını konu alan ilk uluslar arası konferans 1800'ü aşkın katılımcıyla gerçekleşmiştir (Elmas, 2003).

Bugün YSA üzerinde yapılan çalışmalar dünyada büyük bir hızla devam etmektedir. Her geçen gün değişik öğrenme algoritmaları ve ağ mimarileri hakkında yeni önermeler yapılmaktadır (Şen, 2004).

2.4 YSA'nın Uygulama Alanları

Son yıllarda YSA, özellikle günümüze kadar çözümü güç ve karmaşık olan çok farklı alanlardaki problemlerin çözümüne uygulanmış ve genellikle başarılı sonuçlar elde edilmiştir. YSA aşağıdaki özellikleri gösteren alanlarda kullanıma uygun bir araçtır:

- Çok değişkenli problem uzayı,
- Probleme ilişkin değişkenler arasında karmaşık etkileşim,
- Çözüm uzayının bulunmaması, tek bir çözümün olması veya çok sayıda çözüm bulunması.

YSA'lar insan beyninin fonksiyonel özelliklerine benzer şekilde aşağıdaki konularda başarılı bir şekilde uygulanmaktadır (Öztemel, 2003).

- Öğrenme
- İlişkilendirme
- Sınıflandırma
- Genelleme
- Tahmin
- Özellik Belirleme
- Optimizasyon

YSA'ları çok farklı alanlara uygulanabildiğinden bütün uygulama alanlarını burada sıralamak zor olmakla birlikte genel bir sınıflandırma ile YSA'nın uygulama alanları arıza analizi ve tespiti, sağlık, savunma sanayi, haberleşme, üretim, otomasyon ve kontrol olarak altı gruba ayrılabilir.

2.4.1 Arıza Analizi ve Tespiti

Bir sistemin, cihazın ya da elemanın düzenli (doğru) çalışma şeklini öğrenen bir YSA yardımıyla bu sistemlerde meydana gelebilecek arızaların tanımlanma olanağı vardır. Bu amaçla YSA; elektrik makinelerinin, uçakların ya da bileşenlerinin, entegre devrelerin v.s. arıza analizinde kullanılmıştır.

2.4.2 Sağlık

EEG ve ECG gibi tıbbi sinyallerin analizi, kanserli hücrelerin analizi, protez tasarımı, transplantasyon zamanlarının optimizasyonu, kan hücreleri reaksiyonları ve kan analizlerinin sınıflandırılması, kalp krizlerinin önceden tespiti, örüntüleme cihazlarının ürettiği verilerden hastalıkların teşhisi ve hastanelerde giderlerin optimizasyonu, solunum hastalıklarının teşhisi, hamile kadınların karnındaki çocukların kalp atışlarının izlenmesi gibi konularda uygulanmaktadır.

2.4.3 Savunma Sanayi

Silahların otomasyonu ve hedef izleme, nesneleri/görüntüleri ayırma ve tanıma, yeni algılayıcı tasarımı ve gürültü önleme v.s gibi alanlara uygulanmıştır.

2.4.4 Haberleşme

Görüntü ve veri sıkıştırma, otomatik bilgi sunma servisleri, konuşmaların gerçek zamanda çevirisi gibi alanlarda uygulama örnekleri vardır.

2.4.5 Üretim

Üretim sistemlerinin optimizasyonu, ürün analizi ve tasarımı, ürünlerin kalite analizi ve kontrolü, planlama ve yönetim analizi gibi alanlarına uygulanmıştır.

2.4.6 Otomasyon ve Kontrol

Uçaklarda otomatik pilot sistemi otomasyonu, ulaşım araçlarında otomatik yol bulma/gösterme, robot sistemlerin kontrolü, doğrusal olmayan sistem modelleme ve kontrolü, elektrikli sürücü sistemlerin kontrolü gibi yaygın bir uygulama alanı vardır.

2.5 Sağlık Alanında YSA'nın Yeri

YSA, sistemin bağımsız değişkenleri olan girişleri, bağımlı değişkenler olan çıkışlar ile ilişkilendirerek karmaşık, doğrusal olmayan modeller oluşturur. YSA modelleme ve karar verme işlemlerinde kullanılan en önemli araçlardandır. Bunun nedeni, sınırlı ve tamamlanmamış olan veri tabanlarından en iyi sonucu çıkarabilmesi ve farklı eğitim algoritmaları ile eğitilmeleri durumunda başarının artırılabilmesidir. Bu özelliklerinin yanı sıra, YSA'nın klinik protokollerden çıkarılan veriler, ölçümlerden elde edilen laboratuvar verileri, işaretler veya görüntülerden çıkarılan öznitelikler gibi bir sistemin farklı yapıdaki verilerini birleştirerek tümleşik teşhis sistemi oluşturma özelliği vardır (Haykin, 1994).

YSA günümüzde tıbbi teşhis sistemlerini geliştirebilecek güçlü bir araçtır. Şu anda tıp alanındaki uygulamalar daha çok insan vücut parçalarının modellenmesi ve taramalardan elde edilen sonuçlardan (ultrason taraması, kardiyogram vb.)

hastalıkların tanısının konmasına yöneliktir. Hastalığa nasıl tanı koyduğuna dair özgün bir algoritmaya ihtiyaç duymayan YSA veri örneklerinden öğrenme yoluyla tanı koymaya çalışır. YSA'nın biyokimyasal analizlerde, genetik alanında yapılan çalışmalarda, kardiyoloji, gastroenteroloji, onkoloji, nöroloji, jinekoloji ve patolojide tahmin ve teşhis amaçlı kullanımı giderek yaygınlaşmaktadır (Papik, Molnar, Schaefer, Dombovari, Tulassay, Feher, 1998).

YSA, çeşitli tıbbi uygulama alanlarındaki karmaşık klinik verilerin doktorlar tarafından analizi, modellenmesi ve anlaşılması için kullanılabilecek çok güçlü araçlardır. Şimdiye kadar yapılan çalışmalar, YSA'nın sağlık hizmetlerinin verimliliğini artırarak klinik tedavi uzmanlarına yardımcı olmada hayati bir öneme sahip olduğunu göstermektedir. YSA'nın klinik tıptaki önemi, konu ile ilgili yapılmış çeşitli alanlardaki birçok yayından da anlaşılmaktadır (Demirhan, Kılıç, Güler, 2010). Sağlık alanında YSA ile yapılan bazı çalışmalar Tablo 2.1.'de belirtilmiştir.

Tablo 2.1. Sağlık Alanında YSA ile Yapılan Bazı Çalışmalar

Kaynak: Begg, R., Kamruzzaman, J. and Sarker, R. (2006), *Neural networks in healthcare : potential and challenges*, Hershey: Idea Group Inc.

Yazar	Konu	Yöntem	Sonuç
Kennedy vd. (1997)	Akut Miyokard İnfarktüsü (AMI)	ÇKA	Eğitilen YSA, AMI teşhisi için tıp doktorları kadar başarılı olmuştur.
Kukar vd. (1997)	İskemik Kalp Hastalığı (IHD)	Naive Bayes Sınıflandırıcısı Karar Ağaçları ÇKA	Makine öğrenmesi ile IHD'nin tanısı ve analizi yapılmıştır.
Delen vd. (2004)	Meme Kanseri Yaşam Süresi – Yöntemlerin Karşılaştırılması	Karar Ağacı ÇKA Lojistik Regresyon	Karar Ağacı: %93,2 ÇKA: %91,2 LR: %89,2
Ronco (1999)	Meme Kanseri Görüntüleme	ÇKA	%94,04 pozitif belirleyicilik değeri %97,6 negatif belirleyicilik değeri
Dickey vd. (2002)	Bel Ağrısı	ÇKA Doğrusal Diskriminant Analizi	ÇKA eğitim ve test seti için $R^2=0,997$ LDA eğitim seti için $r^2=0,5$, test seti için $r^2=0,14$
Nussbaum vd. (1997)	Gövde Hareketleri Similasyonu	ÇKA	Deneysel EMG verileri ile $R^2=0.4-0.9$

Wigderowitz vd. (2000)	Osteoporoz	ÇKA	%91 doğru; görüntü verileri, röntgen ve kemik yoğunluğundan üstün
Zou vd. (1996)	Psikiyatrik Tam	ÇKA Self Organizing Maps (SOM)	YSA, geleksel uzman sistemlerden daha iyi sonuçlar vermiştir.
Pesonen (1997)	Akut Apandisit	Diskriminant Analizi Lojistik Regresyon Kümeleme Analizi ÇKA	ÇKA ile elde edilen sonuç, diğer yöntemlerle elde edilen sonuçlardan daha iyidir.
Tafeit (2005)	Vücuttaki Yağ	ÇKA Nonlinear Regresyon	ÇKA ile daha iyi sonuçlar elde edilmiştir.

2.6 YSA'nın Üstünlükleri

YSA modelleri, biyolojik sinir ağlarının çalışma biçimlerinden esinlenerek ortaya çıkarılmıştır. YSA, biyolojik olmayan yapı taşlarının düzgün bir tasarımla birbirlerine yoğun olarak bağlanmalarından oluşmaktadırlar. Sinir sisteminin modellenmesi için yapılan çalışmalar sonucu oluşturulan YSA, biyolojik sinir sisteminin üstünlüklerine de sahiptir. Bu üstünlükleri şu şekillerde özetleyebilmek mümkündür (Öztemel 2003).

2.6.1 Doğrusal Olmama

YSA'nın temel işlem elemanı olan hücre doğrusal değildir. Dolayısıyla hücrelerin birleşmesinden meydana gelen YSA'da doğrusal değildir ve bu özellik bütün ağı yayılmış durumdadır. Bu özelliği ile YSA, doğrusal olmayan karmaşık problemlerin çözümünde en önemli araç olmuştur.

2.6.2 Paralellik

Alışılmış bilgi işlem yöntemlerinin çoğu seri işlemlerden oluşmaktadır. Bu da hız ve güvenilirlik sorunlarını beraberinde getirmektedir. Seri bir işlem gerçekleşirken herhangi bir birimin yavaş oluşu tüm sistemi doğruca yavaşlatırken, paralel bir sistemde yavaş bir birimin etkisi çok azdır. Nitekim seri bir bilgisayarın bir işlem elemanı beyine göre binlerce kez daha hızlı işlemesine rağmen, beynin toplam işlem hızı seri çalışan bir bilgisayara göre kıyaslanamayacak kadar yüksektir.

2.6.3 Gerçeklenme Kolaylığı

YSA'nın basit işlemler gerçekleyen türden hücrelerden oluşması ve bağlantıların düzgün olması, ağların gerçekleşmesi açısından büyük kolaylık olmasını sağlamaktadır.

2.6.4 Yerel Bilgi İşleme

YSA'da her bir işlem birimi, çözülecek problemin tümü ile ilgilenmek yerine, sadece problemin gerekli parçası ile ilgilenmektedir ve problemin bir parçası işlemektedir. Hücrelerin çok basit işlem yapmalarına rağmen, sağlanan görev paylaşımı sayesinde, çok karmaşık problemler çözülebilmektedir.

2.6.5 Hata Toleransı

Sayısal bir bilgisayarda, herhangi bir işlem elemanını yerinden almak, onu etkisiz bir makineye dönüştürmektedir. Ancak YSA'da bir elemanda meydana gelebilecek hasar çok büyük önem teşkil etmez. YSA'nın paralel çalışması hız avantajı ile birlikte yüksek hata sağlamaktadır. Seri bilgi işlem yapan bir sistemde herhangi bir birimin hatalı çalışması, hatta bozulmuş olması tüm sistemin hatalı çalışmasına veya bozulmasına sebep olacaktır. Paralel bilgi işleme yapan bir sistemde ise, sistemin ayrı ayrı işlem elemanlarında meydana gelecek olan hatalı çalışma veya hasar, sistemin performansında keskin bir düşüşe yol açmadan, performansın sadece hata birimlerinin bir oranınca düşmesine sebep olur. YSA, çok sayıda hücrenin çeşitli şekillerde bağlanmasından oluştuğundan paralel dağılmış bir yapıya sahiptir ve ağına sahip olduğu bilgi, ağıdaki bütün bağlantılar üzerine dağılmış durumdadır. Bu nedenle, eğitilmiş bir YSA'nın bazı bağlantılarının veya bazı hücrelerinin etkisiz hale gelmesi, ağına doğru bilgi üretmesini önemli ölçüde etkilemez. Bu nedenle, geleneksel yöntemlere göre hatayı tolere etme yetenekleri son derece yüksektir.

2.6.6 Öğrenebilirlik

Alışlagelmiş veri işleme yöntemlerinin çoğu programlama yolu ile hesaplamaya dayanmaktadır. Bu yöntemler ile tam tanımlı olmayan bu problemin çözümü yapılamaz. Bunun yanında, herhangi bir problemin çözümü için probleme

yönelik bir algoritmanın geliştirilmesi gerekmektedir. YSA problemleri verilen örneklerle çözer. Çözülecek problemler için yapı aynıdır. YSA'nın arzu edilen davranışı gösterebilmesi için amaca uygun olarak ayarlanması gerekir. Bu, hücreler arasında doğru bağlantıların yapılması ve bağlantıların uygun ağırlıklara sahip olması gerektiğini ifade eder. YSA' nın karmaşık yapısı nedeniyle bağlantılar ve ağırlıklar önceden ayarlı olarak verilemez ya da tasarlanamaz. Bu nedenle YSA, istenen davranışı gösterecek şekilde ilgilendiği problemten aldığı eğitim örneklerini kullanarak problemi öğrenmelidir.

2.6.7 Genelleme

YSA, ilgilendiği problemi öğrendikten sonra eğitim sırasında karşılaşmadığı test örnekleri için de arzu edilen tepkiyi üretebilir. Örneğin, karakter tanıma amacıyla eğitilmiş bir YSA, bozuk karakter girişlerinde de doğru karakterleri verebilir ya da bir sistemin eğitilmiş YSA modeli, eğitim sürecinde verilmeyen giriş sinyalleri için de sistemle aynı davranışı gösterebilir.

2.6.8 Uyarlanabilirlik

YSA, ilgilendiği problemdeki değişikliklere göre ağırlıklarını ayarlar. Yani, belirli bir problemi çözmek amacıyla eğitilen YSA, problemdeki değişimlere göre tekrar eğitilebilir, değişimler devamlı ise gerçek zamanda da eğitime devam edilebilir. Bu özelliği ile YSA, uyarlamalı örnek tanıma, sinyal işleme, sistem tanılama ve denetim gibi alanlarda etkin olarak kullanılır.

2.6.9 Donanım ve Hız

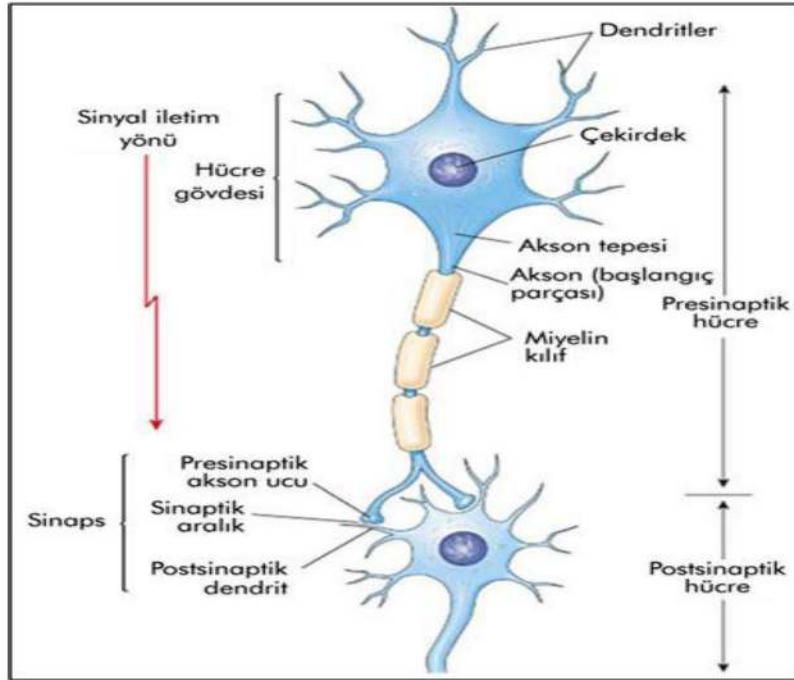
YSA, paralel yapısı nedeniyle büyük ölçekli entegre devre (VLSI) teknolojisi ile gerçekleştirilebilir. Bu özellik, YSA'nın hızlı bilgi işleme yeteneğini artırır ve gerçek zamanlı uygulamalarda arzu edilir.

2.6.10 Analiz ve Tasarım Kolaylığı

YSA' nın temel işlem elemanı olan hücrenin yapısı ve modeli, bütün YSA yapılarında yaklaşık aynıdır. Dolayısıyla, YSA'nın farklı uygulama alanlarındaki yapıları da standart yapıdaki bu hücrelerden oluşacaktır. Bu nedenle, farklı uygulama

alanlarında kullanılan YSA'ları benzer öğrenme algoritmalarını ve teorilerini paylaşabilirler. Bu özellik, problemlerin YSA ile çözümünde önemli bir kolaylık getirecektir.

2.7 Biyolojik Nöron Yapısı



Şekil 2.1. Nöron Yapısı

Kaynak: <http://www.sinancanan.net.tr/2011/03/sinirbiliminde-yeni-arayislar-1.html>

İnsan beyni, her biri basit hesaplama (işlemci gibi) kabiliyetine sahip, nöron diye adlandırılan yaklaşık 2×10^{10} sinir hücresinden oluşur. Bir sinir hücresi; dentrit, soma, akson ve snapslar olmak üzere dört kısımdan oluşur.

Dendritler diğer nöronlardan gelen sinyaller için nöronun giriş alıcıları gibi çalışırlar. Dendritin görevi diğer sinir hücrelerinden iletilen sinyalleri, sinir hücresinin çekirdeğine iletmektir. Bu yapı basit gibi görünsede günümüzde dendritlerin görevlerinin daha karmaşık olduğu yolundaki söylemler hâkim olan görüştür. Hücrenin çekirdeği ile herbir dendrit arasında farklı bir iletişim söz konusudur. Bu sebeple bazı dendritlerin etkileşimde ağırlıklı (dominant) pay sahibi, diğerlerinin de pasif (resesif) olduğu gözlenmektedir. Bu ise dışarıdan alınan

sinyallerde seçicilik gibi önemli bir olgunun sinir hücresi tarafından gerçekleştirilmesi anlamını taşımaktadır.

Soma (hücre gövdesi) ise dendritler yoluyla iletilen tüm sinyalleri alıp toplayan merkezdir. Biyolojik olarak hücre çekirdeği (nükleus) olarak da bilinen yapıdır. Çekirdek gelen toplam sinyali diğer sinir hücrelerine göndermek üzere, bilgiyi aksona iletir.

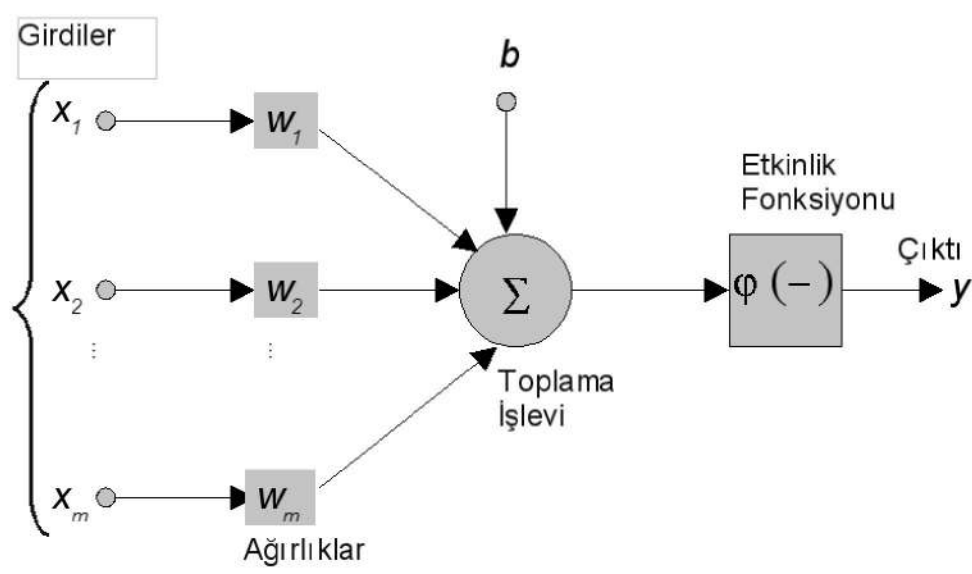
Akson nöronun çıkış kanalıdır ve sinir telleri boyunca sinir hücresinin aktif potansiyelini, diğer nöronlarla snaptik bağlantılara taşır. Akson, hücre çekirdeğinden aldığı toplam bilgiyi bir sonraki sinir hücresine dağıtmakla görevlidir. Ancak akson bu toplam sinyalin ön işlemden geçirilmeden diğer sinir hücresine aktarılmasına engel olur. Çünkü akson ucunda snapslar denilen birimlere bilgiyi aktarır.

Bir nöronun aksonik sinir telinin somaya veya diğer nöronun dendritine bağlantısı snaps olarak adlandırılır. Genellikle her bir nöronda 1000-10000 snaps vardır. Snapslar aksondan gelen toplam bilgiyi ön işlemden geçirdikten sonra diğer sinir hücrelerinin dendritlerine iletmekle görevlidir. Snapsların ön işlem ile gerçekleştirdiği görev çok önem taşımaktadır. Bu ön işlem gelen toplam sinyalin, belli bir eşik değerine göre değiştirilmesinden ibarettir. Böylece toplam sinyal olduğu gibi değil, belli bir aralığa indirgenerek diğer sinir hücrelerine iletilmiş olunur. Bu açıdan, her gelen toplam sinyal ile dendrite iletilen sinyal arasında bir korelasyon (ilişki) oluşturulur. Buradan yola çıkılarak “öğrenme” işleminin snapslarda gerçekleştiği fikri ortaya atılmış ve bu hipotez, günümüz yapay sinir ağı dünyası için teori haline dönüşmüştür. Yapay sinir ağı modelleri üzerinde “öğrenme” bu teoriye dayanılarak, snapslar ve dendritler arasında yer alan ağırlık katsayılarının güncellenmesi olarak algılanmaktadır (Schalkoff, 1992).

Biyolojik olan sinir hücresinin elemanlarından yapay sinir ağı hücre modeline geçiş yapmak gerekmektedir. Bu nedenle yapay sinir hücresi, gerçek biyolojik hücreyle aynı ilkelere dayandırılmaya çalışılmıştır (L. Fausett, 1994).

2.8 Yapay Sinir Hücresi

Biyolojik sinir ağlarında olduğu gibi yapay YSA'da da temel unsur, yapay sinir hücresidir. Yapay sinir hücresi, YSA'nın çalışmasına esas teşkil eden en küçük ve temel bilgi işleme birimidir. Ağ içinde yer alan tüm nöronlar bir veya birden fazla girdi alırlar ve tek bir çıktı verirler. Bu çıktı yapay sinir ağının dışına verilen çıktılar olabileceği gibi başka nöronlara girdi olarak da kullanılabilirler.



Şekil 2.2. Yapay Sinir Hücresinin Yapısı

Kaynak: Uğur, A., Kınacı, A.C. (2006) *Yapay Zeka Teknikleri ve Yapay Sinir Ağları Kullanılarak Web Sayfalarının Sınıflandırılması*

Her YSA hücresinin girdiler, ağırlıklar, toplama fonksiyonu, aktivasyon fonksiyonu ve eşik fonksiyonu olmak üzere beş temel elemanı vardır:

2.8.1 Girdiler

Çevreden aldığı bilgiyi yapay sinir hücresine getiren girdiler ($x_1, x_2, \dots, x_n$) kendinden önceki sinir hücresinden veya dış dünyadan gelebilir. Girdiler ağın öğrenmesi istenen örnekler tarafından belirlenir.

2.8.2 Ağırlıklar

Ağırlıklar ($W_1, W_2, \dots, W_n$) yapay sinir hücresi tarafından alınan girdilerin sinir ağı üzerindeki etkisini belirleyen uygun katsayılarıdır. Hücreye gelen bilginin önemini ve hücre üzerindeki etkisini gösterir. Her bir girdi kendine ait bir ağırlığa sahiptir. Ağırlıklar, yapay sinir ağının bilgisini depolar.

2.8.3 Toplama Fonksiyonu

Bu fonksiyon, bir hücreye gelen net girdiyi hesaplar. Bunun için değişik fonksiyonlar kullanılmaktadır. En yaygın olanı ağırlıklı toplamı bulmaktır. Burada her gelen girdi değeri kendi ağırlığı ile çarpılarak toplanır. Böylece ağıla gelen net girdi bulunmuş olur. Bu fonksiyon Eş. (2.1) ile formüle edilir.

$$NET = \sum_{i=1}^n W_i x_i \quad (2.1)$$

Burada x girdileri, W ağırlıkları, n ise hücreye gelen toplam girdi sayısını göstermektedir. Ancak YSA'da bu fonksiyonun kullanılması şart değildir. Bir problemde en uygun toplama fonksiyonunu belirlemek için bulunmuş bir formül yoktur. Genellikle deneme yanılma yolu ile toplama fonksiyonu belirlenmektedir. (Öztemel, 2006)

2.8.4 Aktivasyon Fonksiyonu

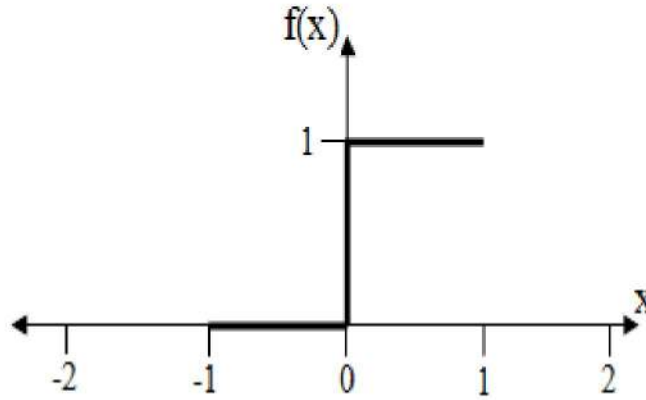
Aktivasyon fonksiyonu $f(NET)$, hücreye gelen net girdiyi işleyerek hücrenin bu girdiye karşılık üreteceği çıktıyı belirler. Toplama fonksiyonunun çıktısı aktivasyon formülünde girdi değeri olarak kullanılır. Ancak bu girdi değerlerini belirli bir seviyenin üstünde tutmak için bir eşik değeri seçilmelidir. Toplama fonksiyonundan gelen değerler bu eşik değerinden yukarıda ise işleme tabi tutulur. Aktivasyon fonksiyonu, girdileri uygulanan algoritma ile gerçek bir çıktıya dönüştürür. Aktivasyon fonksiyonunda genel olarak türevi alınabilen fonksiyonlar kullanılır.

Aktivasyon fonksiyonu ağırlıklı ortalama değerini genellikle 0–1 arasında bir çıktıya dönüştürür. Çıktının 0–1 arasındaki bir değere dönüştürülmesinin temel nedeni toplam fonksiyonu aracılığı ile elde edilen değerlerin çok büyük değerler alabilmesidir (Yıldız, 2009).

2.8.4.1 Eşik Fonksiyonu

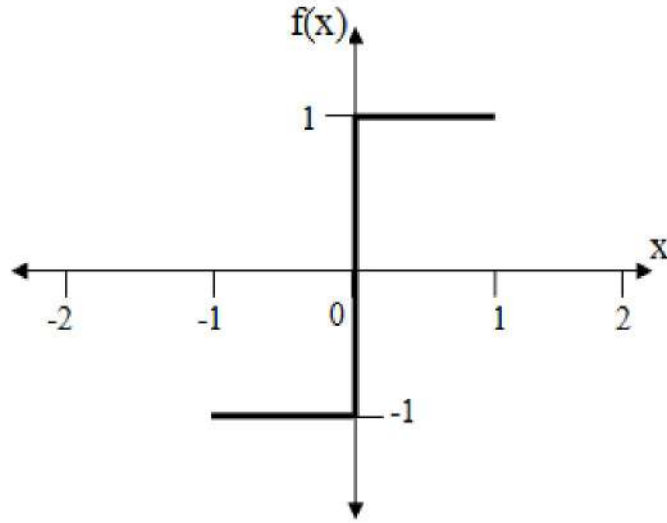
Eşik (basamak) fonksiyonunun özelliği yapay sinir hücresine gelen girdilerin ağırlıklı toplamı 0 değerine ulaşınca kadar 0 değeri, girdilerin ağırlıklı değeri belli bir değeri aştıktan sonra 1 değerini üretmesidir (Yıldız, 2009). McCulloch-Pitts modeli olarak bilinen eşik aktivasyon fonksiyonlu hücreler, mantıksal çıktı verir ve sınıflandırıcı ağlarda tercih edilir. Eşik fonksiyonu tek veya çift kutuplu fonksiyon olabilir.

$$f(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases} \quad (2.2)$$



Şekil 2.3. Tek Kutuplu Eşik Fonksiyonu

$$f(x) = \begin{cases} 1 & x \geq 0 \\ -1 & x < 0 \end{cases} \quad (2.3)$$

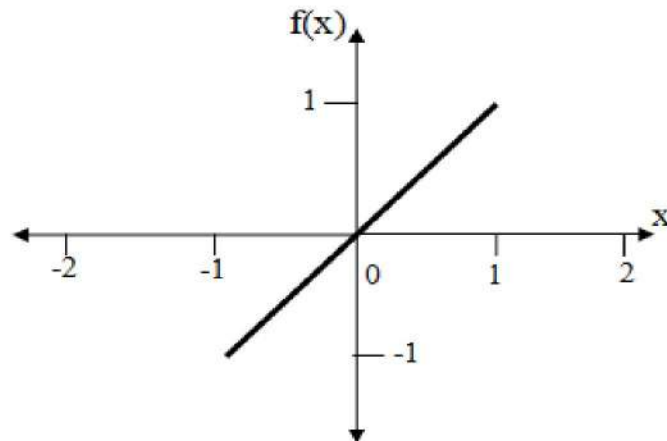


Şekil 2.4. Çift Kutuplu Eşik Fonksiyonu

2.8.4.2 Doğrusal Fonksiyon

Bazı durumlar yapay sinir hücresinde geçiş fonksiyonu olarak doğrusal fonksiyon kullanılabilir. Doğrusal fonksiyon olduğunda yapay sinir hücresine gelen girdilerin ağırlıklı değerleri aynen çıktı olarak üretilir. Doğrusal geçiş fonksiyonu doğrusal bir problem için uygun bir geçiş fonksiyonudur.

$$f(x) = x \quad (2.4)$$

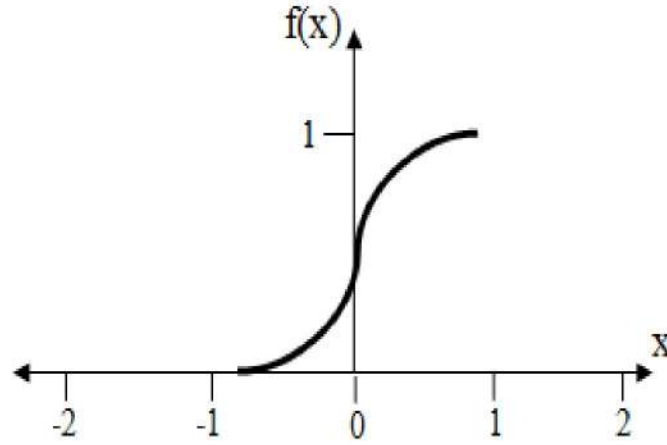


Şekil 2.5. Doğrusal Aktivasyon Fonksiyonu

2.8.4.3 Sigmoid Fonksiyon

Belki de en çok kullanılan geçiş fonksiyonlarının başında Sigmoid fonksiyonu gelir. Bunun temel nedeni sigmoid fonksiyonun doğrusal ve doğrusal olmayan fonksiyonların her ikisinin de modellenmesinde oldukça dengeli çıktılar üretmesidir (Yıldız, 2009).

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2.5)$$

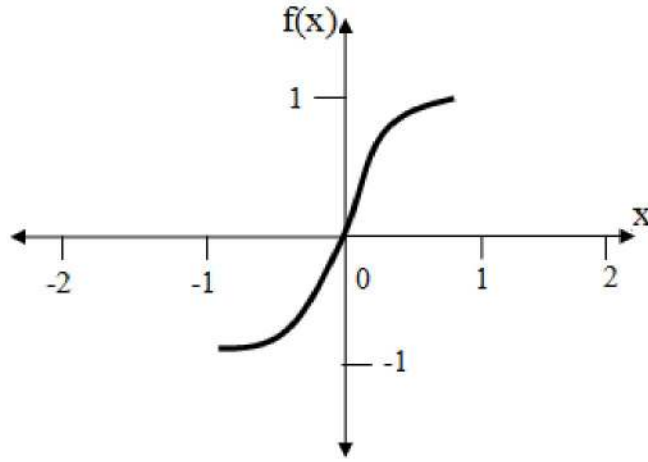


Şekil 2.6. Sigmoid Fonksiyonu

2.8.4.4 Hiperbolik Tanjant Fonksiyonu

Hiperbolik tanjant fonksiyonu, gelen net girdinin tanjant fonksiyonundan geçirilmesi ile hesaplanmaktadır ve sigmoid aktivasyon fonksiyonunun farklı bir çeşitidir. Sigmoid aktivasyon fonksiyonunda çıktı 0 ile 1 arasında bir değer alırken, hiperbolik tanjant fonksiyonunda çıktı -1 ile 1 arasındadır ve şu şekilde hesaplanır.

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.6)$$

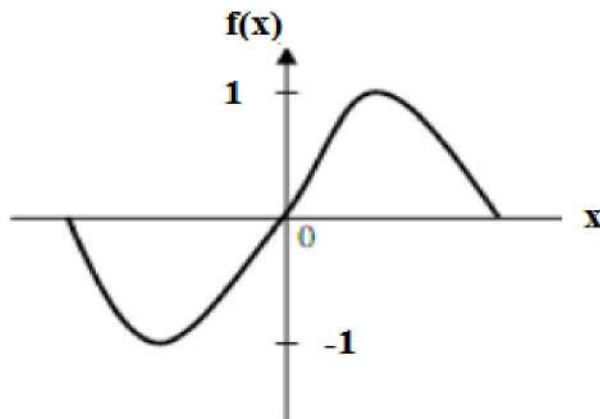


Şekil 2.7. Hiperbolik Tanjant Fonksiyonu

2.8.4.5 Sinüs fonksiyonu

Öğrenilmesi düşünülen olaylar sinüs fonksiyonuna uygun dağılım gösteriyorsa bu gibi durumlarda aktivasyon fonksiyonu olarak sinüs fonksiyonu kullanılır.

$$f(x) = \sin(x) \quad (2.7)$$



Şekil 2.8. Sinüs Fonksiyonu

2.8.5 Çıktı

Çıktılar, son süreç elemanlarıdır. Çıktı, aktivasyon fonksiyonu tarafından belirlenmektedir. Her nöron kendi çıktısını diğer bir nörona girdi olarak göndermektedir. Bir nörondan sadece bir çıktı değeri oluşmaktadır.

Bir j nöronunun eşik (bias) değeri θ_j ile gösterilirse, çıktı fonksiyonu Eş. (2.10) ile elde edilmiş olur.

$$\text{Çıktı} = f(NE_T) \quad (2.8)$$

$$NE_T = \sum_{i=1}^n W_{ij} x_i + \theta_j \quad (2.9)$$

$$\text{Çıktı} = f\left(\sum_{i=1}^n W_{ij} x_i + \theta_j\right) \quad (2.10)$$

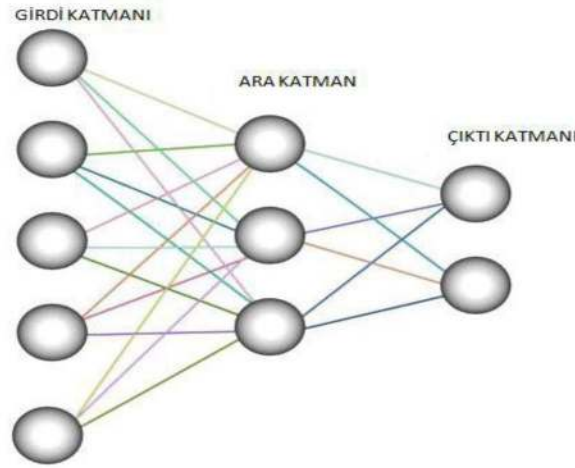
2.9 YSA'nın Yapısı

Daha önce belirtildiği gibi, yapay sinir hücreleri bir araya gelerek YSA'yı oluştururlar. Sinir hücrelerinin bir araya gelmesi rasgele olmaz. Genel olarak hücreler 3 katman halinde ve her katman içinde paralel olarak bir araya gelerek ağı oluştururlar. Bu katmanlar:

Girdi katmanı, YSA'da dışarıdan girdilerin ağı alınmasını sağlayan hücrelerin bulunduğu ilk katmandır. Bu katmandaki proses elemanları dış dünyadan bilgileri alarak ara katmanlara transfer etmekle sorumludurlar. Bazı ağlarda girdi katmanında herhangi bir bilgi işleme olmaz.

Ara katmanlar, girdi katmanından gelen bilgiler işlenerek çıktı katmanına gönderirler. Bu bilgilerin işlenmesi ara katmanlarda gerçekleştirilir. Bir ağ için birden fazla ara katmanı olabilir.

Çıktı katmanı, bu katmandaki proses elemanları ara katmandan gelen bilgileri işleyerek ağın girdi katmanından sunulan girdi seti için üretmesi gereken çıktıyı üretirler. Üretilen çıktı dış dünyaya gönderilir.



Şekil 2.9. Yapay Sinir Ağının Yapısı

2.10 YSA'da Öğrenme

YSA'da öğrenme kısaca, istenen bir işlevi yerine getirecek şekilde ağırlık değerlerinin ayarlanması sürecidir. Başlangıçta bu ağırlık değerleri rasgele olarak atanır. YSA, kendilerine örnek gösterildikçe ağırlık değerlerini değiştirirler. Amaç, ağa gösterilen örnekler için doğru çıktıları üretecek ağırlık değerlerini bulmaktır. Ağın doğru ağırlık değerlerine ulaşması, örneklerin temsil ettiği olaylar hakkında genellemeler yapabilme yeteneğine kavuşması anlamına gelmektedir. YSA'nın bilinen örneklerden belirli bilgileri çıkartarak bilinmeyen örnekler hakkında genelleme yapabilme özelliğine kavuşması işlemine, "ağın öğrenmesi" denmektedir. Öğrenme sürecinin iki aşaması vardır. Birinci aşamada ağa gösterilen örnek için ağın üreteceği çıktı belirlenir. Bu çıktı değerinin doğruluk derecesine göre ikinci aşamada ağın bağlantılarının sahip olduğu ağırlıklar değiştirilir. Ağın eğitimi tamamlandıktan sonra öğrenip öğrenmediğini (performansını) ölçmek için yapılan denemelere ise, ağın "test edilmesi" denmektedir. Test etmek için ağın öğrenme sırasında görmediği örneklerden yararlanılır. Ağ, eğitim sırasında belirlenen bağlantı ağırlıklarını kullanarak görmediği bu örnekler için çıktılar üretir. Elde edilen çıktıların doğruluk

değerleri ağıın öğrenmesi hakkında bilgiler verir. Sonuçlar ne kadar iyiye, eğitim performansının da o kadar iyi olduğunu gösterir. Eğitimde kullanılan örnek setine “eğitim seti”, test için kullanılan sete ise “test seti” adı verilmektedir (Öztemel, 2008).

Matematiksel olarak öğrenme Eş. (2.11) ile ifade edilir.

$$W_{yeni} = W_{eski} \pm \Delta W \quad (2.11)$$

Öğrenme sürecinin başlangıcında YSA ağırlıkları rastgele atanmış durumdadır. Girdiler, girdi katmanından başlayarak gizli katmanlara ve çıktı katmanına işlenerek geçirilir. Böylelikle ağırlıklar ile toplam ve aktivasyon fonksiyonlarının etkisi altında bir çıktı dizisi üretmiş olur. Bu çıktılar ile hedef çıktılar arasında hesaplanan fark “hata” olarak adlandırılır. Bu hata, ağırlıkları ile istenilen çıktılar arasındaki farkın giderilmesi için ağı içinde kullanılmaktadır.

Öğrenmenin ilk adımı aktivasyon olarak nitelendirilebilir. Sinir hücresine giren sinyallerin toplamının o hücreyi aktif hale getirebilecek bir değere sahip olup olmadığının belirlenmesi gerekir. Eğer toplam sinyal, hücreyi ateşleyebilecek eşik değerini atlatabilecek kadar yüksek ise o hücre aktiftir ($y=1$) aksi durumda o hücre pasiftir ($y=0$). Çok basit anlamda; girdi sinyallerine 1 ya da 0 cevabını vererek sınıflandırma yapabilen böyle bir hücre, hangi girdi sinyaline 1, hangi girdi sinyaline 0 diyeceği hakkında karar vermiş sayılmaktadır. “Karar vermek” ve “sınıflandırmak” öğrenme sürecinin temel yapı taşlarını oluşturmaktadır (L. Fausett, 1994).

2.10.1 Öğretmensiz öğrenme

Öğretmensiz öğrenme veya kendi kendine öğrenme (Self/Unsupervised Learning) çıktılarının istenen değerleri ağı tanıtmadan dışarıdan herhangi bir etki olmaksızın aldığı bilgileri kendi içerisinde kıyaslama yaparak sınıflandırması ile oluşan öğrenme sürecidir. Bunu yapabilmek için YSA ilk aldığı örneği (ya da örnekleri, kaç sınıfa ayırmak istiyorsa o kadar örnek alınabilir) bir sınıf olarak ilan eder ve sonradan gelecek tüm girdi sinyallerini o sınıfa benzetmeye çalışır. Bu

şekilde, tüm girdi sinyallerini kendi aralarında benzeyip benzememelerine göre ayırt eder.

Öğretmensiz öğrenme metodunda ağırlık ayarlamaları, hedef çıktılar ile mukayese edilerek yapılmazlar ve ağırlık ayarlamaları için sağlanan bir öğretme sinyali yoktur. Ağ, giriş işaretine göre kendi kendini düzenler (self-organization). Bu öğrenme metodunda ağırlıklar kabul edilebilir çıktılar ortaya çıkarmazlarsa, metot elde edilen ağırlıkları test etmez. Bu metot daha çok sınıflandırma problemleri için kullanılır.

2.10.2 Öğretmenli öğrenme

Öğretmenli öğrenmede (supervised learning), yapay sinir ağının dışarıdan etki ile eğitilmesi söz konusudur. Bu tip bir öğrenmede, girdi değerlerinin ne tür bir çıktı vermesi gerektiği önceden bilinmekte ve yapay sinir ağı ağırlıkları bu korelasyona göre güncellenmektedir. Her örnek için hem girdiler hem de o girdiler karşılığında oluşturulması gereken çıktılar sisteme gösterilirler. Bu öğrenme metodu sayesinde, mevcut hatayı en aza indirmek için bağlantılardaki ağırlıklar yeniden düzenlenir. Bu işlem kabul edilebilir bir hata seviyesine ulaşıncaya kadar devam eder. Sistemin görevi girdileri öğretmenin belirlediği çıktılara haritalamaktır. Bu sayede girdiler ve çıktılar arasındaki ilişkiler öğrenilmektedir.

2.10.3 Takviyeli öğrenme

Takviyeli öğrenmede öğrenen sisteme bir öğretmen yardımcı olur. Fakat öğretmen her girdi seti için üretilmesi gereken çıktı setini sisteme göstermek yerine sistemin kendisine gösterilen girdilere karşılık çıktısını üretmesini bekler ve üretilen çıktının doğru veya yanlış olduğunu gösteren bir sinyal üretir. Sistem öğretmenden gelen bu sinyali dikkate alarak öğrenme sürecini devam ettirir.

2.11 YSA'nın Sınıflandırılması

Hücrelerin bağlantı şekillerine, öğrenme kurallarına ve aktivasyon fonksiyonlarına göre çeşitli YSA yapıları geliştirilmiştir. Her bir sinir hücresi arasındaki bağlantıların yapısı ağı yapısını belirler. İstenilen hedefe ulaşmak için

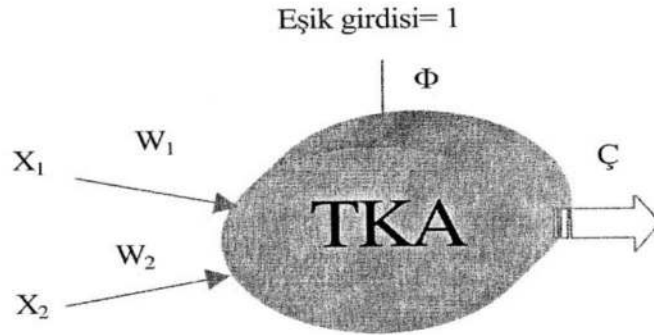
bağlantıların nasıl değiştirileceği öğrenme algoritması tarafından belirlenir. Kullanılan bir öğrenme kuralına göre, hatayı sıfıra yakınsayacak şekilde, ağırlıklar değiştirilir. YSA yapılarına ve öğrenme algoritmalarına göre sınıflandırılırlar.

2.11.1 YSA'nın Yapısındaki Katmanlara Göre Sınıflandırılması

YSA, içerdikleri katman sayısına göre tek katmanlı ve çok katmanlı olarak sınıflandırılabilirler.

2.11.1.1 Tek Katmanlı Ağlar

Tek katmanlı yapay sinir ağları (TKA) sadece girdi ve çıktı katmanlarından oluşur. Her ağın bir veya daha fazla girdisi ve çıktısı vardır. Çıktı üniteleri bütün girdi ünitelerine bağlanmaktadır. Her bağlantının bir ağırlığı vardır. Tek katmanlı algılayıcılarda çıktı fonksiyonu doğrusal bir fonksiyondur. Diğer bir deyişle ağa gönderilen örnekler iki sınıf arasında paylaştırılarak iki sınıfı birbirinden ayıran doğru bulunmaya çalışılır.



Şekil 2.10. İki Girdi ve Bir Çıktıdan Oluşan En Basit TKA Modeli

Kaynak: Öztemel, E. (2003), *Yapay Sinir Ağları*, İstanbul: Papatya Yayıncılık.

Ağın çıktısı ağırlıklandırılmış girdi değerlerinin eşik değeri ile toplanması sonucu bulunur. Bu girdi değeri bir aktivasyon fonksiyondan geçirilerek ağın çıktısı hesaplanır. Bu çıktı, Eş. (2.12) ile formülize edilmektedir (Öztemel, 2008).

$$C = f \left(\sum_{i=1}^m W_i X_i + \phi \right) \quad (2.12)$$

TKA temel olarak doğrusal özellik gösteren olayları çözmekte başarılı iken doğrusal özellik göstermeyen olayların çözülmesinde yetersiz kalmaktadırlar.

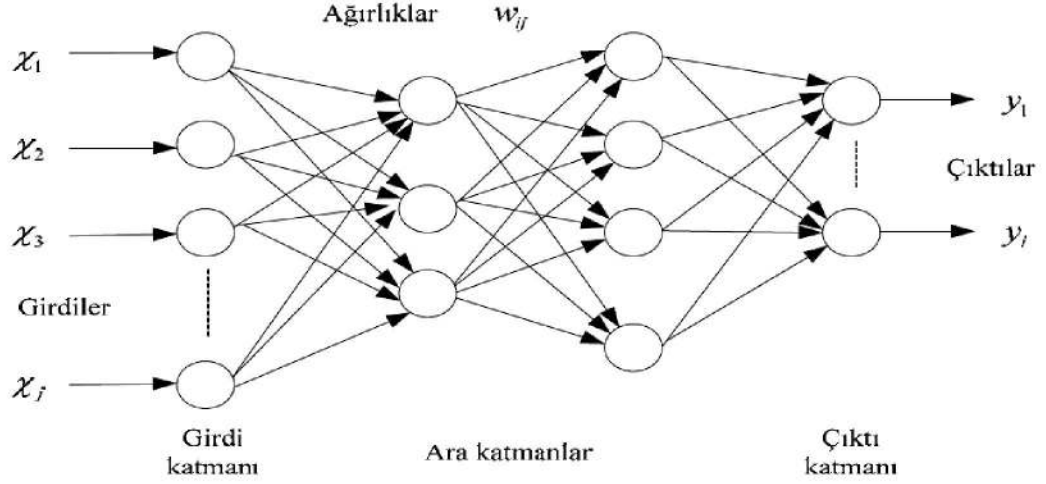
2.11.1.2 Çok katmanlı ağlar

Günümüzde en çok bilinen ve yaygın biçimde kullanılan yapay sinir ağı *Çok Katmanlı Yapay Sinir Ağları* (ÇKA) yöntemidir. Birçok öğretim algoritmasının bu ağı eğitmede kullanılabilir olması, bu modelin yaygın kullanılmasının sebebidir. Genel amaçlı, esnek ve çoklu katmanlar halinde organize edilmiş nöronlardan oluşan doğrusal olmayan modellerdir. ÇKA hemen hemen her fonksiyonu tahmin etme özelliğine sahip bir ağıdır ve TKA'nın genelleştirilmiş halidir. Özellikle tanıma sistemlerinde kullanılmaktadır.

ÇKA, girdi katmanı, bir veya birden fazla gizli katman ve çıktı katmanından oluşmaktadır. Bir katmandaki her nöron, bir sonraki katmanın bütün nöronları ile bağlantılıdır. Aynı katmandaki nöronlar arasında veya geri-besleme şeklinde bağlantıları yoktur. Alınan girdiler, bağlantı ağırlıkları ile çarpılıp gizli katmana iletilmektedir. Gizli katmandaki nöronlara gelen girdiler toplanarak aynı şekilde gizli katman ile çıktı katmanı arasındaki bağlantı ağırlıkları ile çarpılarak çıktı katmanına iletilir. Çıktı katmanındaki nöronlar da kendisine gelen girdileri toplayarak buna uygun bir çıktı üretirler. Burada girdi katmanından çıktı katmanına doğru, gizli katmanlar üzerinden tek yönlü bir iletişim mevcuttur.

ÇKA, öğretmenli öğrenme stratejisine göre çalışır. Eğitim sırasında ağa, hem girdiler hem de bu girdilere karşılık gelen çıktı değerleri gösterilir. Ağın görevi bu girdiye karşılık gelen çıktı değerini öğrenmektir. ÇKA'nın öğrenme kuralı en küçük kareler yöntemine dayalı Delta Öğrenme Kuralının genelleştirilmiş halidir. Genelleştirilmiş Delta Kuralı (Backpropagation) iki safhadan oluşur. 1. safhaya, ileri doğru hesaplama safhası adı verilir ve bu safhada ağa gösterilen örnek seti için ağın çıktısı hesaplanır.

2. safhaya geriye doğru hesaplama safhası adı verilir; bu safhada ise ağırlıklar güncellenir.



Şekil 2.11. İki Gizli Katmana Sahip ÇKA

Bir ÇKA'nın ayırt edici üç önemli özelliği vardır (Haykin, 1994).

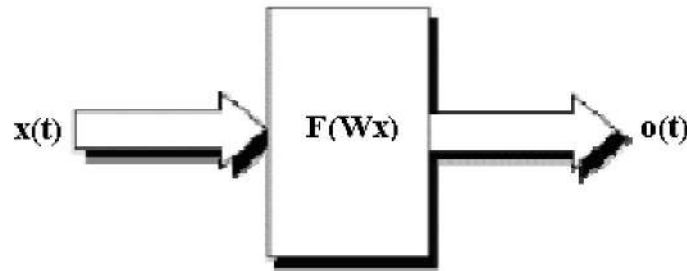
1. Ağdaki her bir nöron doğrusal olmama özelliği içerir. Doğrusal olmama biçimi genellikle kullanılan sigmoid (lojistik) fonksiyonu ile sağlanır.
2. ÇKA, gizli nöronlardan oluşan bir ya da daha fazla gizli katmana sahiptir. Gizli nöronlar, girdi verisi yapısındaki karmaşık yapıyı öğrenmede ağı başarılı kılarlar.
3. Ağ bağlantıları sayesinde çok yüksek derecede bilgi işleme becerisi gösterir. Ağı bilgi işleminde değişiklik olabilmesi, bağlantı sayısında ya da ağırlıklarında değişikliği gerektirir.

2.11.2 YSA'nın Mimari Yapılarına Göre Sınıflandırılması

Genel olarak YSA'yı mimari yapılarına göre, ileri beslemeli ve geri beslemeli ağlar olmak üzere sınıflandırılırlar.

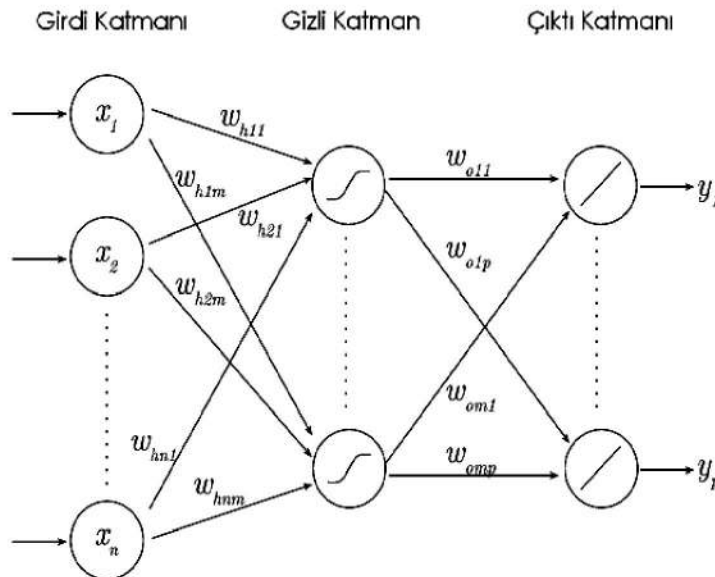
2.11.2.1 İleri beslemeli yapay sinir ağı

İleri Beslemeli Yapay Sinir Ağları (İBYSA)'nda işlemci elemanlar genellikle katmanlara ayrılmışlardır ve bir katmandaki hücrelerin çıktıları bir sonraki katmana ağırlıklar üzerinden girdi olarak verilir. İBYSA'da katmanlar ileri yöndedir, tersine bir yönelme yoktur. Her bir katmandaki hücreler sadece bir önceki katmanın hücrelerince beslenir (Karlık, 1994).



Şekil 2.12. İleri Beslemeli Ağ için Blok Diyagramı

İBYSA nöron adı verilen çok katmanlı karar verici nöronları kapsar. İlk katman girdi katmanıyken son katman ise çıktı katmanıdır. Girdi ve çıktı katmanları dış çevreyle bağlantılı olan katmanlardır. En az bir gizli katman ortada bulunmaktadır. İBYSA'da her bir girdi nöronu ilk gizli katmandaki her nörona bağlıdır ve her bir katmanın çıkışı bir sonraki katmanın girdisidir (Auger, 2001).



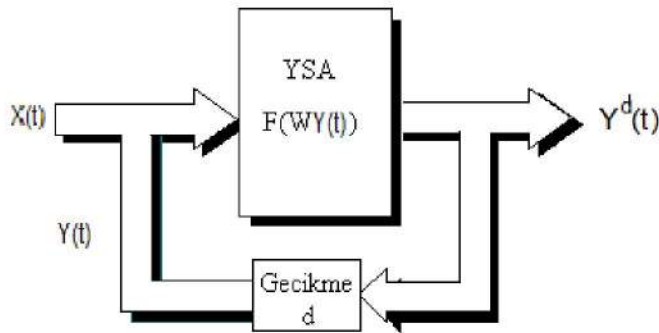
Şekil 2.13. İleri Beslemeli 3 Katmanlı YSA

Gizli katman ve katmanlardaki nöron sayısı hatayı azaltacak doğrultuda denemeler ile bulunur. Gizli katman sayısı az seçildiği zaman bilgiyi modelleyemez. Çok sayıda seçilirse kuralları genelleştirmek için ağın kabiliyetini kısıtlar ve model yeni gelen veriler için iyi çalışmaz. Bir önceki katmandaki nöronlar bir sonraki katman nöronlarına bağlanmışlardır. Bu ağlarda bilgi girdi katmanından çıktı katmanına doğru yayılır. Geri besleme yoktur ve sadece bitişik katman işlemcileri arasında bağlantılar vardır.

Bu tip YSA'nın eğitiminde genellikle *Geri Yayılım Öğrenme Algoritması* (GYA) kullanılmaktadır. Şekil tanıma, sinyal işleme ve sınıflandırma gibi problemlerde genellikle bu topoloji uygulanmaktadır.

2.11.2.2 Geri Beslemeli Yapay Sinir Ağları

Bir *geri beslemeli yapay sinir ağı* (GBYSA), çıktı ve ara katmanlardaki çıktıların, girdi birimlerine veya önceki ara katmanlara geri beslendiği bir ağ yapısıdır. Böylece, girdiler hem ileri yönde hem de geri yönde aktarılmış olur. GBYSA'nın dinamik hafızaları vardır ve bir andaki çıktı hem o andaki hem de önceki girdileri yansıtır. Bu ağlar çeşitli tipteki problemlerin tahmininde oldukça başarı sağlamışlardır. Bu ağlara örnek olarak; Hopfield, Düzenleyici Harita (Self Organizing Map – SOM), Elman ve Jordan ağları verilebilir (Alavala, 2002).



Şekil 2.14. Geri Beslemeli Ağ için Blok Diyagram

Burada;

$X(t)$: t nöronlu girdi katmanı,

$Y(t)$: t nöronlu girdi katmanına ait gizli katman,

d : gecikme sayısı,

$Y_d(t)$: d gecikmeli t nöronlu çıktılar,

$F(W Y(t))$: Ağın birimlerinde tercih edilen aktivasyon fonksiyonudur.

2.12 YSA Tasarımı

2.12.1 Ağ Yapısı

Nöronların bağlantı şekilleri, oluşturdukları katman sayısı ve katmanlar arasındaki veri iletim şekli ağın yapısını belirler. Bir ağın yapısı, ağın fonksiyonelliği ve performansı üzerinde önemli bir rol oynar. Ağ seçimi, çözülmek istenen problemle ilgilidir. Hangi ağ modelinin hangi problemin çözümü için daha uygun olduğunun bilinmesi oldukça önemlidir. YSA ağda kullanılan yapıya göre isimlendirilirler. Yapılar, ağda kullanılan katman sayısı (tek katmanlı, çok katmanlı), öğrenme algoritması (danışmanlı, danışmansız, takviyeli), öğrenme kuralı, iletişim yönü (ileri beslemeli, geri beslemeli, yarışmacı) gibi belirleyici özellikleri ile isimlenirler (Bayır, 2006).

Bir ağda, bütün girdi nöronları girdi katmanında, bütün çıktı nöronları çıktı katmanında ve gizli nöronlarda bir veya daha fazla katmana dağıtılmıştır. Tahmin için kurulan ÇKA'nın tasarımında aşağıdaki değişkenlerin belirlenmesi gerekir.

1. Girdi nöronu sayısı
2. Gizli katman ve gizli nöron sayısı
3. Çıktı nöronu sayısı

Girdi nöronları, girdi veri gruplarının ağa sunulduğu terminallerdir. Bu tabakadaki nöron sayısı, giriş veri sayısı kadardır ve her bir giriş nöronu bir veri alır. Burada veri işlemeden bir sonraki tabakaya yani gizli katmana geçer.

Zhang (1998), birçok tahmin problemi için tek gizli katmanın yeterli olacağını ifade etmiştir. Ancak, bazı özel problemlerde, iki gizli katmanlı ağ

yapısının tek gizli katmanlı ağ yapısından daha iyi sonuç vermesi muhtemeldir. Kaastra ve Boyd (1996), tek gizli katmanın yeterli olacağını, nadir de olsa iki gizli katman kullanılabileceğini fakat iki gizli katmandan fazla gizli katmana ihtiyaç olmadığını ifade etmektedir.

Gizli katmanlardaki nöron sayısının tespitinde sanal bir kural yoktur. Az sayıda gizli nöron veri yapısındaki ilişkiyi öğrenmede başarısız olabilir, çok sayıda gizli nöron ise ağıın öğrenmesi yerine ezberlemesine sebep olur. Literatürde, yapılan deneysel çalışmaların sonucunda, “n” girdi nöronu ve “m” çıktı nöronu sayısını göstermek üzere, tek gizli katmanlı ağlarda gizli nöron sayısına ilişkin bazı öneriler şunlardır (Hamzaçebi, 2011)

- a) n (Tang ve Fishwick, 1993)
- b) $2n + 1$ (Lippmann, 1987)
- c) $2n$ (Wong, 1991)
- d) $\sqrt{n * m}$ (Masters, 1993)
- e) $0.75 * n$ (Baily ve Thompson, 1990)

Çıktı nöronları, ağıın en uç tabakasıdır. Gizli katmandan aldığı veriyi ağıın kullandığı fonksiyonla işleyerek çıktısını verir. Çıkış katmanındaki nöron sayısı, ağa sunulan her verinin çıkış sayısı kadardır. Bu tabakadan elde edilen değerler yapay sinir ağının söz konusu problem için çıkış değerleridir. İleri besleme safhasında, giriş tabakasındaki nöronlar, veri değerlerini doğrudan gizli katmana iletir. Gizli katmandaki her bir nöron kendi giriş değerlerini ağırlıklandırarak toplam değer hesap ederler ve bunları bir taşıma fonksiyonu ile işleyerek bir ileri tabakaya veya doğrudan çıkış tabakasına iletirler. Tabakalar arasındaki ağırlıklar rasgele küçük rakamlardan seçilir.

2.12.2 Başlangıç Değerlerini Belirleme

Bağılantı ağırlıklarının başlangıç değerleri ağıın hata fonksiyonu için genel minimumu yakalama gücünü etkiler. İki nöron arasındaki bağlantı ağırlığının

değişimi her iki nöronun aktivasyon fonksiyonunun türevi ile yakından ilgilidir. Bu sebeple aktivasyon fonksiyonu çıktısını ya da türevinin sonucunu sıfır yapacak başlangıç ağırlık değerlerinden kaçınmak gerekir. Başlangıç ağırlık değerleri ne çok büyük ne de çok küçük değer olmalıdır. Bir genel yaklaşım başlangıç ağırlık değerlerini $[-0.5 \ 0.5]$ ya da $[-1 \ 1]$ aralığından atamaktadır (Hamzaçebi, 2011).

2.12.3 Eğitim Kümesinin Büyüklüğü

YSA'nın eğitimi için gerekli olan eğitim verisi miktarı ile ilgili Eş. (2.13), test kümesi için istenilen doğruluk derecesi ve kurulan ağırlık yapısına göre eğitim kümesindeki veri sayısının ne olması gerektiğine ilişkin bir yaklaşım sunmaktadır.

$$e = \frac{s(W)}{s(S)} \quad (2.13)$$

Burada,

e : hata oranı

$s(W)$: bağlantı ağırlıklarının sayısı

$s(S)$: eğitim kümesindeki veri sayısı

olarak tanımlanmıştır.

2.12.4 Öğrenme Parametreleri

GYA' da ağırlıkların değişmesinde iki parametre önemli rol oynar. Bu parametreler öğrenme oranı ve momentum katsayısıdır. Öğrenme oranı, bağlantıların ağırlık değerlerindeki değişim miktarını belirler. Öğrenme oranı genellikle 0 ile 1 arasında bir değerdir. Öğrenme oranının büyük değeri değişim miktarını artırırken, küçük değerde değişim miktarını azaltacaktır. Öğrenme oranının çok büyük değeri kararsızlığa, çok küçük değeri de öğrenme sürecinin kabul edilemeyecek şekilde yavaşlamasına sebep olur. Öğrenme oranının çok düşük olması da genel minimumu bulmaya imkân vermeyebilir (Kaastra ve Boyd, 1996). Tang (1991), az karmaşık veri yapısında yüksek öğrenme oranının iyi olduğunu ifade etmektedir.

Momentum parametresi, büyük öğrenme oranıyla eğitim sürecinin hızlanmasını sağlarken, salınma eğilimini de en aza indirmeye yardımcı olur. Momentum katsayısı ağın toparlanmasına yardım eden bir faktördür. Daha önceki değişimin bir kısmını o andaki değişime eklemek şeklinde ifade edilebilir. Bu durum veri kümesindeki hatalı verilerden dolayı öğrenme işleminin yanlış yönde ilerlemesine engel olmayı sağlamaktadır. 0 ile 1 arasında değerler alır (Zhang, Patuwo, Hu,1998).

2.12.5 Aktivasyon Fonksiyonu

Aktivasyon fonksiyonu, transfer fonksiyonu olarak da adlandırılmaktadır. Bir nöronun veya ağı girdisi ve çıktısı arasındaki ilişkiyi belirlemektedir. Uygulamada, sınırlı, tekdüze artan ve türevi alınabilen aktivasyon fonksiyonları kullanılmaktadır. Genel olarak, bir ağı, aynı ya da farklı katmanlarındaki nöronlar farklı aktivasyon fonksiyonunu kullanabilirler. Örneğin; her nöronun aktivasyon fonksiyonu ayrı ayrı belirlenebileceği gibi, kullanılacak aktivasyon fonksiyonu katman katman da belirlenebilir. 1. gizli katmanda sigmoid, 2. gizli katmanda doğrusal, çıktı katmanında hiperbolik tanjant tipli aktivasyon fonksiyonları kullanılabilir (Çelik, 2008). Ancak uygulamaların çoğunda aynı katmandaki nöronların aynı aktivasyon fonksiyonunu kullandıkları görülmektedir. Uygulamada kullanılan aktivasyon fonksiyonları genelde şunlardır:

1. Sigmoid (Lojistik) Fonksiyonu
2. Hiperbolik Tanjant Fonksiyonu
3. Doğrusal Fonksiyon

2.12.6 Veri Normalleştirme

Hedef değerlerin ağ çıktısı ile aynı aralıkta olacak şekilde dönüşüme tabi tutulması gerekir. Sigmoid ya da hiperbolik tanjant gibi doğrusal olmayan aktivasyon fonksiyonları, bir nöronun çıktısını (0,1) veya (-1,1) aralığına sıkıştırırlar. Bu sebeple hesaplama hatalarından kaçınmak için hem çıktıları hem de girdileri normalleştirme avantajlı olacaktır. Veri normalleştirme eğitim sürecine başlamadan uygulanır. Veri

normalleştirme yaklaşımlarında, sıklıkla aşağıdaki formüller kullanılmaktadır (Hamzaçebi, 2011).

a) $[0, 1]$ aralığında doğrusal dönüşüm:

$$x_n = \frac{x_0 - x_{min}}{x_{maks} - x_{min}} \quad (2.14)$$

b) $[a, b]$ aralığında doğrusal dönüşüm:

$$x_n = (b - a) \frac{x_0 - x_{min}}{x_{maks} - x_{min}} + a \quad (2.15)$$

c) Basit normalizasyon:

$$x_n = \frac{x_0}{x_{maks}} \quad (2.16)$$

d) İstatistiksel normalizasyon:

$$x_n = \frac{x_0 - \bar{x}}{s} \quad (2.17)$$

x_n ve x_0 , normalleştirilmiş ve orijinal veriyi temsil etmektedir. x_{min} , x_{maks} , $\bar{x}$ ve s sırasıyla minimum, maksimum, ortalama ve standart sapmayı ifade etmektedir.

2.12.7 Öğrenme Algoritması

Ağ eğitimi için birçok değişik optimizasyon metodu vardır. Seçilen ağ yapısına uygun olan öğrenme algoritmasını seçmek gerekmektedir. YSA'nın az veriyle öğrenmesi ve genelleme yapabilmesi istenir. Ağın genellemesi yapılan testlerle kontrol edilmelidir. Aşırı öğrenme söz konusu ise ağ kabul edilebilir çıkışlar

üretmez. En yaygın kullanılan öğrenme (eğitim) algoritması esasında bir gradyan dik iniş algoritması olan GYA' dır.

2.12.8 Veri Kümesinin Düzenlenmesi

YSA kullanarak tahminde bulunurken veri kümesi eğitim-test ya da eğitim-doğrulama-test olmak üzere kısımlara ayrılmalıdır. Eğitim kümesi verileri kurulan ağı öğrenmesini, doğrulama kümesi verileri eğitim sürecinin durdurulacağı zamanın belirlenmesi ve test kümesi verileride ağı sonuçlarının genelleştirilebilmesi için kullanılır. Literatürde eğitim, doğrulama ve test kümelerinin belirlenmesine yönelik az da olsa öneriler vardır. Birçok araştırmacı %80, %10, %10 veya %70, %15, %15 ya da %60, %20, %20 kurallarını temel alan yöntemler izlemişlerdir. Eğitim-test kümesi için oransal dağılım olarakta %80, %20 veya %90, %10 şeklinde yöntemler kullanılmıştır.

2.12.9 Performans fonksiyonu seçimi

Tahmin hatası, gerçek gözlem değerleri ile tahmin edilen değer arasındaki farktır. Tahmin hatası:

$$e = Y - F \quad (2.18)$$

eşitliği ile hesaplanır. F modelin ürettiği değer, Y ise gerçek gözlem değeridir.

İleri beslemeli ağlar için kullanılan performans fonksiyonlarından sık kullanılan üçü olan hata kareler ortalaması (MSE), hata kareler toplamı (SSE) ve MSE'nin karekökü (RMSE) sırasıyla aşağıda ifade edilmiştir.

$$MSE = \frac{1}{n} \sum_{i=1}^n (d_i - y_i)^2 \quad (2.19)$$

$$SSE = \sum_{i=1}^n (d_i - y_i)^2 \quad (2.20)$$

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (d_i - y_i)^2} \quad (2.21)$$

BÖLÜM 3

3. GERİ YAYILIM(BACKPROPAGATION) YAPAY SİNİR AĞLARI

3.1 Giriş

Tek katmanlı sinir ağlarının sınırlamalarının gösterilmesi, 1970' li yıllarda sinir ağlarındaki ilginin azalmasında önemli bir faktördü. ÇKA eğitiminin etkili bir genel yönteminin keşfi, çok çeşitli türlerdeki problemlerin çözülmesi için bir araç olarak YSA'nın yeniden ortaya çıkmasında önemli bir rol oynamıştır. Geri yayılım ya da genelleştirilmiş delta kuralı olarak bilinen bu yöntem ağ tarafından hesaplanan çıktının toplam karesel hatasını en aza indirmek amaçlı olan gradyan iniş (gradient descent) metodudur. Bir sinir ağı, ağın mimarisi, bir nörondaki aktivasyon fonksiyonu ve eğitim algoritması ile karakterize edilir. ÇKA'nın eğitimi için genellikle kullanılan eğitim algoritması hatayı geriye yayma (Back Propagation) algoritmasıdır (Arslan ve İnce, 1995). GYA, basitliği ve uygulamadaki görüş açısı gibi başarılarından dolayı ağ eğitimi için en popüler algoritmalarından biridir (Aktaş, Okumuş, 2003). Bu algoritma; hataları geriye doğru çıkıştan girişe azaltmaya çalışmasından dolayı geri yayılım ismini almıştır. Geri yayımlı öğrenme kuralı ağ çıkışındaki mevcut hata düzeyine göre her bir tabakadaki ağırlıkları yeniden hesaplamak için kullanılmaktadır. Bir geri yayımlı ağ modelinde girdi, gizli ve çıktı olmak üzere 3 katman bulunmakla birlikte, problemin özelliklerine göre gizli katman sayısını artırabilmek mümkündür.

GYA'nın eğitimi üç aşamadan oluşur: girdi eğitim örüntüsünün ileri beslemesi, hesaplama ve hatanın geri yayılımı ve ağırlıkların güncellenmesi. Eğitimden sonra, ağ uygulaması sadece geri besleme evresinin hesaplanmasından oluşur. Eğitim yavaş olsa bile, eğitilmiş bir ağ çıktısını çok hızlı bir şekilde üretebilir. Sayısız geri yayılım çeşidi eğitim sürecinin hızını arttırmak için geliştirilmiştir ve geliştirilmeye devam edilmektedir.

Tek katmanlı bir ağ, ciddi bir şekilde öğrenebileceği eşlemelerde sınırlandırılmış olmasına rağmen, çok katmanlı bir ağ (bir veya birden fazla gizli katmanla) istenen bir doğruluğu, herhangi bir süreklilikte eşlemeyi öğrenebilir.

3.2 Öğrenme Oranı, Momentum Katsayısı ve Bias

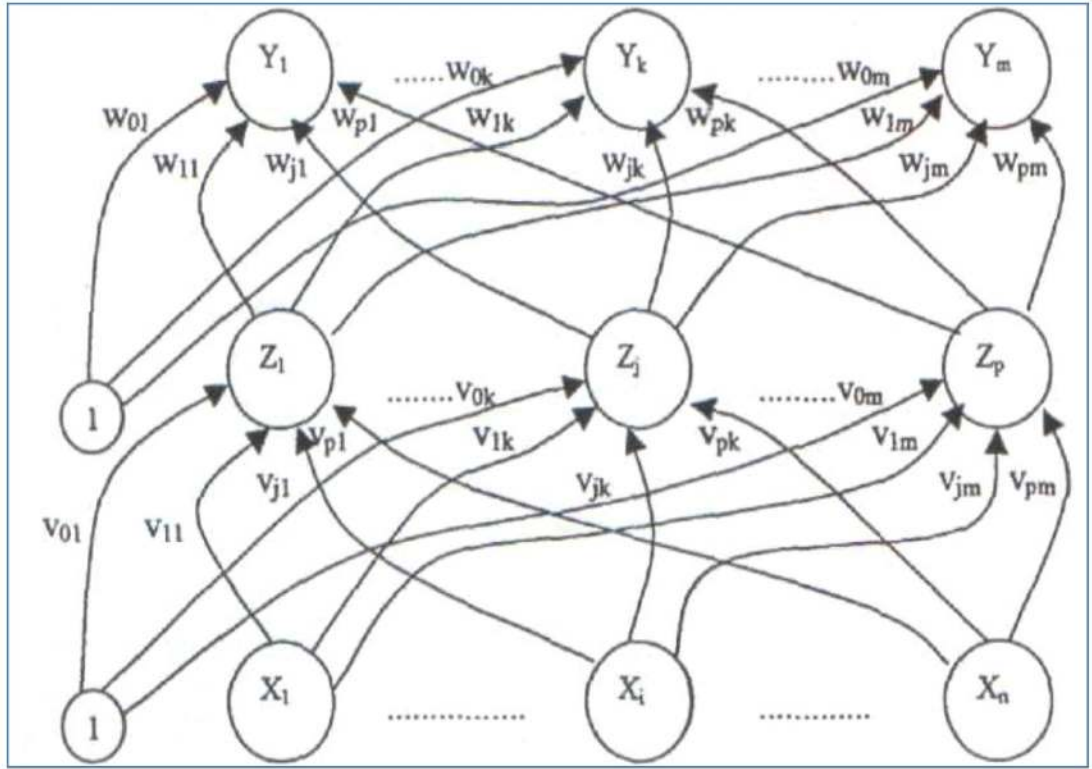
Öğrenme oranı, bağlantı ağırlıklarındaki değişme oranını belirler ve 0-1 arasında değerler alır. Öğrenme oranı büyüdükçe değişim miktarı artmakta ve çok büyük değerleri kararsızlığa neden olmaktadır. Öğrenme oranının çok küçük değerleri ise öğrenme işleminin yavaşlamasına yol açmaktadır.

Momentum katsayısı, ağı daha hızlı toparlanmasına yardım eden ve bir önceki değişimin bir kısmını bir sonraki değişime ekleyerek ağırlıkları yenilemek için kullanılan bir faktör şeklinde ifade edilebilir. Momentum katsayısı, ağı öğrenme esnasında yerel minimum değerine takılmasını yani ağı salınımını engellemektedir. 0-1 arasında değerler alabilen momentum katsayısının yüksek değerleri çoğunlukla öğrenme işlemini hızlandıracağından daha uygundur.

Bias sabit aktivasyon değeriyle özel işlem elmanıdır. Girdi vektörünün tüm elamanları 0 iken öğrenme gerçekleşmez. Bu sorunu gidermek için her katmana değeri 1 olan bias nöronları yerleştirilir. Bir geri yayılım ağında bias girdileri kullanılması daha iyi öğrenmeye neden olacaktır.

3.3 Standart Geri Yayılım Algoritması

Daha önce belirtildiği gibi GYA'nın eğitimi üç aşamadan oluşur: girdi eğitim örüntüsünün ileri beslemesi, hesaplama ve hatanın geri yayılımı, ağırlıkların güncellenmesi.



Şekil 3.1. Bir Gizli Tabakalı Geri Yayımlı Ağ

Kaynak: Fausett, L. (1994). *Fundamentals of Neural Networks: Architecture, Algorithm and Applications*, New Jersey: Printice Hall.

x : Girdi eğitim vektörü,

$$x = (x_1, \dots, x_i, \dots, x_n) .$$

t : Çıktı hedef vektörü,

$$t = (t_1, \dots, t_k, \dots, t_m) .$$

δ_k : Çıktı ünitesi Y_k 'da bir hata nedeniyle w_{jk} 'nın ağırlık güncellemesi için hata düzeltme payı.

δ_j : Çıktı tabakasından gizli ünite Z_j 'ye hata bilgisinin geri yayılımı nedeniyle v_{ij} 'nin ağırlık güncellemesi için hata düzeltme payı.

α : Öğrenme oranı

X_i : Girdi ünitesi i ,

Bir girdi ünitesi için, girdi sinyali ve çıktı sinyali aynıdır (yani x_i).

v_{oj} : Gizli ünite j ' de bias.

Z_j : Gizli ünite j ,

Z_j için net girdi z_in_j ile ifade edilirse,

$$z_in_j = v_{oj} + \sum_i x_i v_{ij} \quad (3.1)$$

Z_j 'nin çıktı sinyali z_j ile ifade edilirse,

$$z_j = f(z_in_j) \quad (3.2)$$

w_{ok} : Çıktı ünitesi k 'da bias

Y_k : Çıktı ünitesi k ,

Y_k için net girdi y_in_k ile ifade edilirse,

$$y_in_k = w_{ok} + \sum_j z_j w_{jk} , \quad (3.3)$$

Y_k 'nın çıktı sinyali y_k ile ifade edilirse,

$$y_k = f(y_in_k) \quad (3.4)$$

İleri besleme sırasında herbir girdi ünitesi (X_i), bir girdi sinyali alır ve bu sinyali gizli ünitelerin ($Z_1, \dots, Z_p$) herbirine iletir. Her gizli ünite aktivasyonunu hesaplar ve sinyallerini (z_j) herbir çıktı ünitesine gönderir. Her çıktı ünitesi (Y_k) verilen girdi örüntüsü için ağ çıktısı oluşturacak olan aktivasyonunu (y_k) hesaplar (L. Fausett, 1994).

Eğitim sürecinde her çıktı ünitesi, o ünitenin örüntüsü için ilişkili olan hatayı belirlemek için hesaplanan aktivasyonunu (y_k), hedef değeriyle (t_k) karşılaştırır. Bu hataya dayanarak, δ_k ($k = 1, \dots, m$) hesaplanır. δ_k çıkış ünitesi, Y_k 'daki hatayı önceki katmandaki tüm ünitelere (Y_k ile bağlantılı tüm gizli üniteler) geri dağıtmak için kullanır. Benzer şekilde faktör δ_j ($j = 1, \dots, p$) her gizli ünite Z_j için hesaplanır. Hatayı giriş katmanına geri yaymak gerekli değildir ancak δ_j , gizli katman ile gizli katman arasındaki ağırlıkları güncellemek için kullanılır.

Tüm δ faktörler belirlendikten sonra, tüm katmanlar için olan ağırlıklar aynı anda güncellenir. Ağırlık w_{jk} (gizli ünite Z_j 'den çıkış ünitesi Y_k 'ya) için güncelleme, faktör δ_k ve gizli ünite Z_j 'nin çıktı ünitesi z_j aktivasyonuna dayandırılır. Ağırlık v_{jk} (giriş ünitesi X_i 'den gizli ünite Z_j 'ye) için güncelleme, faktör δ_j ve x_i giriş ünitesinin aktivasyonuna dayandırılır (L. Fausett, 1994).

GYA için bir aktivasyon fonksiyonu, bir takım önemli özelliklere sahip olmalıdır: Sürekli, türevlenebilir ve tekdüze bir şekilde azalmayan bir fonksiyon olmalıdır. Ayrıca hesaplama verimi için, türevinin kolay hesaplanabilir olması arzu edilen bir durumdur. Kullanım için en uygun aktivasyon fonksiyonlarından biri Eş. (3.5) ile gösterilen ve (0,1) aralığına sahip olan ikili sigmoid fonksiyondur.

$$f(x) = \frac{1}{1 + \exp(-x)} \quad (3.5)$$

$$f'(x) = f(x)[1 - f(x)] \quad (3.6)$$

3.4 Eğitim Algoritması

Algoritma aşağıdaki sırayı takip eder (L. Fausett, 1994).

Adım 0 : Ağırlıkları başlat. (Küçük rastgele değerler ata).

Adım 1 : Durdurma işlemi hatalı olduğunda 2-9 adımlarını yap.

Adım 2 : Her bir eğitim çifti için, 3-8 adımlarını yap.

İleri besleme:

Adım 3 : Her giriş ünitesi ($X_i, i = 1, \dots, n$), x_i giriş sinyalini alır ve bu sinyali gizli katmandaki tüm ünitelere iletir.

Adım 4 : Her gizli ünite ($Z_j, j = 1, \dots, p$) ağırlıklı giriş sinyallerini toplar.

$$z_in_j = v_{oj} + \sum_{i=1}^n x_i v_{ij} \quad (3.7)$$

Çıkış sinyalini hesaplamak için kendi aktivasyon fonksiyonunu uygular,

$$z_j = f(z_in_j) \quad (3.8)$$

ve bu sinyali çıkış ünitelerine gönderir.

Adım 5 : Her çıkış ünitesi ($Y_k, k = 1, \dots, m$) kendi ağırlık giriş sinyallerini toplar,

$$y_in_k = w_{ok} + \sum_{j=1}^p z_j w_{jk} \quad (3.9)$$

ve kendi çıkış sinyallerini hesaplamak için kendi aktivasyon fonksiyonunu uygular,

$$y_k = f(y_in_k) \quad (3.10)$$

Hatanın geriyayılımı:

Adım 6 : Her çıkış ünitesi ($Y_k, k = 1, \dots, m$) , giriş eğitim örüntüsüne karşılık gelen bir hedef örüntü alır, hata bilgi terimini hesaplar.

$$\delta_k = (t_k - y_k)f'(y_{in_k}) \quad (3.11)$$

ağırlık düzeltme terimini hesaplar (Daha sonra w_{jk} güncellemek için kullanılan),

$$\Delta w_{jk} = \alpha \delta_k z_j \quad (3.12)$$

bias düzeltme terimini hesaplar (Daha sonra w_{ok} güncellemek için kullanılan)

$$\Delta w_{ok} = \alpha \delta_k \quad (3.13)$$

ve δ_k 'yı aşağı katmandaki ünitelere gönderir.

Adım 7 : Her gizli ünite ($Z_j, j = 1, \dots, p$) kendi delta girişlerini (yukarı katmandaki ünitelerden) toplar.

$$\delta_{in_j} = \sum_{k=1}^m \delta_k w_{jk} \quad (3.14)$$

kendi hata bilgi terimini hesaplayacak olan aktivasyon fonksiyonunun türevi ile çarpar,

$$\delta_j = \delta_{in_j} f'(z_{in_j}) \quad (3.15)$$

ağırlık düzeltme terimini (daha sonra v_{ij} 'yi güncellemek için kullanılan) hesaplar.

$$\Delta v_{ij} = \alpha \delta_j x_i \quad (3.16)$$

ve bias düzeltme terimini hesaplar (daha sonra v_{oj} 'yi güncellemek için kullanılan)

$$\Delta v_{oj} = \alpha \delta_j \quad (3.17)$$

Ağırlıkları ve Biasları güncelle:

Adım 8 : Her çıkış ünitesi ($Y_k, k = 1, \dots, m$) bias ve ağırlığını günceller ($j = 0, \dots, p$):

$$w_{jk}(\text{yeni}) = w_{jk}(\text{eski}) + \Delta w_{jk} \quad (3.18)$$

Her gizli ünite ($Z_j, j = 1, \dots, p$), bias ve ağırlığını günceller ($i = 0, \dots, n$):

$$v_{ij}(\text{yeni}) = v_{ij}(\text{eski}) + \Delta v_{ij} \quad (3.19)$$

Adım 9 : Durdurma durumunu test et.

3.5 Kesin Yerel Geri Yayılım (Strictly Local Backpropagation)

Geri yayılımın biyolojik olabilirlikten yoksun olduğundan dolayı sinirsel simulasyona uygun olmayışı eleştirilmektedir. Tartışmalardan biri geri yayılım algoritmasında proses elemanların arasında bilgi paylaşımı yapması gerektiği üzerineydi. Bu bölümde anlatılacak geri yayılımın modifiye edilmiş versiyonu bu açığı kapatmak için geliştirilmiştir (D. Fausett, 1990).

Geri yayılımın bu versiyonunda ağın, kortikal üniteler, sinaptik üniteler ve talamik üniteler olmak üzere 3 ünite türünü kapsadığı görülür. Her tür kendisine gelen kesin yerel bilgiyle kesin hesaplamalar yapar.

Bir kortikal ünite kendi girdilerini toplar ve sonuçlanan değeri yukarısındaki bir sonraki üniteye sinyal olarak iletir. Tam aksine, girdi kortikal üniteleri sadece bir girdi sinyali alır, bu yüzden toplamaya gerek yoktur. Gizli kortikal üniteleri kendi girdi sinyallerini toplar ve sonuçlanan sinyali, kendisiyle bağlantılı olan her sinaptik üniteye iletir. Çıktı kortikal üniteleri ayrıca kendi girdi sinyallerini toplar ancak her çıktı kortikal ünitesi, yukarısındaki sadece bir sinaptik üniteyle bağlantılıdır.

Sinaptik ünitelerin fonksiyonu tek bir girdi sinyali alır, o girdiye bir aktivasyon fonksiyonu uygular, sonucu bir ağırlıkla çarpar ve sonucu üstteki tek bir üniteye gönderir. Girdi sinaptik üniteler (girdi kortikal üniteler ve gizli kortikal üniteler arasındaki) kendi aktivasyon fonksiyonları olarak özdeşlik fonksiyonu kullanırlar. Her çıktı sinaptik ünitesi için olan ağırlık 1' dir. Her biri sinyalini bir talamik üniteye gönderir. Talamik ünitenin amacı, hesaplanan çıktı ile hedef değerini karşılaştırmaktır. Eğer ki eşleşmezler ise talamik ünite, altındaki çıktı sinaptik ünitesine bir hata sinyali gönderir. Bu, bilginin üniteler arasında paylaşılmış olması gereken geleneksel geri yayılım algoritmasının geri yayılım sürecinde olur.

$$\Delta w_{jk} = \alpha \delta_k z_j \quad (3.20)$$

ve

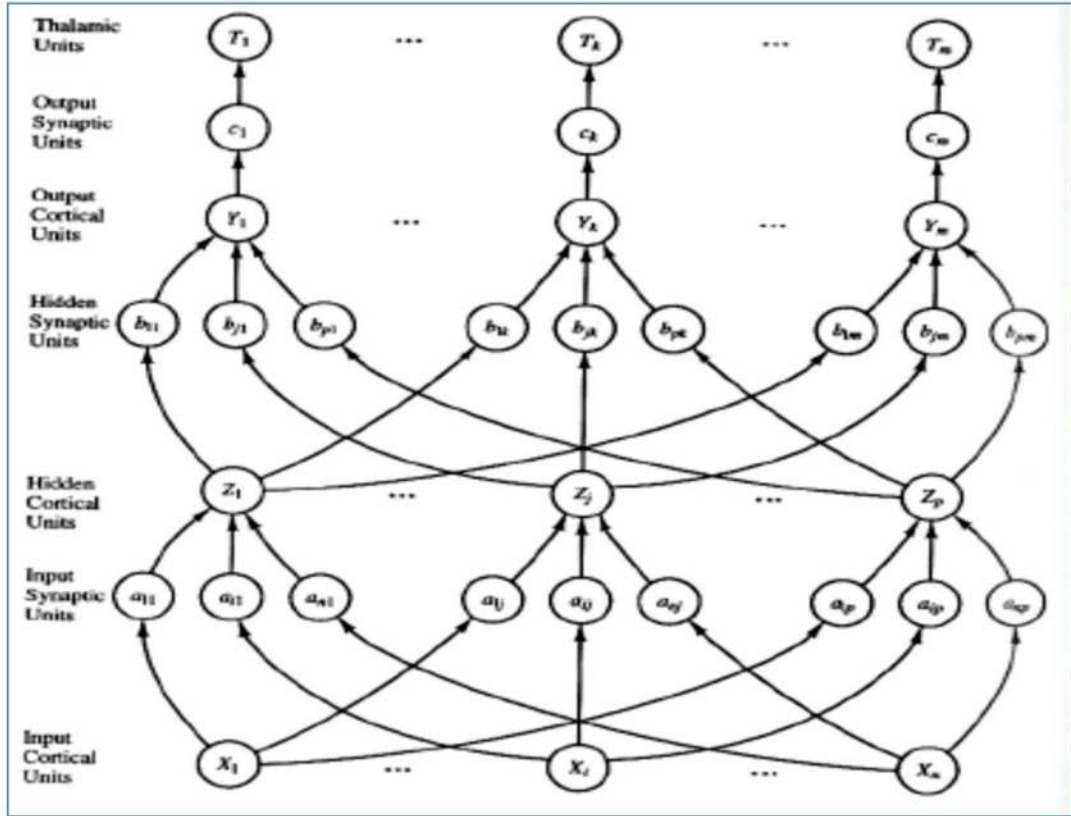
$$\Delta v_{ij} = \alpha \delta_j x_i \quad (3.21)$$

Her çıkış sinaptik ünitesi üstündeki talamik üniteden hata sinyali alır, bu sinyali 1 olan ağırlığıyla ve aktivasyon fonksiyonunun türevi ile çarpar. Sonuçlanan değeri altındaki gizli kortikal üniteye gönderir.

Her çıkış kortikal ünitesi δ_k olarak tanımlanan kendi tek girdi sinyalini alttaki gizli sinaptik ünitelere gönderir. Her gizli sinaptik ünite ağırlık güncelleme terimini daha sonra kullanmak üzere hesaplar (kendi giriş sinyali δ_k ürünü, aktivasyonunu ve bir öğrenme oranını). Daha sonra kendi girdi sinyalini ağırlığıyla ve aktivasyon fonksiyonunun türevi ile çarpar. Sonuçlanan değeri alttaki gizli

kortikal üniteye gönderir. Daha sonra gizli kortikal ünitesi kendi giriş sinyalini toplar ve sonuçlanan değeri üstteki girdi sinaptik üniteye gönderir. Girdi sinaptik ünite daha sonra kullanılmak üzere ağırlık güncelleme terimini hesaplar (kendi giriş sinyali δ_k ürünü, aktivasyonunu ve bir öğrenme oranını).

Kesin yerel geri yayılım algoritması, standart geri yayılım algoritmasına karşı yapılan eleştirilere cevap vermesinin yanısıra ağırlık güncelleme oranını, kullanılan aktivasyon fonksiyonlarının daha fazla çeşidine izin vererek artırır.



Şekil 3.2. Kesin Yerel Geri yayılım Ağı

Kaynak: Fausett, L. (1994). *Fundamentals of Neural Networks: Architecture, Algorithms and Applications*, New Jersey: Prentice Hall.

Bir ileri beslemeli geri yayılım ağında, standart ve kesin yerel geri yayılım algoritması karşılaştırılırsa, hesaplamaların aynı olduğu sadece farklı şekillerde düzenlendiği görülür.

Tablo 3.1. Standart Geri Yayılım ve Kesin Yerel Geri Yayılımın Karşılaştırılması

STANDART GERİ YAYILIM	KESİN YEREL GERİ YAYILIM
İleri Besleme <i>Girdi ünitesi:</i> Girdi sinyalini alır Girdi ünitesinden gizli üniteye yol: Ağırlıkları çarpar <i>Gizli ünite:</i> Girdi sinyallerini toplar Aktivasyon fonksiyonunu uygular Gizli üniteden çıktı ünitesine yol: Ağırlıkları çarpar <i>Çıktı ünitesi:</i> Girdi sinyallerini toplar Aktivasyon fonksiyonunu uygular	İleri Besleme <i>Girdi kortikal ünitesi:</i> Girdi sinyalini alır <i>Girdi sinaptik ünitesi:</i> Ağırlığı çarpar <i>Gizli kortikal ünitesi:</i> Girdi sinyallerini toplar <i>Gizli sinaptik ünitesi:</i> Aktivasyon fonksiyonunu uygular Ağırlığı çarpar <i>Çıktı kortikal ünitesi:</i> Girdi sinyallerini toplar <i>Çıktı sinaptik ünitesi:</i> Aktivasyon fonksiyonunu uygular
Hatanın geri yayılımı <i>Çıktı ünitesi:</i> Hatayı hesaplar f' ile çarpar Çıktı ünitesinden gizli üniteye yol: Ağırlıklarla çarpar Ağırlık düzeltmeyi hesaplar <i>Gizli ünite:</i> Girdiyi üst ünitelerden toplar f' ile çarpar Gizli üniteden girdi ünitesine yol: Ağırlık düzeltmeyi hesaplar	Hatanın geri yayılımı <i>Talamik ünitesi:</i> Hatayı hesaplar <i>Çıktı sinaptik ünitesi:</i> f' ile çarpar <i>Çıktı kortikal ünitesi:</i> Girdi δ_k 'yı alt ünitelere gönderir <i>Gizli sinaptik ünitesi:</i> Ağırlık düzeltmeyi hesaplar δ_k 'yı ağırlıkla çarpar f' ile çarpar <i>Gizli kortikal ünitesi:</i> Girdiyi üst ünitelerden toplar <i>Girdi sinaptik ünitesi:</i> Ağırlık düzeltmeyi hesaplar
Ağırlık güncelleme	Ağırlık güncelleme

3.6 Levenberg-Marquardt ve Bayesian Regülasyon (Regulation)

Metodu

Levenberg-Marquardt metodu, Gauss-Newton ve gradyan iniş algoritmalarının en iyi özelliklerinden oluşur ve bu iki metodun kısıtlamalarını kaldırır. Bu algoritma maksimum komşuluk üzerine kurulmuş en küçük kareler hesaplama metodudur. Genel olarak bu metot yavaş yakınsama probleminden etkilenmez.

Gauss-Newton algoritmasının kombinasyonu olan Levenberg-Marquart algoritması optimizasyon problemlerinde gradyan iniş algoritmasına göre daha etkindir ve daha hızlıdır (Foo, Ghosh, 2002).

k. iterasyondaki ağ ağırlık vektörü Eş. (3.22) ile gösterilirse:

$$w^{(k+1)} = w^{(k)} + \Delta w^{(k)} \quad (3.22)$$

Ortalama karesel hata fonksiyonu (mean square error):

$$E(w) = \frac{1}{2} \sum_{i=1}^n (e_i)^2 = E(w) = \frac{1}{2} \sum_{i=1}^n (d_i - y_i)^2 \quad (3.23)$$

$$\nabla E(w) = J^T(w) e(w) \quad (3.24)$$

$$\nabla^2 E(w) = J^T(w) J(w) \quad (3.25)$$

$$e(w) = (e^1, e^2, e^3, \dots, e^m)^T \quad (3.26)$$

$J(w)$ jakobiyen matrisi:

$$J = \begin{pmatrix} \frac{\partial e^1}{\partial w_1} & \dots & \frac{\partial e^1}{\partial w_N} \\ \vdots & \ddots & \vdots \\ \frac{\partial e^m}{\partial w_1} & \dots & \frac{\partial e^m}{\partial w_N} \end{pmatrix} \quad ((3.27)$$

$$\Delta w = -[\nabla^2 E(w)]^{-1} \nabla E(w) \quad (3.28)$$

Eş. 3.24 ve Eş. 3.25, Eş. 3.28 'de yerine koyulursa

$$\Delta w = -[J^T(w) J(w)]^{-1} J^T(w) e(w) \quad (3.29)$$

μ pozitif bir sabittir ve I birim matristir. μ çok büyükse Levenberg Marquart

$$\Delta w = -[J^T(w) J(w) + \mu I]^{-1} J^T(w) e(w) \quad (3.30)$$

algoritması gradyan iniş (gradient descent) algoritmasına yaklaşıır. Eğer μ , 0 ise Levenberg-Marquart algoritması Gauss-Newton metodu olur (He, Si, Tylavsky, 2000).

Bayesian regülasyonu Levenberg Marquardt optimizasyonuna göre ağırlık ve bias değerlerini günceller. Karesel hata ve ağırlıkların kombinasyonunu minimize eder ve ağı üretmek için doğru kombinasyonu belirler. Daha iyi genelleştirme yeteneğine sahip bir ağı oluşturmak için, Mackay (1992) regülasyon ile ağı parametrelerinin boyutunu sınırlayan method önermiştir. Regülasyon teknikleri ağırlık ve bias değerlerinin daha küçük değerlerde kalması için ağı zorlar. Bu ağıın cevabının daha yumuşak olmasına, ağıın ezberleme (overfitting) olasılığının azalmasına ve gürültüyü yakalamasına neden olur (Mackay, 1992)

BÖLÜM 4

4. UYGULAMA

4.1 Uygulamada Kullanılacak Değişkenler Hakkında Genel Bilgiler

Kolesterol, hayvanların vücut dokularındaki hücre zarlarında bulunan ve kan plazmasında taşınan bir sterol, yani bir steroid ve alkol birleşimidir. Daha düşük miktarlarda bitkilerde de bulunur. Kolesterol, özellikle hayvansal gıdalarda bulunur ve vücuttaki kolesterolün ancak bir kısmı gıda kaynaklıdır; çoğu vücut tarafından sentezlenir. Vücudun her hücresinde bulunmakla beraber, onun sentezlendiği veya hücre zarlarının daha çok olduğu organ ve dokularda, örneğin karaciğer, omurilik ve beyinde, ayrıca ateromlarda, kolesterolün yoğunluğu daha yüksektir. Kolesterol kanda normalden fazla bulunması halinde damarlarda birikerek damar sertleşmesine yol açar. Bazen de safra pigmentleri ile birleşerek safra taşlarının oluşumunda rol oynar.

Kolesterol pek çok biyokimyasal reaksiyonda yer almasına rağmen özellikle lipoproteinlerin kolesterolü taşıma biçimleri ve kandaki kolesterol düzeyleriyle kalp hastalıkları arasındaki bağlantıdan dolayı bilinir. Vücut, kolesterolü kullanarak hormonlar (kortizol, üreme hormonları), D vitamini ve yağları sindiren safra asitlerini üretir. Bu işlemler için kanda çok az miktarda kolesterol bulunması yeterlidir. Eğer kanda fazla miktarda kolesterol varsa kan damarlarında birikir, sertleşmeye ve daralmaya yol açar. Aterosklerozda damar duvarında biriken tek madde kolesterol değildir; akyuvarlar, kan pıhtısı, kalsiyum gibi maddeler de birikir. Ateroskleroza halk arasında damar sertliği, damar kireçlenmesi de denir. Yüksek kan kolesterolünün zararlarından bahsedilirken söz konusu olan "kötü kolesterol", yani düşük yoğunluklu lipoprotein (LDL) tarafından taşınan kolesterol düzeyidir. Yüksek yoğunluklu lipoprotein (HDL) tarafından taşınan kolesterola "iyi kolesterol" denir.

Kolesterol hücre zarlarının (membranlarının) inşası ve bakımı için gereklidir. Kolesterol içeren membranlar daha geniş sıcaklık aralığında

akışkanlıklarını korurlar. Kolesterol, yağların sindirimine yarayan safranin sentezlenmesinde kullanılır. Ayrıca yağda çözünen vitaminlerin metabolizmasında rolü önemlidir. Aldosteron, testosteron, östrojen ve projesteron gibi steroid hormonlarının ve kortizolun sentezlerinde yer alır. Başka araştırmalar kolesterolün sinir hücreleri arasındaki sinapslarda ve bağışıklık sistemi hücrelerinin işlevlerinde rol oynadığını gösterir. Hücre membranının yapısına etkisi sonucunda hücre sinyal iletimine ve membranlardaki iyon ve proton geçirgenliğine de etki eder.

LDL (Düşük yoğunluklu lipoprotein), kanda kolesterol taşıyan ve yoğunluğu 1,019-1,063 g/mL arasında olan lipoprotein sınıfına karşılık gelir. Karaciğerde üretilen çok düşük yoğunluklu lipoprotein (VLDL) metabolizması sonucu oluşur. LDL tanecikleri 18-25 nm çapındadır, taşıdığı lipidlerin yanı sıra apolipoprotein B-100 (apoB-100) ve apoE proteinlerini içerir. LDL seviyesi ile kalp hastalıkları arasındaki bağlantıdan dolayı sıkça "kötü" kolesterol olarak anılır. LDL'nin başlıca işlevi, kolesterol ve trigliserit üreten hücre ve dokulardan bu molekülleri alıp bunlara gereksinimi olan hücre ve dokulara taşımaktır. Yapısında %21 protein, %11 trigliserit, %22 fosfolipid, %37 kolesterol ester, %8 serbest kolesterol ve %1 serbest yağ asitleri bulunur. Vücuttaki toplam kolesterolün %70'i LDL'de bulunmaktadır.

HDL (Yüksek yoğunluklu lipoproteinler), vücuttaki dokulardan karaciğere kolesterol taşıyan bir lipoprotein sınıfıdır. Yapısında %50 protein, %24 fosfolipid, %2 kolesterol, %4 yağ ve %20 kolesterol ester bulunur. HDL, karaciğerde üretilir. HDL arterlerde oluşan ateromlardaki kolesterolü alıp vücuttan atılmak üzere karaciğere taşıdığı için bu lipoproteinde bulunan kolesterol 'iyi kolesterol' olarak bilinir.

HDL lipoproteinlerin en küçükleridir. Yüksek oranda protein içermelerinden dolayı yoğundurlar. Başlıca apolipoprotein A-I (apoA-I) ve apoA-II proteinlerini içerirler. Bu lipoproteinler karaciğerde fosfolipidler eşliğinde bileşikler olarak sentezlendiğinde madenî para gibi yassı bir görünümüne olur. Bu yeni oluşmuş tanecikler yakınından geçtikleri hücrelerin membranlarından kolesterol molekülleri emebilirler. Plazmada bulunan Lesitin kolesterol asiltransferaz adlı

enzim bu kolesterolu kolesteril estere dönüştürür. Kolesteril esterler, kolesterolden daha hidrofobik lipitler olduğundan dolayı HDL'in ortasında birikirler ve bu birikmenin sonunda HDL küresel bir biçim alır. HDL dolaşım sırasında hücrelerde kolesterol emmeye devam eder ve büyür. Bu yüzden HDL'nin koruyucu özelliği taşıdığı kolesterol miktarı ile değil, büyük HDL taneciklerinin sayısı ile ilişkilidir.

Trigliserit, gliserol (gliserin) ve üç yağ asidinden oluşan bir esterdir. Bitkisel ve hayvansal yağların ana bileşenidir. Trigliceritler enerji kaynağı olarak metabolizmada önemli rol oynarlar. Karbohidratlar ve proteinlerin iki katı enerji taşırlar (9 kalori/g). İnce barsakta trigliseritler, lipaz enzimleri ve safranin etkisiyle gliserol ve yağ asitlerine ayrışır, bunlar da kana geçer. Kanda, gliserit ve yağ asitlerinin biraraya gelmesiyle trigliseritler yeni baştan oluşurlar ve lipoproteinlere katılırlar. Lipoproteinler, diğer işlevlerinin yanı sıra, yağ hücreleriyle diğer hücreler arasında (trigliserit moleküllerinin parçası olarak) yağ asitleri taşımayada yararlar. Vücuttaki çoğu hücre, gereksinimlerine bağlı olarak yağ asitlerini ya salgılar ya da içine alır. Yağ hücreleri ayrıca trigliseritleri sentezleyip depolama yeteneğine sahiptirler. Vücut, enerji kaynağı olarak yağ asitlerine ihtiyaç duyduğunda, glukagon hormonunun verdiği işaret üzerine, hormon duyarlıklılı lipaz enzimi trigliseritleri yağ asitlerine parçalar.

Kreatinin, böbrek fonksiyonlarının göstergelerinden biridir. Yüksekliği - Düşüklüğü kas metabolizmasının önemli bir son ürünüdür ve böbreklerden idrar ile tümüyle ve değişmeden atılır. Böbrek fonksiyonları bozulmuşsa bu atılım yavaşlar ve kanda kreatinin yükselir.

Glukoz (Açlık), kandaki şeker miktarının belirli bir açlık sonrası ölçülen seviyesidir. Normal kişiler aç kalınca insülin dışındaki hormonlar kan şekerini arttırırken insülin aşağıya çeker. Şeker hastalarında insülin hormonu iyi çalışmadığından kan şekeri yüksek çıkar. Açlık kan şekeri ölçümü şeker hastalığının teşhisinde kullanıldığı gibi, tedavinin etkili olup olmadığını anlamak içinde kullanılır. Açlık kan şekerinin normal değeri sağlıklı kişilerde 70-100 mg/dl arasındır.

4.2 Bulgular

Çalışmada 273 bireye ait LDL, HDL, Trigliserit, Total Kolesterol, Kreatinin, Glukoz (Açlık), Yaş ve Cinsiyet verileri, Özel Malatya Sevgi Hastanesi Dâhiliye Polikliniğinden sağlanmıştır. Kullanılacak olan veriler %80 eğitim, %20 test kümesi olmak üzere iki gruba ayrılmıştır. LDL değerlerinin tahmininde YSA modelleri kullanılmıştır.

Farklı iterasyon (epoch) sayılarında denemeler yapılarak eğitim gerçekleştirilmiş daha sonra test için ayrılan veriler ile kurulan tüm modeller test edilmiştir. Test işlemi sonucunda bulunan tahmin değerleri, gerçek değerlerle karşılaştırılmış ve hata kareleri ortalaması (MSE) dikkate alınarak değişik mimarilere sahip yapay sinir ağı modellerinin tahmin performansları karşılaştırılmıştır.

YSA modellerinde aktivasyon fonksiyonu olarak sigmoid aktivasyon fonksiyonu kullanılmıştır. Sigmoid aktivasyon fonksiyonu yalnızca [0.1] aralığında değerler üretebildiği için çıktılarında buna uygun olarak ölçeklendirilmesi gerekir. Aşırı küçük ve aşırı büyük değerlerin ağı yanlış yönlendirme olasılığını yok etmek için girdi setleri [0.1] aralığında normalleştirilmiştir.

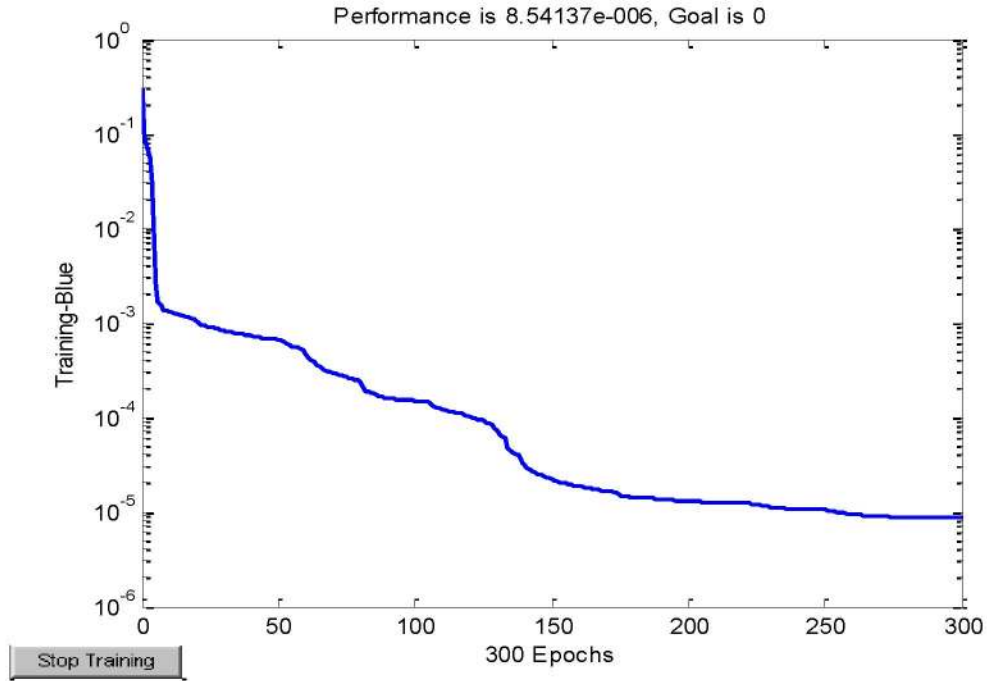
Modelin oluşturulmasında MATLAB programlama tabanı kullanılmıştır. YSA, 1 girdi katmanı, 1 gizli (ara) katman ve 1 çıktı katmanı olacak şekilde oluşturulmuş, öğrenme oranı ve momentumlu geri yayılım algoritması, Levenberg-Marquardt geriyayılım algoritması ve Bayesyen düzeltmeye dayalı geri yayılım algoritması ile eğitilmiştir.

Tek gizli katmanlı ağlarda gizli nöron sayısına ilişkin önerilerden n (Tang ve Fishwick, 1993), $2n + 1$ (Lippmann, 1987), $2n$ (Wong, 1991) olacak şekilde gizli nöron sayıları belirlenmiştir. Veri kümesinden örnekler rastgele seçilerek eğitim ve test yapılmış ve değişik YSA modelleri, öğrenme algoritmaları ve gizli nöron sayıları denenerek en iyi sonuç elde edilmeye çalışılmıştır. Yapılan çalışmalarda elde edilen minimum hata değerleri çizelgelerde sunulmuştur.

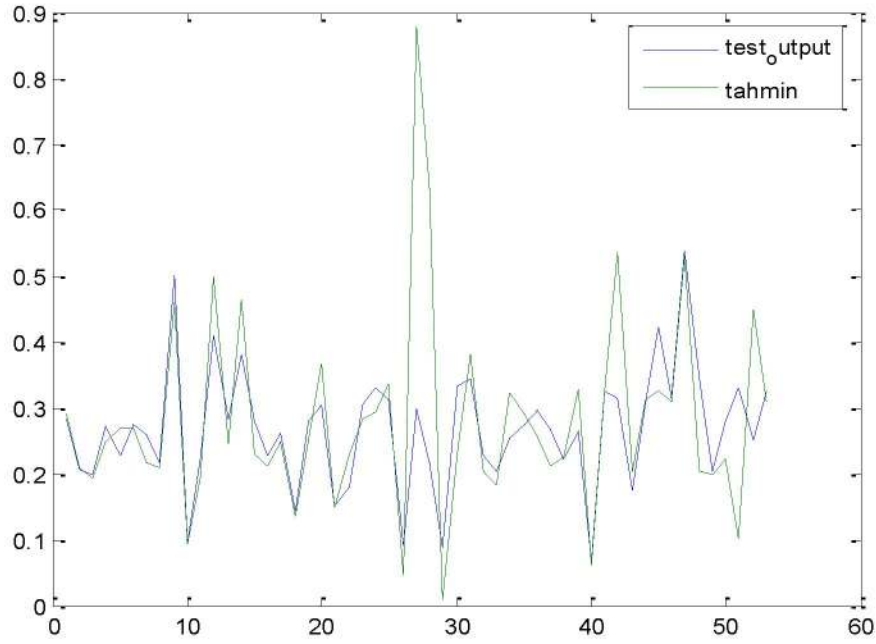
Tablo 4.1. Levenberg-Marquardt Öğrenme Algoritmasının Farklı Gizli Nöron Sayılarına Göre Performansı

Öğrenme Algoritması	Gizli Nöron Sayısı	MSE
Levenberg-Marquardt	$n = 7$	0,022319
Levenberg-Marquardt	$2n = 14$	0,014170
Levenberg-Marquardt	$2n+1 = 15$	0,051133

Tablo 4.1. incelendiğinde Levenberg-Marquardt öğrenme algoritmasında en iyi performansın 14 gizli nörona sahip ağ yapısında elde edildiği görülmektedir. Bu ağ yapısının eğitim performansı Şekil 4.1. de, test performansıda Şekil 4.2. de gösterilmiştir.



Şekil 4.4. Gizli Nöron Sayısı 14 Olan Levenberg-Marquardt Öğrenme Algoritmasının Eğitim Performansı

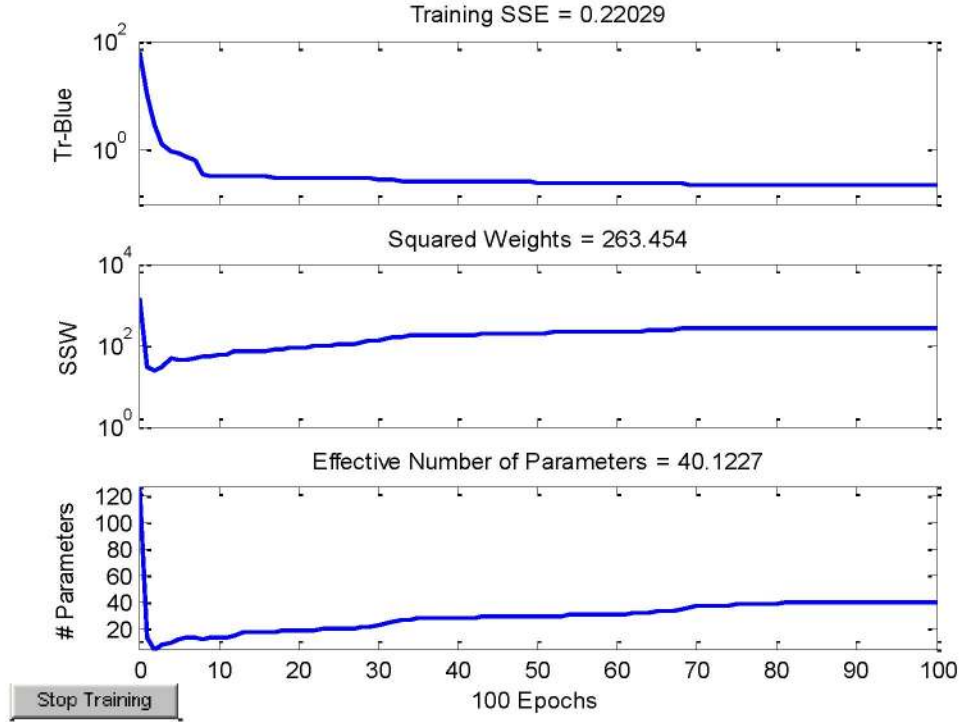


Şekil 4.2. Gizli Nöron Sayısı 14 Olan Levenberg-Marquardt Öğrenme Algoritmasının Test Performansı

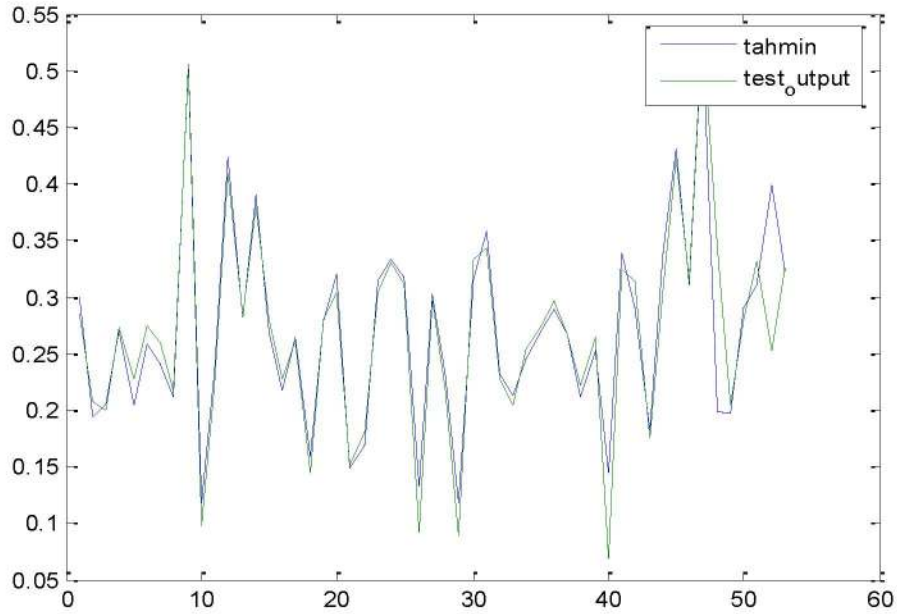
Tablo 4.2. Bayesian Regulasyon (Regulation) Öğrenme Algoritmasının Farklı Gizli Nöron Sayılarına Göre Performansı

Öğrenme Algoritması	Gizli Nöron Sayısı	MSE
Bayesian Regulasyon (Regulation)	$n = 7$	0,009830
Bayesian Regulasyon (Regulation)	$2n = 14$	0,001102
Bayesian Regulasyon (Regulation)	$2n+1 = 15$	0,001104

Tablo 4.2. incelendiğinde Bayesian Regulasyon (Regulation) öğrenme algoritmasında en iyi performansın 14 gizli nörona sahip ağ yapısında elde edildiği görülmektedir. Bu ağ yapısının eğitim performansı Şekil 4.3. de, test performansıda Şekil 4.4. de gösterilmiştir.



Şekil 4.3. Gizli Nöron Sayısı 14 Olan Bayesian Regülasyon (Regulation) Öğrenme Algoritmasının Eğitim Performansı



Şekil 4.4. Gizli Nöron Sayısı 14 Olan Bayesian Regülasyon (Regulation) Öğrenme Algoritmasının Test Performansı

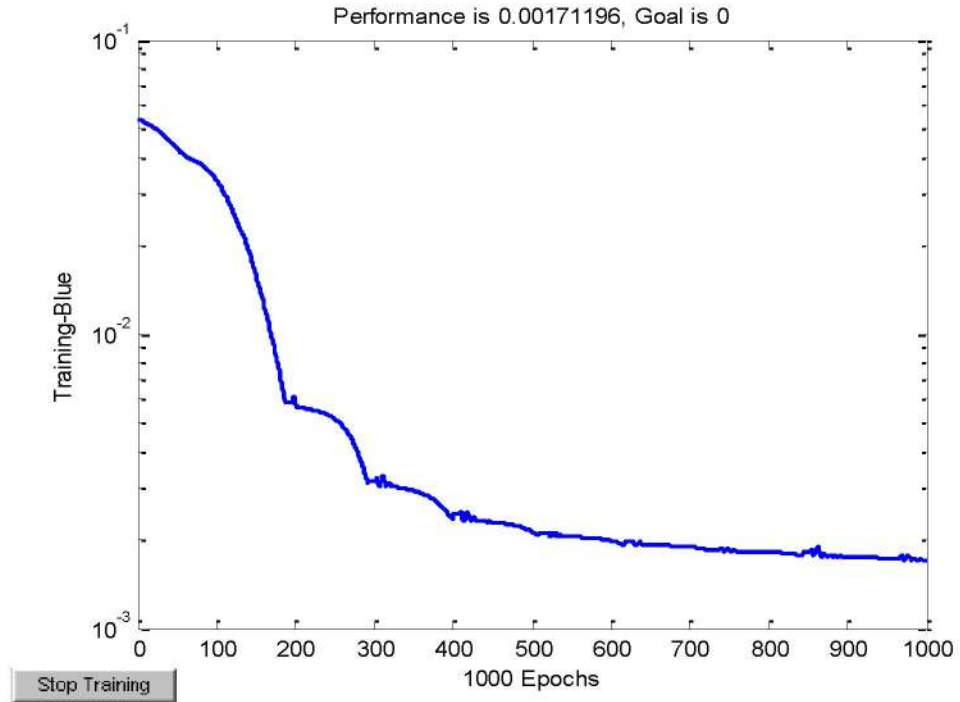
Öğrenme oranı ve momentumlu geri yayılım algoritması ile eğitilen yapay sinir ağları için en iyi performans veren ağ yapısına ulaşmak amacıyla çeşitli

momentum ve öğrenme katsayılarında denemeler yapılarak optimum ağ yapıları elde edilmeye çalışılmıştır. Yapılan denemeler sonucunda, öğrenme katsayısının 0.01, momentum katsayısının 0.9 olarak kullanılması ile öğrenme oranı ve momentumlu geri yayılım algoritması ile eğitilen en iyi yapay sinir ağı yapısı elde edilmeye çalışılmıştır.

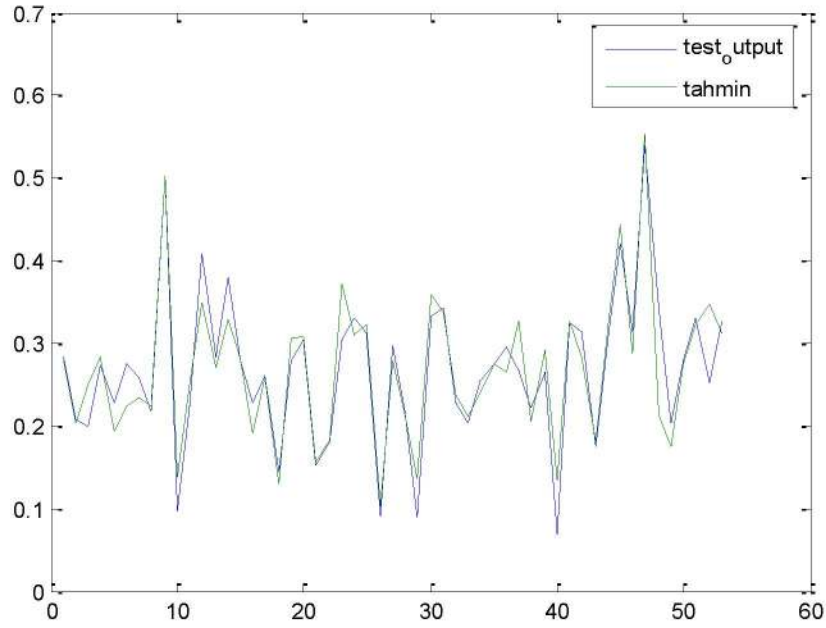
Tablo 4.3. Öğrenme Oranı ve Momentumlu Geri Yayılım Öğrenme Algoritmasının Farklı Gizli Nöron Sayılarına Göre Performansı

Öğrenme Algoritması	Gizli Nöron Sayısı	MSE
Öğrenme oranı ve momentumlu geri yayılım	$n = 7$	0,002141
Öğrenme oranı ve momentumlu geri yayılım	$2n = 14$	0,001249
Öğrenme oranı ve momentumlu geri yayılım	$2n+1 = 15$	0,001524

Tablo 4.3. incelendiğinde öğrenme oranı ve momentumlu geri yayılım öğrenme algoritmasında en iyi performansın 14 gizli nörona sahip ağ yapısında elde edildiği görülmektedir. Bu ağ yapısının eğitim performansı Şekil 4.5. de, test performansıda Şekil 4.6. da gösterilmiştir.



Şekil 4.5. Gizli Nöron Sayısı 14 Olan Öğrenme Oranı ve Momentumlu Geri Yayılım Öğrenme Algoritmasının Eğitim Performansı



Şekil 4.6. Gizli Nöron Sayısı 14 Olan Öğrenme Oranı ve Momentumlu Geri Yayılım Öğrenme Algoritmasının Test Performansı

4.3 Sonuç ve Tartışma

YSA yöntemi tıp alanında ve başka birçok alanda tahmin modellemesi için geliştirilen yöntemlerden biridir. Bu çalışmada tahmin modellemesi için YSA modelleri araştırılmıştır. Bu amaç doğrultusunda YSA teorisi detaylı bir şekilde incelenmiş, uygulama alanları ve sık kullanılan bazı YSA mimarileri tanıtılmıştır. Uygulama kısmında ise farklı YSA modelleri kullanılarak LDL Kolesterol değerleri tahminine yönelik tahmin modelleri geliştirilmiştir.

Yaptığımız çalışmanın uygulama kısmında kolesterol tanısında kullanılan değişkenler incelenmiştir. LDL Kolesterol değerini tahmin etmek için kullanılan YSA modeli ve öğrenme algoritmasının hangisinin daha iyi sonuç verdiği araştırılmıştır. Kullanılan YSA modelinin tahmin performansı hata kareleri ortalaması (MSE) değerlerine göre değerlendirilmiştir. Çalışmadaki YSA modellerinden Levenberg- Marquardt, Bayesian Regulation ve Öğrenme oranı ve momentumlu geri yayılım öğrenme algoritmalarının karşılaştırılması sonucunda Bayesian Regulation öğrenme algoritmasının daha iyi sonuç verdiği görülmüştür.

Ayrıca oluşturulan YSA modelleri incelendiğinde, gizli katmandaki nöron sayılarının sonucu değiştirdiği ve karşılaştırılan öğrenme algoritmalarının hepsinde gizli nöron sayısı $2n$ (Wong, 1991) olan ağ yapılarının daha iyi sonuçlar verdiği görülmüştür.

Sonuç olarak YSA, istatistikî yöntemlerden farklı olarak, verinin özellikleri ile ilgili sağlanması gereken istatistikî varsayımlarda bulunmadığı gibi algoritma veya matematiksel model geliştirmeye de gerek duymaz. Veri içindeki ilişki kalıplarını ve desenleri tanıyarak öğrenir. Bu nedenle modelin kurulumu ve kullanımı açısından basit, açık ve esnektir. Günümüzde, bu özellikleri ile yapay sinir ağları, tahmin yöntemi olarak pek çok alanda kullanılmakta ve diğer yöntemlere tercih edilmektedir. Problemin yapısına uygun olarak kurulmuş ve eğitilmiş bir yapay sinir ağı, günümüzde kullanılan birçok tahmin metodundan çok daha iyi sonuçlar verecektir.

KAYNAKÇA

- Akgül, I. (2003), *Zaman Serilerinin Analizi ve ARIMA Modelleri*, İstanbul: Der Yayınları.
- Aktaş, M., Okumuş, H. İ. (2003), *Doğrudan Moment Kontrollü Asenkron Motorun Stator Direncinin Yapay Sinir Ağı ile Kestirimi*, International XII. Turkish Symposium on Artificial Intelligence and Neural Networks.
- Alavala, C.R. (2003), *Fuzzy logic and neural Networks: Basic concept & appliations*, New Age International Publisher.
- Archer, B. (1980) *Forecasting Demand: Quantitative and Intuitive Techniques*, International Journal of Tourism Management, Vol: 5, March 1980, s. 177.
- Arslan, A., İnce, R.(1995), *Geriye Yayılma Sinir Ağı Kullanılarak Betonarme Kolonların Tasarımı*, Tr. Journal of Engineering and Environmental Sciences, Vol 19, 127-135.
- Auger, M.(2001), *Detection of Laser-Welding Defects Using Neural Networks*, A Thesis for The Degree Of Master of Science (Engineering), Queen's University.
- Baily D., Thomson, D.M. (1990), *Devoloping Neural Network Applications*, AI Expert, September, 33-41.
- Bayır, F. (2006), *Yapay Sinir Ağları ve Tahmin Modellemesi Üzerine Bir Uygulama*, Yüksek Lisans Tezi, İstanbul Üniversitesi Sosyal Bilimler Enstitüsü İşletme Anabilim Dalı Sayısal Yöntemler Bilim Dalı.
- Begg, R., Kamruzzaman, J. and Sarker, R. (2006), *Neural networks in healthcare : potential and challenges*, Hershey: Idea Group Inc.
- Bülbül, S.(1994), *Zaman Serilerinde Üstel Düzeltme Modelleri ve Bir Uygulama*, Marmara Üniversitesi Sosyal Bilimler Enstitüsü Dergisi, Cilt: 1, Sayı: 1, s. 44.
- Caudill, Maureen (1987), *Neural Network Primar Part 1*, AI Expert.
- Çelik, B. (2008), *Yapay Sinir Ağları Metodolojisi İle Zaman Serisi Analizi: Teori ve Uygulama*, Yüksek Lisans Tezi, Marmara Üniversitesi, Sosyal Bilimler Enstitüsü Ekonometri Anabilim Dalı.
- De Lurgio, S. A.(1998), *Forecasting Principles and Applications*, Irwin / Mc Graw - Hill, Boston, s. 21.

- Demirhan, A., Kılıç, Y.A., Güler, İ. (2010), *Tıpta Yapay Zekâ Uygulamaları*, Yoğun Bakım Dergisi 2010; 9(1):31-41.
- Elmas Ç., (2003), *Yapay Sinir Ağları (Kuram, Mimari, Eğitim, Uygulama)*, Ankara: Seçkin Yayıncılık.
- Fausett, D.W. (1990), *Strictly Local Backpropagation*, San Diego: International Joint Conference on Neural Networks.
- Fausett, L. (1994), *Fundamentals of Neural Networks: Architecture, Algoritma and Applications*, New Jersey: Printice Hall.
- Foo, J.S., Ghosh, P.S. (2002), *Artificial Neural Network Modelling of Partial Discharge Parameters For Transformer Oil Diagnosis*, Malaysia: Annual Report Conference on Electrical Insulation and Dielectric Phenomena.
- Hamzaçebi C. (2011), *Yapay Sinir Ağları: Tahmin Amaçlı Kullanımı Matlab ve Neurosolutions Uygulamalı*, Bursa: Ekin yayıncılık.
- Haykin, S.(1994), *Neural networks: A comprehensive foundation*. New York: Macmillan College Publishing Company Inc.
- He, Q., Si, J., Tylavsky, D. (2000), *Prediction of top-oil temperature for transformers using neural network*, IEEE Transactions on Power Delivery, 1205-1211.
- Hebb, D. (1949), *Organisation of Behaviour*, New York: John Wiley and Sons.
- Kaastra, I., Boyd M. (1996), *Designing A Neural Network For Forecasting Financial And Econometric Time Series*, Neurocomputing, 10:215-236.
- Karlık, B. (1994), *Çok Fonksiyonlu Protezler İçin Yapay Sinir Ağları Kullanılarak Miyoelektrik Kontrol*, Yıldız Teknik Üniversitesi, Fen Bil. Enstitüsü Doktora Tezi.
- Kohonen, T. (1984), *State Of The Art In Neural Computing*, IEEE First International Conference on Neural Networks.
- Lippmann, R.P.(1987), *An Introduction to Computing With Neural Nets*, IEEE ASSP Magazine, April, 4-22.
- Mackay, D. J. C. (1992), *Bayesian interpolation*, Neural Computation, 4: 415-447.
- Masters, T. (1993), *Practical Neural Network Recipes in C++*, New York: Academic Press, 25-50.

- Nabiyev, V. V., (2005), *Yapay Zeka*, 1. Bölüm Doğal ve Yapay Zeka, Seçkin Yayıncılık: Ankara.
- Orhunbilge, N. (2002), *Uygulamalı Regresyon ve Korelasyon Analizi*, İstanbul Üniversitesi İşletme Fakültesi Yayını, No: 281, İstanbul, s. 5 ve 12.
- Öztemel, E. (2003), *Yapay Sinir Ağları*, İstanbul: Papatya Yayıncılık.
- Papik, K., Molnar, B., Schaefer, R., Dombovari, Z., Tulassay, Z., Feher, J.(1998) *Application of Neural Networks in medicine-a review*. Med Sci Mon 1998;4:538-46.
- Saraç, Tuğba (2004), *Yapay Sinir Ağları*, Seminer Projesi, Gazi Üniversitesi Endüstri Mühendisliği Anabilim Dalı.
- Schalkoff, R.J. (1992), *Pattern Recognition: Statistical, Structural, Structural and Neural Approaches*, USA: R. R. Donnelley and Sons, Inc.
- SinanCanan.Net, <http://www.sinancanan.net.tr/2011/03/sinirbiliminde-yeni-arayislar-1.html>
- Smith, A. E. (1994), *X-Bar and R Control Chart Interpretation Using Neural Computing*, International Journal of Production Research, 32 (2), 309-320.
- Şen, Z.(2004), *Yapay Sinir Ağları İlkeleri*, İstanbul: Su Vakfı.
- Tang, Z., Fishwick, P.A. (1993), *Feedforward Neural Nets as Models for Time Series Forecasting*, ORSA Journal on Computing, 5(4):374-385.
- Uğur, A., Kınacı, A.C., “*Yapay Zeka Teknikleri ve Yapay Sinir Ağları Kullanılarak Web Sayfalarının Sınıflandırılması*”, Inet-tr 2006, XI. Türkiye’de İnternet Konferansı, TOBB Ekonomi ve Teknoloji Üniversitesi, Ankara.
- Wong, F.S.(1991), *Time Series Forecasting Using Backpropagation Neural Networks*, Neurocomputing, 2:147-159.
- Yıldız B. (2009), *Finansal Analizde Yapay Zekâ*, İstanbul: Beta Basım.
- Zhang, G., Patuwo, B. E., Hu, M. Y. (1998) *Forecasting with artificial neural networks: the state of the art*, International Journal of Forecasting, 14, ss.35-62.
- Zurada, J. M. (1995), *Introduction to Artificial Neural Systems*, Boston: PWS Publishing Company.