

**İSTANBUL ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
İŞLETME ANABİLİM DALI
SAYISAL YÖNTEMLER BİLİM DALI**

YÜKSEK LİSANS TEZİ

**Yapay Sinir Ağları
ve
Kredi Taleplerinin Değerlendirilmesi Üzerine Bir
Uygulama**

**Pakize YİĞİT
2501080232**

**Tez Danışmanı:
Prof. Dr. M. Erdal BALABAN**

İstanbul, 2011



T.C.
İSTANBUL ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
MÜDÜRLÜĞÜ



TEZ ONAYI

Enstitümüz **SAYISAL YÖNTEMLER** Bilim Dalında ders dönemindeki Eğitim - Öğretim Programını başarı ile tamamlayan **2501080232** numaralı **PAKİZE YİĞİT'İN** hazırladığı "**Yapay Sinir Ağları Ve Kredi Taleplerinin Değerlendirilmesi Üzerine Bir Uygulama**" konulu **YÜKSEK LİSANS/ DOKTORA TEZİ** ile ilgili **TEZ SAVUNMA SINAVI**, Lisansüstü Öğretim Yönetmeliği'nin 15.Maddesi uyarınca **29.062011 ÇARŞAMBA** günü saat **14:00'da** yapılmış, sorulan sorulara alınan cevaplar sonunda adayın tezinin **Kabul**üne *** OYBİRLİĞİ /OYÇOKLUĞUYLA** karar verilmiştir.

JÜRİ ÜYESİ	KANAATI(*)	İMZA
PROF.DR.MEHMET ERDAL BALABAN	Kabul	
DOÇ.DR.ÇİĞDEM ARICIGİL ÇILAN	Kabul	
DOÇ.DR. SEVİNC GÜLSEÇEN	Kabul	
DOÇ.DR. EYÜP ÇETİN	Kabul	
YRD.DOÇ.DR.U.TUĞBA GÜR SOY	Kabul	

Adres: Besim Ömerpaşa Caddesi Kaptan-ı Derya Sokakı 34452 Beyant/İstanbul
Tel: 0212 440 00 00 / 14219-14220-14221-14222-14226-14227-14243
Fax: 0212 440 03 40 e-mail: sbe@istanbul.edu.tr

ÖZ

Bu çalışmanın amacı, günümüzün en iyi tahmin yöntemlerinden biri olan Yapay Sinir Ağlarının (YSA) tahmin gücünü kredi taleplerinin tahmininde gösterilmesidir. Uygulamanın YSA sonuçları, kredi taleplerinin değerlendirilmesinde en çok kullanılan istatistik yöntemlerinden biri olan Lojistik Regresyon (LR) sonuçları ile ROC Eğrileri kullanılarak karşılaştırılmış ve bu çalışma için en uygun tahmin yöntemi ve sonucu bulunmaya çalışılmıştır.

Anahtar Kelimeler: Yapay Zeka, Yapay Sinir Ağları, Lojistik Regresyon Analizi, ROC Analizi, Kredi Taleplerinin Değerlendirilmesi.

ABSTRACT

The purpose of this study is to perform an application of Artificial Neural Networks (ANN) which is one of the promising Artificial Intelligence techniques nowadays, as a forecasting technique and to compare it with statistical method; the logistic regression analysis. Performances of models are compared by using ROC curves.

Keywords: Artificial Intelligence, Artificial Neural Networks, Logistic Regression, ROC Curve, Evaluating Consumer Loans, ROC Curve.

ÖNSÖZ

İnsanların, kendileri gibi düşünen, konuşan, hareket eden varlıklar yapma fikri antik çağlardan beri vardır. İnsanların bu merakı ile ortaya çıkan Yapay Zeka, insan beyninin ve sinir sisteminin analiz edilip matematiksel modellenmesiyle gelişerek günümüzdeki teknolojisine erişmiştir. Yapay zeka teknolojilerinden biri olan YSA, günümüzün en çok kullanılan tahmin tekniklerinden birisidir.

YSA, doğrusal olmayan yapılarda da çok iyi tahmin yapabilmeleri, esneklikleri, sınıflandırma, kümeleme, optimizasyon problemlerinde çok iyi sonuçlar verebilmeleri gibi avantajları ile savunmadan sağlığa, finanstan endüstriye kadar pek çok alanda yaygın olarak kullanılmaktadır.

Bu çalışmada, YSA hakkında genel bilgi verilmiş ve YSA'nın tahmin gücünü göstermek üzere kredi talepleri üzerinde bir deneme yapılmıştır. Bilinen istatistik tekniklerle kıyaslayabilmek için LR analizi yapılmış, sonuçlar ROC eğrisinde karşılaştırılmıştır.

İlk bölümde, YSA'nın temeli olan yapay zeka, tarihsel gelişimi ve diğer teknolojileri açıklanmış, ikinci bölümde YSA'nın temel yapısı, çalışma şekli ve buna bağlı olarak parametreleri anlatılmıştır. Üçüncü bölümde, ağ yapıları incelenerek, sık kullanılan bazı ağ topolojileri hakkında detaylı bilgiler verilmiştir. Dördüncü bölümde, yapay sinir ağı modeli tasarımında izlenmesi gereken yol ve önemli bazı parametrelerden bahsedilmiştir. Beşinci bölümde kısaca LR hakkında bilgi verilmiştir. Altıncı bölümde ise, özel bir finans kuruluşundan alınan firmalara ait kredi başvuru bilgileri LR ve YSA ile analiz edilmiş; sonuçlar ROC eğrisinde karşılaştırılarak, en iyi tahmin yöntemi bulunmaya çalışılmıştır.

Bu çalışmam esnasında, bana her zaman destek olan ailem başta olmak üzere, beni her zaman cesaretlendiren ve destek olan Danışman Hocam Prof.Dr.M.Erdal

BALABAN'a, tezin yazım aşamasında destek aldığım hocalarıma ve eğitim hayatım boyunca ders aldığım bütün hocalarıma çok teşekkür ederim.

İÇİNDEKİLER

ÖZ	iii
ABSTRACT	iv
ÖNSÖZ	v
İÇİNDEKİLER	vii
TABLO LİSTESİ	x
ŞEKİL LİSTESİ	xi
KISALTMA LİSTESİ	xii
EKLER	xiii
GİRİŞ	1
1. YAPAY ZEKA	3
1.1 . Yapay Zekanın Tarihi	7
1.1.1 Turing Testi	10
1.1.2 Çin Odası Deneyi	11
1.2. Yapay Zeka Teknolojileri	12
1.2.1. Uzman Sistemler	12
1.2.1.1. Bilgi Tabanı	13
1.2.1.2. Çıkarım Mekanizması	13
1.2.1.3. Kullanıcı Arabirimi	13
1.2.1.4. Bilgiyi Alma Ünitesi	13
1.2.1.5. Açıklama Ünitesi	13
1.2.2. Makine Öğrenmesi	14
1.2.3. Genetik Algoritmalar	15
1.2.3.1. Kromozom ve Gen	15
1.2.3.2. Çaprazlama	16
1.2.3.3. Mutasyon	16
1.2.3.4. Uygunluk Fonksiyonu	16
1.2.3.5. Yeniden üretim	16
1.2.4. Bulanık Mantık	16
2. YAPAY SINIR AĞLARI	19
2.1. Yapay Sinir Ağlarının Tarihsel Gelişimi	20
2.2. Yapay Sinir Ağlarının Yapısı ve Temel Elemanları:	22
2.2.1. Biyolojik Sinir Ağları	22
2.2.2. Yapay Sinir Hücresi (Proses Elemanı)	23
2.2.2.1. Girdiler	24
2.2.2.2. Ağırlıklar	24
2.2.2.3. Toplama Fonksiyonu	25
2.2.2.4. Aktivasyon Fonksiyonu	26
2.2.2.5. Hücrenin Çıktısı	29
2.2.3. Yapay Sinir Ağlarının Yapısı	29
2.2.3.1. Girdi Katmanı	29
2.2.3.2. Ara (Gizli) Katmanlar	29
2.2.3.3. Çıktı Katmanı	29
2.3. Yapay Sinir Ağlarının Çalışması	30
2.3.1. Yapay Sinir Ağlarının Eğitilmesi ve Testi	30
2.3.2. Öğrenme Stratejileri	32

2.3.2.1.	Öğretmenli (Supervised) Öğrenme	32
2.3.2.2.	Destekleyici (Reinforcement) Öğrenme	32
2.3.2.3.	Öğretmensiz (Unsupervised) Öğrenme	33
2.3.2.4.	Karma Stratejiler	33
2.3.3.	Öğrenme Kuralları	33
2.3.3.1.	Çevrimiçi (On-Line) Öğrenme Kuralları	33
2.3.3.2.	Çevrimdışı (Offline) Öğrenme	34
2.3.3.3.	Öğrenme Kurallarından Bazıları	34
2.3.3.3.1	Hebb Kuralı	34
2.3.3.3.2	Hopfield Kuralı	34
2.3.3.3.3	Delta Kuralı	35
2.3.3.3.4	Kohonen Kuralı	35
2.4	Yapay Sinir Ağlarının Avantajları	35
2.5	Yapay Sinir Ağlarının Dezavantajları	39
3.	YAPAY SİNİR AĞI MODELLERİ	40
3.1.	İlk Yapay Sinir Ağları	40
3.1.1.	Tek Katmanlı Algılayıcılar (TKA)	40
3.1.1.1.	Basit Algılayıcılar (Perseptron)	43
3.1.1.2.	ADALINE/MADALINE Modeli	45
3.2.	Çok Katmanlı Algılayıcı ve Geriye Yayılım Yapay Sinir Ağı	47
3.2.1.	Çok Katmanlı Algılayıcıların Yapısı	48
3.2.2.	Geri Yayılım Ağlarının Öğrenmesi	51
3.3.	Yapay Sinir Ağları Destekleyici Öğrenme: Doğrusal Vektör Parçalama/Niceleme Modeli (LVQ)	56
3.3.1.	LVQ2 Ağı	60
3.3.2.	Ceza Mekanizmalı LVQ	61
3.3.3.	LVQ-X Modeli:	62
3.4.	Yapay Sinir Ağı Modeli (Öğretmensiz Öğrenme) Adaptif Rezonans Teori (ART) Ağları	62
3.4.1.	ART1 Ağı:	66
3.5.	Jordan Ağları	71
3.6.	Elman Ağı:	72
3.7.	Hopfield Ağı	75
3.8.	Kohonen Ağları	80
4.	ÇOK KATMANLI YAPAY SİNİR AĞI TASARIMI	84
4.1	Eğitim ve Test Kümelerinin Seçimi:	84
4.2	Katman ve Nöron Sayısının Seçimi	85
4.3	Girdilerin Ölçeklendirilmesi	86
4.4	Fonksiyon Seçimi	86
4.5	Performans Fonksiyonunun Belirlenmesi	87
4.6	Öğrenme Oranı ve Momentum Katsayısının Belirlenmesi	87
4.7	Ağın Devir (Epoch) Sayısı	88
5.	LOJİSTİK REGRESYON	89
5.1.	İkili Lojistik Regresyon Modeli	91
5.2.	Modelin Katsayılarının Testi ve Yorumlanması	94
5.2.1.	Olabilirlik Oran Testi	94
5.2.2.	Wald Testi	95

5.3.	Modelin Uyum İyiliğinin Ölçülmesi	97
6.	UYGULAMA	99
6.1.	Veri Seti.....	99
6.2.	Yapay Sinir Ağları Analiz Sonuçları.....	102
6.2.1.	Ara Katmandaki En Uygun Nöron Sayısının Bulunması	103
6.2.2.	En Uygun Öğrenme Oranı ve Momentum Katsayısının Belirlenmesi 105	
6.2.3.	En Uygun Aktivasyon Fonksiyonunun Bulunması.....	108
6.2.4.	İki Ara Katmanlı Ağ Yapısı Denemesi:	109
6.3.	Lojistik Regresyon Analizi Sonuçları	110
6.4.	ROC Eğrileri (Receiver Operating Characteristic Curves) Analizi	118
6.4.1.	ROC Eğrileri	118
6.4.2.	Roc Eğrisi Analiz Sonuçları.....	121
7.	SONUÇLAR VE DEĞERLENDİRME	124
	KAYNAKÇA.....	126
	EKLER.....	134

TABLO LİSTESİ

Tablo 1.1: Doğal Zeka ile Yapay Zeka Karşılaştırması	6
Tablo 2.1: Toplama Fonksiyonu Örnekleri	25
Tablo 2.2: Eşik Değer Aktivasyon Fonksiyonu	28
Tablo 6.1: Değişkenlerin Betimsel İstatistikleri	101
Tablo 6.2: Ara Katmandaki En Uygun Nöron Sayısı için Deneme Sonuçları	104
Tablo 6.3: En Uygun Öğrenme Oranı ve Momentum Katsayısı Deneme Sonuçları	106
Tablo 6.4: En Uygun Aktivasyon Fonksiyonu Deneme Sonuçları	108
Tablo 6.5: İki ara katman Deneme Sonuçları	109
Tablo 6.6 : En Uygun Ağ Yapısı	109
Tablo 6.7: YSA Sınıflandırma Sonuçları	110
Tablo 6.8: LR Hedef Değişken Betimsel İstatistikleri	111
Tablo 6.9: Başlangıç Adımı Sınıflandırma Tablosu	111
Tablo 6.10: Başlangıç Adımı Anlamlılık Testi	112
Tablo 6.11: Model katsayılarının Omnibus Testleri	112
Tablo 6.12: Model Özet Tablosu	113
Tablo 6.13: Hosmer Lemeshow Uyum İyiliği Testi	113
Tablo 6.14: LR Sınıflandırma Tablosu	114
Tablo 6.15: LR Model Katsayıları ve Anlamlılık Testleri	116
Tablo 6.16: Genel Sınıflandırma Matrisi	120
Tablo 6.17: LR Test Kümesi Sınıflandırma Tablosu	121
Tablo 6.18: ROC Eğrisi Anlamlılık Testi	122
Tablo 6.19: ROC eğrisi Duyarlılık ve Özgüllük Değerleri	123

ŞEKİL LİSTESİ

Şekil 1.1: Turing Test Yapısı	11
Şekil 1.2: Uzman Sistemin Yapısı	14
Şekil 1.3: Üyelik Fonksiyonu Örnekleri (Nabiyev, 2003 s.669).....	18
Şekil 2.1: Beyin Yapısı	23
Şekil 2.2: Yapay Sinir Hücresinin Yapısı	24
Şekil 2.3: Lineer Aktivasyon Fonksiyonu.....	27
Şekil 2.4: Step Aktivasyon Fonksiyonu	27
Şekil 2.5: Sigmoid Aktivasyon Fonksiyonu	28
Şekil 2.6: Eşik Değer Aktivasyon Fonksiyonu	28
Şekil 2.7: Hiperbolik Tanjant Aktivasyon Fonksiyonu.....	29
Şekil 2.8: Bir Yapay Sinir Ağı Örneği	30
Şekil 3.1: İki Girdi ve Bir Çıktı'dan oluşan TKA modeli.....	40
Şekil 3.2: Ağırlık ve Sınıfları Birbirinden Ayıran Doğrunun Geometrik Gösterimi .	42
Şekil 3.3: Bir basit algılayıcı yapısı	43
Şekil 3.4: Bir ADALINE yapısı	46
Şekil 3.5: LVQ Ağının Topolojik Yapısı.....	57
Şekil 3.6: ART ağlarının genel yapısı	64
Şekil 3.7: ART Ağının Çalışma Şekli	65
Şekil 3.8: ART1 Modeli	67
Şekil 3.9: Jordan Ağı.....	71
Şekil 3.10: Elman Ağı	72
Şekil 3.11: Hopfield ağının yapısı.....	76
Şekil 3.12: Kohonen Ağı.....	81
Şekil 4.1: Eğitim Kümesi ve Onaylama Kümelerinin Hatalarının Karşılaştırılması	84
Şekil 6.1: En Uygun Yapay Sinir Ağı Modeli (15-29-1)	107
Şekil 6.2: İdeal ve Kötü Performans Göstergesi Olan ROC Eğrileri.....	119
Şekil 6.3: ROC Eğrisi Grafığı	122

KISALTMA LİSTESİ

ADALINE	Uyabilen Doğrusal Elemanlar
ART	Adaptive Resonance Theory
GA	Genetik Algoritmalar
LMS	En Küçük Kareler Kuralı
LR	Lojistik Regresyon
LVQ	Doğrusal Vektör Parçalama/Niceleme Modeli
MADALINE	Çoklu Uyabilen Doğrusal Elemanlar
PBNN	Olasılık Tabanlı Yapay Sinir Ağları
RBN	Radyal Tabanlı Yapay Sinir Ağları
SOM	Self-Organizing Maps
TKA	Tek Katmanlı Algılayıcılar
VLSI	Büyük Ölçekli Entegre Devre Teknolojisi
YSA	Yapay Sinir Ağları
MSE	Ortalama Hata Kareleri

EKLER

EK-1:Değişkenler Arasındaki Spearman ve Kendall Tau-b Korelasyon	
Analizleri.....	134
EK-2 :Ara Katmandaki En İyi Nöron Sayısının	
Belirlenmesi.....	135
EK-3:En Uygun Öğrenme Oranı ve Momentum Katsayısının	
Belirlenmesi.....	136

GİRİŞ

Yapay Zekanın bir uygulaması olan YSA, son yıllarda geleceğin tahmin edilmesinde; özellikle doğrusal olmayan (non-lineer) problemlerin çözülmesinde kullanılan çok önemli bir tekniktir.

YSA, insanda var olan sinir hücrelerinin çalışma şeklinin, taklit edilmesi ile üretilmiş matematiksel bir modeldir. Yapay Zeka (YZ), dolayısıyla YSA teknolojileri, henüz insan beyninin, %1'ini bile simüle etmekten uzaktır. Fakat elektrokimyasal olarak çalışan doğal sinir hücrelerinden çok daha hızlı çalışabilmesi ve yorulmaması gibi avantajları nedeniyle, tahmin yapmada oldukça başarılıdır. YSA'lar da, insan beynini örnek aldıkları için öğrenerek, hata yapıp düzelterek sonuca ulaşırlar.

Kredi analizinin amacı, kredi talebinde bulunanların, geri ödemesini yapıp yapamayacağını tahmin etmektir. Kredilerin doğru şekilde kullanılması, finansal sistemin devamlılığı ve ekonominin gelişmesi için önemlidir. Geçmişte verilmiş olan ve geri ödemesini yapan ya da yapmayan örneklere göre, bir sistem oluşturulmuş ve kuruluşlar bu şekilde yeni kredileri kabul/red şeklinde değerlendirmektedirler. Bu nedenle, kredi taleplerinin değerlendirilmesi bir sınıflandırma problemidir denebilir.

Çalışmanın birinci bölümünde, YZ konusu incelenmiştir. YZ'nin tarihsel gelişimi ve YSA'dan başka diğer öne çıkan diğer teknolojileri anlatılmıştır.

Çalışmanın ikinci bölümünde, öncelikle YSA'nın tarihçesi işlenmiş ve biyolojik sinir hücresinin yapısından ardından yapay sinir hücrelerinin çalışma şekli anlatılmıştır. Yapay sinir hücrelerinin bileşenleri tanıtılmış, ardından YSA'nın eğitimi detaylı olarak incelenmiştir. Eğitim kapsamına giren öğrenme metotları, hata fonksiyonları, öğrenme katsayısı ve öğrenme kurallarından bu bölümde bahsedilmiştir tanıtılmıştır. Bu bölümde son olarak da, YSA'nın avantaj ve dezavantajlarından bahsedilmiştir.

Üçüncü bölümde ilk YSA modellerinden başlayarak çok katmanlı geri yayılım ağları ve diğer önemli ağlar anlatılmıştır.

Dördüncü bölümde, YSA tasarımında önemli olan noktalardan bahsedilmiş ve ağ topolojisinin seçimi, katman ve nöron sayısının belirlenmesi, aktivasyon, birleştirme ve performans fonksiyonlarından bahsedilmiştir.

Beşinci bölümde, kısaca Lojistik Regresyon Analizi teorisinden bahsedilmiştir.

Altıncı bölümde ise, özel bir şirketten alınan araba kredisi talepleri verisi, YSA ve Lojistik Regresyon Analizleri ile sınıflandırma yapılarak karşılaştırılmıştır. Elde edilen sonuçlar ile ROC Eğrisi analizleri yapılmıştır. Bu çalışmada yapılan uygulamada, YSA'nın, Lojistik Regresyondan daha iyi bir tahmin edici yöntem olduğu ortaya çıkmıştır.

Yedinci bölüm olan sonuç bölümünde ise sonuçlar özetlenerek, yorumlanmıştır.

1. YAPAY ZEKA

Zeka, İnsanın düşünme, akıl yürütme, objektif gerçekleri algılama, yargılama ve sonuç çıkarma yeteneklerinin tamamı olarak tanımlanmaktadır¹. Yapay zeka ise insana özgü bu özelliklerin analiz edilip makinelere kazandırılmasıdır.

Yapay zeka kökenlerini antik çağdan alan, uzun bir gelişim sürecine sahiptir. Bu süreçte, her şeyden önce iki gelişim çizgisi büyük öneme sahiptir: Bunlardan ilki olan yapay insan miti, yani ikinci bir insan türünün yaratılması düşüncesi yüzyıllara dayanır. İkincisi olan insana özgü akıl yetilerinin makineleştirilmesi düşüncesinin kökeni yazı dilinin, özellikle yazılı hesaplamanın kültürel gelişimiyle sağlanmış ve biçimselleştirme hakkında farklı kuramlara, sonuçta ise “düşüncenin bir hesaplama biçimi” olduğu (Sybille Kraemer) yönündeki, modern bilişsel bilimlerde de belirgin olan anlayışa yol açmıştır. Üçüncü ve modern bir gelişim çizgisi olarak, bugünkü dijital bilgisayarların da temelini oluşturan klasik programlanabilir evrensel hesaplama makinesinin icadı gelir. Bu makineler, düşüncenin hesaplama işlemine dönüştürülmesinin başlangıcı olmuştur².

1930’larda temel matematik alanında çığır açan İngiliz Alan Turing, ilk İngiliz elektronik bilgisayarının yapımında yer almıştır. 1950’de “Computing Machinery and Intelligence” adlı makalesinde makinelerin düşünüp düşünemeyeceği sorusunu cevaplamak için “Turing Testi” adında bir test önermiştir. Turing, “yapay zeka” disiplininin babası olarak görülebilir³. Yapay zeka terimi ise ilk defa 1956 yılında Dartmouth College’de çalışmalarını Princeton Üniversitesi’nde sürdüren Mc Carthy, Minsky, Shannon ve Rochester’in düzenledikleri bir konferansta Mc Carthy tarafından kullanılmıştır⁴.

Yapay zekayı bazı ünlü bilim adamları şöyle tanımlamıştır:

¹ Türk Dil Kurumu, Türk Dil Kurumu Sözlüğü, (Çevrimiçi) www.tdk.gov.tr.

² Görz, Günther.,Nebel Bernhard., Yapay Zeka, Çev.Özgür POZAN, İstanbul,İletişim Yayınları,2005, s.9.

³ Ae,s.10.

⁴ Erhan Altuntaş, Tuncay ÇELİK, Yapay Zeka (2003),(Çevrimiçi) 2010, <http://bagem.tripod.com/>

“Bilgisayarları düşündürmeye ... kelimenin tam anlamıyla zihne sahip makineler yapmaya çalışan yeni ve heyecan verici bir çaba” (Haugeland, 1985).

“Karar verme, problem çözme, öğrenme ... gibi insan düşüncesi ile ilişkilendirilen aktivitelerin otomasyonu” (Bellman, 1978).

“İnsanların zekalarını kullanarak gerçekleştirdiği fonksiyonları gerçekleştiren makineleri yapma sanatı” (Kurzweil, 1990).

“İnsanların halihazırda daha iyi olduğu işleri bilgisayarların nasıl yapabileceğine dair yürütülen çalışma” (Rich ve Knight, 1991).

“Zihinsel yeteneklerin hesaplamalı modeller aracılığıyla incelenmesi” (Charniak ve McDermott, 1985).

“Algılamayı, düşünmeyi ve davranmayı mümkün kılan bilgisayar modellerinin incelenmesi ” (Winston, 1992).

“İnsan yapımı şeylerde akıllı davranışın incelenmesi ” (Nilsson, 1998).

Bu tanımlara göre, özetle yapay zeka,

- İnsan gibi düşünen
- İnsan gibi davranan
- Akılcı düşünen
- Akılcı davranan

bilgisayar sistemleri ile ilgilenir⁵.

Diğer bir deyişle yapay zeka, hem bilişsel sistemleri simule etmeyi (akılcı/insan gibi düşünme) hem de akıllı sistemleri yapılandırmayı (akılcı/insan gibi davranma)

⁵ Stuart J. Russel, Peter Norving, Artificial Intelligence A Modern Approach, New Jersey, Prentice Hall, 1995, s.5.

kendine amaç edinmiş bir bilimsel disiplindir. Her iki çabanın da ortak yönü, bilginin akılcı ve insanda olduğu gibi işlenmesinin ilkelerini araştırıp bulmaktır⁶.

Yapay zeka araştırmalarında insan beynini model alır. Fakat insan beyni algılama, iç organların denetimi, duygular, hareket gibi bir çok görevi aynı anda yapar. Fakat yapay zeka, beynin sadece zeka kısmı ile ilgilenir⁷. Temel bir yapay zeka sistemi ise zekanın şu üç temel özelliğini göstermelidir: algılama, kavrama/ idrak, eylem⁸.

⁶ Görz ve Nebel,a.g.e,s.11

⁷ Nadir Bencan, Yapay Beyin Değil, Yapay Zeka (2007),(Çevrimiçi),www.yapay-zeka.org,2010.

⁸ Ercan ÖZTEMEL, Yapay Sinir Ağları, Papatya Yayıncılık, 2.bs., İstanbul,2006,s.20-21.

Özellik	Yapay Zeka	Doğal Zeka
Kalıcılık	Kalıcıdır. Programlarda ve veri tabanında değişiklik yapılmadıkça silinmez.	Zamanla unutulabilir.
Kopyalanabilme	Sayısal bilgi olduğu için kolaylıkla kopyalanabilir ve başka yerlere transfer edilebilir	Bir insanın kendi uzmanlığını, kendi zekası ile öğrendiğini başkasına aktarması kolay değildir. Bir eğitim süreci sonunda eğitilenin kapasitesi kadar aktarabilir.
Maliyet	Sayısal bilginin ilk defa üretilmesi maliyetli olabilir fakat kopyalama maliyeti çok düşüktür.	Bilgiler sayısal olmadığı, insana bağlı olduğu için herkese ayrı ayrı öğretmek maliyetlidir.
Tutarlılık	Tutarlıdır, aynı koşullarda sonuç hep aynı olur.	Kişinin ruh haline göre bile varacağı kararlar farklı olabilir.
Yaratıcılık	Yoktur. Ne öğrendiyse, parametreler ne ise sadece ona yanıt verir.	Yaratıcılık ve bir şeyden yararlanarak başka şeyleri de üretebilme öğrenebilme yeteneği vardır.
Muhakeme Gücü	Öğrendiği kadarı ile dar bir alanda çalışır.	Muhakeme gücü vardır. Bir konudaki problemi çözmek için başka konulardaki tecrübelerinden de yararlanabilir.

Tablo 1.1: Doğal Zeka ile Yapay Zeka Karşılaştırması⁹

Bu bölümde, yapay zekanın tarihi ve yapay zekaya bağlı teknolojiler anlatılacaktır.

⁹Vasif V. Nabiyev, Yapay Zekâ – Problemler, Yöntemler, Algoritmalar, Seçkin Yayınevi, 1.bs.Ankara,2003,s.94.

1.1. Yapay Zekanın Tarihi

İnsana benzer makinelerin, robotların oluşturulması düşüncesi antik çağdan beri insanların ilgisini çekmektedir. Yunan mitolojisinde yapılan heykelleri canlandırma izlerine rastlanır. M.Ö. 2500 yılında Eski Mısır'da konuşan heykellerden bahsedilmiştir¹⁰. George F. Luger, *Yapay Zeka*¹¹ adlı kitabında, yapay zekanın temellerini Aristo (M.Ö. 384-322)' ya dayandırır. Çünkü, akıl- mantık ilişkisini tarihte ilk defa Aristo kurmuştur: *Politika* adlı yapıtında, emirlere itaat eden ve gelecekle ilgili planlar yapan, efendiyi de kölelerinden vazgeçirecek otomatlardan (robotlardan) bahsetmiştir.

15. yy'da ilk modern ölçme aleti olan saate, kurma mekanizması eklenmiştir. 16. yy'da, bu mekanizma geliştirilerek ilk mekanik hayvanlar yapılmıştır (Örneğin, Da Vinci'nin Yürüyen Aslan'ı (walking lion,1515)). 17. yüzyılda Descartes, hayvan vücudunun karmaşık bir makineden başka bir şey olmadığını öne sürmüştür. Pascal ise ilk mekanik hesap makinesini yapmış, Leibnitz de bunu çarpma ve bölme için geliştirmişti¹². 19. yy'da yapay zeka ile ilgili sayısız eserin yanında, Mary Wollstonecraft Shelley'in "*Frankenstein ya da Modern Prometheus*" adlı romanındaki yapay olarak üretilmiş insan prototipi, içerdiği eleştirel yaklaşım nedeniyle önemlidir¹³. Aynı yüzyılda, Charles Babbage, astronomici John Herschel ile birlikte mühendislerin ve gemicilerin kullanımı için bir matematik tablosu hazırlama işine girişmişlerdi. Farklar Makinesi adı verilen bu makine aslında 25.000 mekanik parçadan oluşan bir hesap makinesiydi. Fakat 19. yy'ın teknolojisiyle farklar makinesinin yapımı imkansızdı. Babbage, bunun üzerine Farklar Makinesi' ni bırakıp Analitik Makine üzerinde çalışmaya başladı. Bu makine, mekanik parçalardan oluşmasına rağmen bu günkü bilgisayarlarda yer alan bellek ve işlemciye karşılık gelen birimler içeriyordu ve bu işlemci programlanabilme özelliğine sahipti. Babbage, Analitik Makine'yi de tamamlayamadı. Ada Byron

¹⁰ Uğur Halıcı, Yaman Kayıhan, Yapay Zeka,(Çevrimiçi),www.yapay-zeka.org,2010.

¹¹ George F. Luger,Artificial Intelligence Structures and Strategies for Complex Problem Solving,England, Fifth Edition,2005, s.5.

¹²(www.aaai.org, 2010)

¹³ Luger,a.g.e.,s.4.

(Lady Lovelace), Babbage ile yazılarak Analitik Makinenin Bernoulli sayılarının hesaplanmasında kullanılması için Babbage'a bir plan çıkarmasını önermişti. Bu "plan" geliştirilen "ilk bilgisayar programı" olarak anılmaktadır¹⁴.

1910 yılında, Bertrand Russell ve Alfred North Whitehead Principia Mathematica adlı kitaplarında biçimsel mantık kavramına yeni bir boyut getirdiler. 1921'de Karel Capek'in tiyatro oyunu "R.U.R."da robot kelimesi ilk kez kullanıldı. 1936'da Alan Turing, "On Computable Numbers with an Application to the Entscheidungsproblem" adlı çalışmasında, soyut bir "bilgisayarı", insanın kalem ve kağıtla yapabildiği her işlemi gerçekleştirebilen bir makineyi tanımlıyordu. Fakat model karar verme problemine çözüm getiremiyordu. 1943'de, yapay zekanın kuruluşunda bir diğer hareket noktası olan Warren McCulloch ve Walter Pitts'in "A Logical Calculus of the Ideas Immanent in Nervous Activity" çalışması yayınlandı. McCulloch ve Pitts basitleştirilmiş sinir hücresi ağlarının Turing hesaplanabilirliğinin de hesaplayabildiği bir fonksiyon ağı hesapladığını ve hesaplanabilir her fonksiyonun bir sinir hücresi ağı tarafından hesaplanabileceğini göstermektedir. 1950'de Turing, "Computing Machinery and Intelligence" adlı makalesinde "Turing Testi" ile insanın makineden nasıl ayrıldığını konu etti. Yapay zekanın yeni bir bilim olarak görülmesi ise ilk kez 1956 yılındadır. 1956'da IBM tarafından Dartmouth Koleji'nde yapılan bir seminere Marvin Minsky, Allen Newell ve Herbert Simon katılmış ve aynı toplantıya katılan John McCarthy bu alandaki çalışmalara ilk defa "yapay zeka" adını verdi. İlk kuram ispatlayan programlardan Mantık Kuramcısı (Logic Theorist) burada Newell ve Simon tarafından tanıtıldı. 1950'lerin sonlarında Newell ve Simon, "insan gibi düşünme" yaklaşımına göre üretilmiş ilk program olan Genel Sorun Çözücü (General Problem Solver) 'yü geliştirdi. Simon, daha sonra fiziksel simge varsayımını ortaya atmış ve bu kuram, insandan bağımsız zeki sistemler yapma çalışmalarıyla uğraşanların hareket noktasını oluşturdu. 1958'de şekil tanıma ve kendi kendine adapte olan sistemler üzerine çalışan Rosenblatt'ın algılayıcılar (perceptrons) üzerine çalışmaları önem

¹⁴ Halıcı ve Kaythan, ag.e.

kazandı. Bu dönemde aynı zamanda, önemli yapay zeka programlama dillerinden biri olan LISP, MIT’de John McCarthy tarafından geliştirildi¹⁵.

Bundan sonraki dönemde yapay zekanın gelişiminde dört önemli dönem vardır.

- 1) 1950’lerin sonu 1960’ların başındaki ilk dönemde, basit yap-bozların çözümü, mantıksal ve geometrik önermelerin kanıtlanması, satranç gibi oyunlarla uğraşılmıştır. “Power Based Approach” kavramıyla tanımlanan bu ilk dönemde, bu yöntemden çok şey beklenmiş fakat bunun problemleri çözmekteki başarısının sınırlı olduğu görülmüştür.
- 2) 1960’larda Amerikan Üniversitelerinde yapay zekanın, dilin işlenişi, otomatik problem çözme ve görsel sahne çözümlemesi gibi özellikleri incelenmeye başlanmıştır. Amerikan Savunma Bakanlığı’nın desteğiyle yapay zeka çok ilerlemiştir.
- 3) 1970’li yıllarda, entegre robot sistemleri, uzman sistemler yapay zekanın en çok gelişen kısımlarıdır. Uzman sistemler ile “Power Based Approach” anlayışının tersine, problem çözümüne yönelik biçimselleştirilmiş bilginin ve özel işleme tekniklerinin kullanımı ön plana çıkmıştır (Bilgi Temelli Yaklaşım). Araştırmalarda, günlük dilin tanınip anlaşılması, kimyadaki çözümleme ve sentez, tıbbi teşhis ve tedavi, mineralojideki öngörüler, teknik sistemlerde kurulum ve hata çözümlemesi gibi konularda araştırmalara ağırlık verilmiştir. Başta Almanya ve İngiltere olmak üzere Avrupa Üniversitelerinde yapay zeka araştırma grupları kurulmuştur.
- 4) 1980’lerde çalışmalar büyük ölçüde matematikleştirilmiş, bilgi işleme düşüncesi öne çıkmış ve durumsallık bölünmüş YSA gibi konular öne çıkmıştır. 1990’lardan beri tümleşik yaklaşımlara ve buna bağlı olarak “bilgi işlem” ve “akıllı sistem” kavramlarının yeni konulara dönüşmelerine yönelik çalışmalar yapılmıştır¹⁶.

¹⁵ Şakir Kocabaş, Yapay Zeka Araştırma ve Uygulama Alanları,(Çevrimiçi),<http://www.sakirkocabas.com/2010>, s.3.

¹⁶Görz ve Nebel,a.g.e,s.31-33

1.1.1 Turing Testi

Alan Turing (1912 - 1954) İngiliz matematikçi ve bilgisayar bilimcisiydi. II. Dünya Savaşı sırasında o ve bir kaç entelektüel Alman şifrelerinin kırılmasında çok önemli bir rol oynamıştır. 1936 yılında yayınladığı “On Computable Numbers with an Application to the Entscheidungsproblem” makalesi ile matematiğin doğasına ilişkin anlayışta devrim yaratmıştır¹⁷. Bu makalenin konusu matematiksel mantığın soyut bir problemi ile ilgilidir ve bu problemi çözerken Turing bugün Turing Makinesi diye adlandırılan, program depo eden genel amaçlı bilgisayarı kuramsal olarak icat etmeyi başarmıştır.

Turing Testi, Alan Turing’in “Bilgisayarlar düşünebilir mi” sorusunu sınamak için geliştirdiği bir testti. 1950 yılında felsefe derginde yayınlanan “Computing Machinery and Intelligence” adlı makalesinde Turing, bir makinenin düşünmesi ile ilgili açıklamaların mantıksal olarak mümkün olup olmadığını açıklıyordu¹⁸. Makalenin, şifre kırıcı ve bilgisayar programcısı olarak çalışmış bir matematikçi tarafından bir felsefe dergisinde yayınlanması çok ilginçti¹⁹. Turing testinde denek, sorgulayıcıyla bir terminal aracılığıyla haberleştiriliyordu. Eğer sorgulayıcı, deneğin insan mı ya da bir makine mi olduğunu anlayamazsa denek Turing testini geçmiş sayılıyordu.

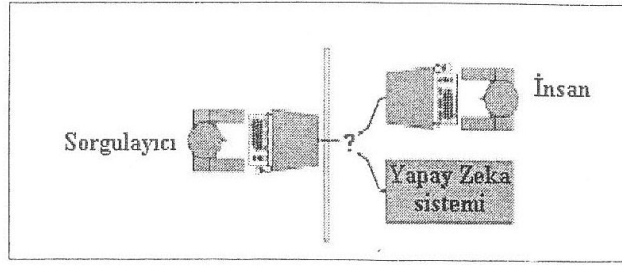
Belli bir uzmanlık alanı ile ilgili istatistiki bilgilere dayanarak, Turing testini kısmen de olsa geçebilen programların yazılması mümkündür. SHRDLU (T. Winograd), ELIZA (J. Weizenbaum) gibi programlar, tamamiyle insan zekasını temsil etmeseler de Turing Testini basit bir biçimde geçtikleri söylenebilir. Örneğin ELIZA programı kullanıcıdan aldığı cevaplar doğrultusunda sorular sormakta veya cümleler oluşturmaktadır. Fakat burada anlama söz konusu değildir²⁰. Dolayısıyla makinelerin Turing Testini geçebilmeleri için daha zaman vardır.

¹⁷Whitby Blay., Yapay Zeka, Çev.Çiğdem KARABAĞLI, İstanbul,2005, s.30

¹⁸Nabiyev,a.g.e.,s.83

¹⁹Whitby, a.g.e.s.31

²⁰Nabiyev, a.g.e.s.85



Şekil 1.1: Turing Test Yapısı²¹

1.1.2. Çin Odası Deneyi

Makinelerin Turing Testini geçebilseler bile hiç bir zaman düşünüp anlama yeteneğine sahip olamayacaklarını savunan Berkeley Üniversitesinden John Searle, Çin Odası Testi'ni önermekteydi.

Bir odada kilitli olduğunuzu düşünün ve odada da üzerlerinde Çince tabelalar bulunan sepetler olsun. Fakat siz Çince bilmiyorsunuz. Ama elinizde Çince tabelaları İngilizce olarak açıklayan bir kurallar kitabı bulunsun. Kurallar Çinceyi tamamen biçimsel olarak, yani söz dizinlerine uygun olarak açıklamaktadır. Daha sonra odaya başka Çince simgelerin getirildiğini ve size Çince simgeleri odanın dışına götürmek için, başka kurallarda verildiğini varsayın. Odaya getirilen ve sizin tarafınızdan bilinmeyen simgelerin oda dışındakilerce 'soru' diye, sizin oda dışına götürmeniz istenen simgelerin ise 'soruların yanıtları' diye adlandırıldığını düşünün. Siz kilitli odanın içinde kendi simgelerinizi karıştırıyorsunuz ve gelen Çince simgelere yanıt olarak en uygun Çince simgeleri dışarı veriyorsunuz. Dışta bulunan bir gözlemcinin bakış açısından sanki Çince anlayan bir insan gibisiniz. Çince anlamanız için en uygun bir program bile Çince anlamanızı sağlamıyorsa, o zaman herhangi bir sayısal bilgisayarın da Çince anlaması olanaklı değildir. Bilgisayarda da sizde olduğu gibi, açıklanmamış Çince simgeleri işleten bir biçimsel program vardır ve bir dili anlamak demek, bir takım biçimsel simgeleri bilmek demek değil, akıl durumlarına sahip olmak demektir²².

²¹A.e.g.,s.83

²² Altuntaş ve Çelik,e.g.e.

1.2. Yapay Zeka Teknolojileri

Değişik yerlerde olayların farklı yönleri, insanları ilgilendirmektedir. Bilgisayarların insanların karar verme ve problem çözme mekanizmalarını taklit etmesinin sağlanması da dolayısıyla farklı teknolojilerin doğmasına neden olmaktadır. Günümüzde 60'dan fazla yapay zeka teknolojisinin bahsedilmektedir²³.

Genel olarak en bilinen yapay zeka teknolojileri şöyledir:

1.2.1. Uzman Sistemler

Uzman sistemler, bir problemi o konuda uzman olan kişilerin çözebildiği gibi çözebilen ve bu bilgiyi belli bir formatta saklayabilen bilgisayar programlarıdır. Bunun için bu sistemler Bilgiye Dayalı Sistemler (Knowledge Based Systems) olarak da adlandırılırlar²⁴.

1960'lı yıllarda, yapay zeka üzerine çalışan araştırmacılar, genel amaçlı programlar üretmenin çok zor ve yararsız olduğunu fark ettiler. Bunun yerine, bir konuda özelleşmiş programlar üretmeye karar verdiler²⁵. İlk geliştirilen uzman sistemlerden biri 1970'lerde tıpta bazı hastalıkların teşhisini yapabilen MYCIN programıdır²⁶. Bu gün ise tıp, bankacılık, kimya, uzay araştırmaları, coğrafya, savunma gibi pek çok alanda kullanılmaktadır.

Bir uzman sistem temelde beş elemandan oluşur²⁷:

²³ Öztemel, a.g.e.,s.15

²⁴ Graham Winstanley, Artificial Intelligence in Engineering, New York,1991.

²⁵ N.P.Padhy, Artificial Intelligence and Intelligent Systems,USA,2005.

²⁶ Kocabaş, a.g.e.,s.5.

²⁷ Mahmut GÜLESİN, Murat Tolga ÖZKAN, “Uzman Sistem Yaklaşımı ile Cıvata ve Disli Çark Seçimi”, Türk J Engin Environ Sci.25 (169 – 177) , TÜBİTAK 2001,s.170.

1.2.1.1. Bilgi Tabanı

İlgili alanla ilgili bütün bilgi ve tecrübelerin kurallar (Eğer.. ise o zaman...), bilgi çatıları, bilgi sınıfları ve presedürler halinde bulunduğu veri tabanıdır.

1.2.1.2. Çıkarım Mekanizması

Uzman sistemin bir anlamda beynidir. Uzman sistemin, yorumlayan, muhakeme eden, sonuç üreten kısmıdır. Sistemin veri tabanındaki verileri nasıl kullanacağını yorumlayan birimdir²⁸. Genel olarak iki tür çıkarım vardır²⁹: ileri doğru zincirleme (başlangıçtaki gerçekten hareketle sonuca gitmek) ve geri doğru zincirleme (sonuçtan hareketle başlangıçtaki gerçeğe gitmek).

1.2.1.3. Kullanıcı Arabirimi

Uzman sistem ile kullanan kişilerin iletişimini sağlayan bölümdür. Problemlere üretilen sonuçlar için *Neden?* ve *Niçin?* sorularına cevap vererek açıklar.

1.2.1.4. Bilgiyi Alma Ünitesi

Kullanıcıya veri tabanındaki bilgileri düzeltme, alma, ekleme, çıkarma, silme gibi işlemlerin yapılmasına olanak sağlayan birimdir³⁰.

1.2.1.5. Açıklama Ünitesi

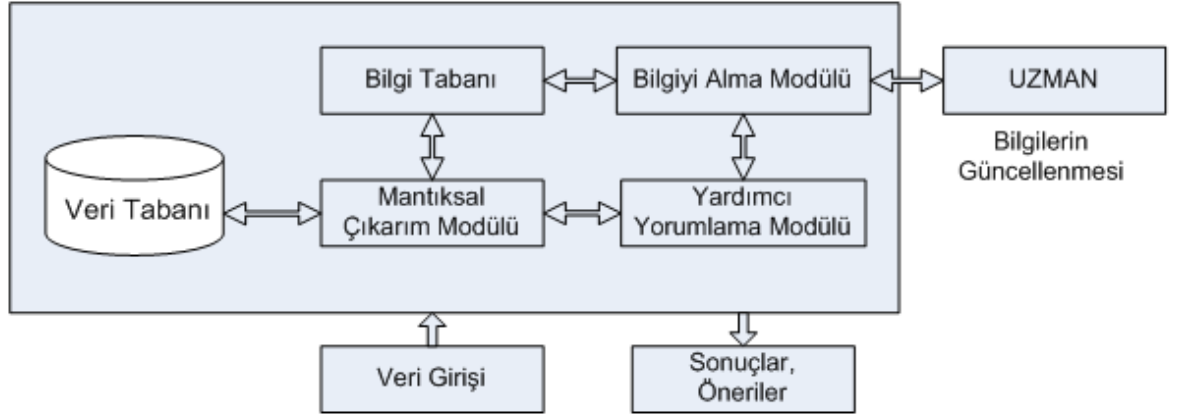
Çıkarımların nasıl yapıldığını açıklayan birimdir. Çalışma anında sorular sorarak kullanıcıyı açıklamalı olarak yönlendirir³¹.

²⁸ Ömer YILDIZ, Döviz Kuru Tahmininde Yapay Sinir Ağlarının Kullanımı, Eskişehir Osmangazi Ü.İşletme ABD Yüksek Lisans Tezi, 2006, s.42

²⁹ Öztemel, a.g.e., s.15-16.

³⁰ Gülesin ve Özkan, a.g.e., s.170.

³¹ Winstanley, a.g.e.



Şekil 1.2: Uzman Sistemin Yapısı³²

1.2.2. Makine Öğrenmesi

Makine öğrenmesi, bilgisayarların bir olay ile ilgili bilgileri ve tecrübeleri öğrenerek gelecekte oluşacak benzeri olaylar hakkında kararlar verebilmesi ve problemlere çözümler üretebilmesidir³³. Öğrenmede sadece örneklerden öğrenme değil aynı zamanda takviye öğrenimi ve öğretmenle öğrenim söz konusudur. Bir öğrenme algoritması veri kaynakları ve onun beraberinde yer alan giriş bilgi ve sonuçlardan oluşur. Makine öğrenimi, önceki olay örneklerini ve sonuçlarını inceleyerek, yeni bir benzer olay için genellemeler yapar. Bir makine öğrenimi sistemi eğitim kümesi adı verilen veri seti kullanır. Bu set içinde örnek gözlem kodları bulunan ve makine tarafından okunabilen bazı formlar bulunur³⁴.

YSA ile öğrenen bilgisayarların da bilgi sahibi olması örnekler yolu ile sağlanmaktadır. YSA dışındaki bazı öğrenme şekilleri şöyle sıralanabilir³⁵.

- Alışkanlık yolu ile öğrenme
- Görerek öğrenme
- Talimatlardan öğrenme

³²Nabiyev,a.g.e.

³³ Öztemel,a.g.e.,s.21

³⁴ Ömer Akgöbek, Fuat Çakır,Veri madenciliğinde bir uzman sistem tasarımı,Harran Üniversitesi Akademik Bilişim'09,2009, Şanlıurfa,s.803

³⁵ Öztemel,a.g.e.,s.22.

- Örneklerden öğrenme
- Analoji yolu ile öğrenme
- Açıklamalardan öğrenme
- Deney yolu ile öğrenme
- Keşfetmek yolu ile öğrenme

Bir sistem bir öğrenme kuralı kullanılabildiği gibi, birden fazla öğrenme kuralı da kullanılabilir.

1.2.3. Genetik Algoritmalar

Genetik algoritmalar (GA), karmaşık, çözümü başka yöntemlerle neredeyse imkansız olan problemlerin çözümünde kullanılır. Evrimsel hesaplama tekniğinin bir parçası olan genetik algoritma, Darwin'in evrim teorisinden esinlenerek oluşturulmuştur.

Genetik algoritmalar 1960'larda John Holland tarafından bulundu ve 1960 ve 1970'lerde Michigan Üniversitesinde Holland öncülüğünde, arkadaşlarıyla birlikte geliştirildi. Holland (1975), yayınladığı kitabı (*Adaptation in Natural and Artificial Systems*) ile Genetik Algoritmalar ile ilgili çalışmalara kılavuzluk etti.

Genetik algoritmalar, evrimsel programlama ve evrim stratejileri ile birlikte evrim algoritmaları adı altında geçmektedir. Genetik algoritmalar ile problemlerin çözümlenmesinde arzu edilen sonucu üretecek özelliklerin kalıtım yolu ile başlangıç çözümlerinden elde edilen yeni çözümlere onlardan da daha sonraki çözümlere geçtiği kabul edilmektedir³⁶.

Genetik algoritmaların temel elemanları şöyledir³⁷:

1.2.3.1. Kromozom ve Gen

Uygulamada ilk adım, bağımsız parametrelerin kromozomlar içerisinde kodlanmasıdır. Bir kromozom elemanlarından her biri çözümün bir özelliğini gösterir. Bunlara da gen denir.

³⁶ Öztemel, a.g.e., s.17

³⁷ A.e., s.17

1.2.3.2. Çaprazlama

Çaprazlama operatörü Genetik algoritmalarındaki en önemli elemandır³⁸. Çaprazlamada, alınan iki kromozomun genetik parçalarının yer değiştirmesi ile yeni bir kromozom üretilir. Tek nokta, iki nokta ve düzgün çaprazlama olmak üzere üç çeşit çaprazlama vardır.

1.2.3.3. Mutasyon

Genetik algortmada, çözümde çeşitlendirme yapmak için mutasyona başvurulur. Mutasyonda genlerden biri rassal olarak değiştirilir. Eğer ikili (0-1) yapıda bir metot tercih edilmişse bu durumda bir genin 0 ise 1, 1 ise 0 yapılması bir mutasyon olacaktır. Genellikle kullanılan mutasyon oranı, birin birey gen uzunluğuna bölümü seviyesindedir³⁹.

1.2.3.4. Uygunluk Fonksiyonu

Kromozomların ne kadar iyi olduğunu bulan fonksiyona uygunluk fonksiyonu denir. Bu fonksiyon işletilerek kromozomların uygunluklarının bulunması ise hesaplama (evaluation) adı verilir. Bu fonksiyon genetik algoritmanın beynini oluşturmaktadır. GA da probleme özel çalışan tek kısım bu fonksiyondur. Çoğu zaman GA' nın başarısı bu fonksiyonun verimli ve hassas olmasına bağlı olmaktadır.

1.2.3.5. Yeniden üretim

Kromozomlar, çaprazlama ve mutasyon neticesinde üretilen yeni kromozomlar nedeni ile çoğalacaktır. Bunlar arasından seçilen kromozomlar sonraki nesil çözümü olarak yeniden çaprazlanıp gelecek çözümleri üretirler.

1.2.4. Bulanık Mantık

Gerçek yaşamda olaylar karmaşık ve belirsizdir. Örneğin, sıcak hava denildiğinde Türkiye'de yaşayanlar ile kutuplarda yaşayanların sıcaklık kavramı farklıdır. İnsanlar havanın derecesini belirtmek yerine, sıcak, soğuk, normal, sıcak ama rüzgarlı gibi

³⁸Öznur İşçi, Serdar Körükoğlu, Genetik Algoritma Yaklaşımı ve Yöneylem Araştırmasında Bir Uygulama, Celal Bayar Üniversitesi İ.İ.B.F, Yönetim ve Ekonomi, 10 (2), 2003, s.195

³⁹ Hakan Er, M. Koray Çetin ve Emre İpekçi Çetin, Finansta Evrimsel Algoritmik Yaklaşımlar: Genetik Algoritma Uygulamaları, Akdeniz İ.İ.B.F. Dergisi (10), (73-94), 2005, s.76.

ifadeler kullanırlar. Bilgisayarlar ise “binary logic” adı verilen temeli Aristo’ya dayanan ikili mantık sistemine göre çalışır. Yani bilgisayarlar için 1 ya da 0 vardır: sıcak-soğuk, var-yok, uzun-kısa. Fakat gerçek hayatta, ılık, normal, orta gibi kavramlarda vardır. Bilgisayarın bu kavramları anlayabilmesi için bulanık mantık geliştirilmiştir.

Bulanık mantık, 1965 yılında Lütfi Zadeh’ in California Berkeley Üniversitesi’nde ‘Bulanık Kümeler’ adlı makalesi ile başlamıştır. 1974 yılında, S. Assilian ve E.H. Mamdani bulanık mantığı ilk kez bir buhar makinesinin kontrolünde kullanmıştır⁴⁰.

Pratikte genel olarak klasik küme şeklinde beliren değişim aralıklarının bulanıklaştırılması, bulanık küme oluşturmaktır. Bunun için bir aralıkta bulunabilecek öğelerin hepsinin bir üyelik derecesine sahip olacak yerde, 0 ile 1 arasında değişik değerlere sahip olması düşünülür⁴¹. Örneğin hava sıcaklığının normal olması durumunda değişkenin adı “normal” olabilir. Üyelik fonksiyonu ise herhangi bir sıcaklık değerinin “normal” olma üyeliğini gösterir. 1 tam üyelik durumunu 0 ise üye olmama durumunu gösterir⁴².

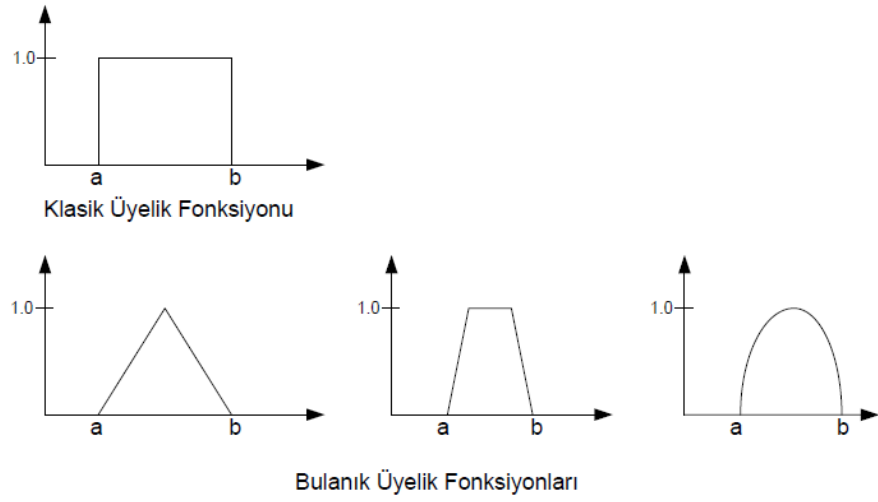
Bulanıklaştırmadan sonra, bulanık önermelerin işlenmesi aşamasına geçilir. Genellikle, üyelik fonksiyonlarının üst üste konulması sonucu kurallara göre ortak alanın bulunmasıdır. Eğer kurallar “ve” bağlacı ile bağlanmış ise, üyelik fonksiyonlarının küçük değeri, “veya” bağlacı ile bağlanmış ise, üyelik fonksiyonlarının büyük değerleri alınarak oluşturulur⁴³.

⁴⁰Padhy,ag.e.,s.

⁴¹M. CengizKayacan,Abdurrahman S. Çelik ve Özlem Salman, Tornalama İşlemlerinde Kesici Takım Aşınmasının Bulanık Mantıkla Modellenmesi, Mühendis ve Makine,526(47-54)2003,s.50.

⁴²Öztemel,a.g.e.,s.19

⁴³A.e.,s.20.



Şekil 1.3:Üyelik Fonksiyonu Örnekleri (Nabiyev, 2003 s.669)

Bulanık önermeler işlendikten sonra, netleştirme işlemi yapılır. Netleştirme, çözüm alanından tek bir değer elde edilmesi işlemidir⁴⁴.

⁴⁴Öztemel,a.g.e.s.20

2. YAPAY SİNİR AĞLARI

YSA, insan beyninin öğrenme yolu ile yeni bilgiler türetebilme, oluşturabilme ve keşfedebilme gibi yeteneklerini herhangi bir yardım almadan otomatik olarak gerçekleştirmek amacı ile geliştirilen bilgisayar sistemleridir⁴⁵.

YSA, beyindeki sinir sisteminin basit bir modelini bilgisayarlarda oluşturarak beynin, öğrenme, ilişkilendirme, sınıflandırma, genelleme gibi özelliklerini uygular. Beynin temel elemanı olan nöronu model olarak alan YSA, bir anlamda nöronun çalışma şeklini basit bir şekilde bilgisayarlarda simüle etmeye çalışır.

Makinelerin hesaplama yapabilmek için biyolojiden esinlenmesi, çok önemli bir gelişme olarak nitelendirilmektedir⁴⁶. İnsan beyni, örneğin, birbirine ağ ile bağlanmış tahminen 10^{10} tane nöron içermektedir ve bunların her biri ortalama 10^4 'ü ile bağlantılıdır. Nöron aktiviteleri, birbiriyle ilişkili nöronların uyarılması veya engellenmesi ile gerçekleşir. Nöronlarda oluşan sinyaller, saniyede 100 metre hızla diğer hücrelere iletilir. Bu nedenle insanlar çok zor ve karmaşık durumlarda, çok hızlı bir şekilde karar verir. Örneğin, görülen kişinin annemiz olduğuna karar vermek yaklaşık 10^{-1} sn. sürer⁴⁷.

Bilgisayarlar, hesaplama, muhasebe gibi ezber gerektiren işleri çok hızlı bir şekilde yerine getirir⁴⁸. Bilgisayarların elektriksel, nöronların ise kimyasal olarak çalışması nedeni ile de hesaplama gibi işlemler insanlara göre oldukça hızlı bir şekilde yerine getirilir⁴⁹.

İnsan beyinde, tek bir nöronun bozulması, beynin bütün faaliyetlerinin durmasına neden olmaz. Fakat bilgisayarda, tek bir komutunun yanlış olması bile o programın

⁴⁵ Öztemel,A.e.,s.29

⁴⁶ Dave Anderson, George McNeill, Artificial Neural Networks Technology, Kaman Sciences Corporation, New York,1992.

⁴⁷ Tom M. Mitchell,Machine Learning,McGraw-Hill Science/Engineering/Math,1997,s.82.

⁴⁸ Anderson ve Mc Neill,a.g.e.,s.15

⁴⁹ Yıldız,a.g.e.,s.46

çalışmamasına ya da yanlış çalışmasına neden olur. Bunun nedeni bilgisayarların seri, beynin ise paralel işlemci mantığı ile çalışmasından kaynaklanır. Yani, bilgisayarlarda tek bir merkezi işlemci her hareketi sırasıyla gerçekleştirmektedir. Beyinde ise her bir sinir hücresi, büyük bir problemin bir parçasıyla ilgilenen birbirine paralel bağlanmış bir işlemci eleman yapısındadır. Sinir hücreleri kendi başlarına yavaş olmalarına karşın sistem paralel çalışmasından dolayı hızlıdır⁵⁰.

Belirtildiği gibi, bilgisayarlar sayısal hesaplamalarda, insanlardan çok daha hızlıdır. Fakat, görüntüleri görüntü ve ses tanıma ve işleme, plan yapma, tecrübe yeteneği ile yorum yapabilme ve daha önce hiç karşılaşmadığı olaylar hakkında hızlı ve doğru karar verebilme gibi özellikleri ile beyin, makinelerden daha üstündür. Beynin ve bilgisayarların birbirlerinden farklı olarak üstün oldukları bu özellikler ve son zamanlarda beynin çalışma sistemi üzerine edinilen bilgilerin artması, insan beynini modelleyerek çalışan YSA araştırmalarını da arttırmıştır.

2.1. Yapay Sinir Ağlarının Tarihsel Gelişimi

1890 yılında insan beyni hakkındaki ilk çalışma William James tarafından yayınlanmıştır. 1940 yılından önce Helmholtz, Pavlov, Poincare gibi bazı bilim adamlarının YSA üzerinde çalıştıkları bilinmektedir. Fakat bu çalışmalarda mühendislik tabanı olmadığından önem verilmemektedir. 1940'lı yıllarda Hebb, Rosenblatt, McCulloch ve Pitts gibi bilim adamları çalışmalarını mühendislik ve matematik alanlarına kaydardıkları için bu günkü YSA'nın temelini oluşturmaya başladılar⁵¹. 1943 yılında bir nöroloji uzmanı olan Warren McCulloch ile matematikçi olan Walter yayımlanan bir makalede bir sinir hücresinin matematiksel modelini oluşturmuşlardır. Bu çalışmada, elektrik devreleri ile basit sinir ağının nasıl çalıştığını modellemişlerdir⁵².

⁵⁰ Nuray Baş, Yapay Sinir Ağları ve Bir Uygulama, Mimar Sinan Üniversitesi Fen Bilimleri Enstitüsü Yüksek Lisans Tezi, 2006.s.6.

⁵¹ Öztemel, a.g.e., s.37

⁵² Anderson ve McNeill, a.g.e., s.17.

1949 yılında Donald Hebb, The Organization of Behavior adlı kitabında bu gün bir çok öğrenme kuralının temelini oluşturan “Hebbian öğrenme” kuralını ortaya koymuştur⁵³. 1951 yılında ilk nuro-bilgisayar üretildi. 1958 yılında Frank Rosenblatt tarafından geliştirilen algılayıcı model (perceptron) YSA çalışmalarında önemli bir gelişmedir. Perceptron tek katmanlı bir yapay sinir ağıdır. Buradan yola çıkarak YSA’da çok önemli olan çok katmanlı algılayıcı modeller geliştirildi. 1959 yılında Bernard Widrow ve Marcian Hoff ADALINE (ADaptive LINear Elements) modelini geliştirdiler. ADALINE 1970’lerin sonlarında ortaya çıkan ve ADALINE modelinin çok katmanlı hali olan MADALINE’nin temelini oluşturdu. MADALINE, telefon hatlarında oluşan yankıların yok edilmesi için kullanılmış olup literatürde gerçek hayatta uygulanan ilk yapay sinir ağı olarak geçer⁵⁴.

YSA’nın 1960’lara kadar olan başarıları fazla abartıldı ve yapabildiklerine dair beklenti yükseldi. YSA’nın bu gelişmesi dönemin ile düşünen makinelerin insanların hayatını ele geçirdiği Asimov’un kitapları ve Kubrick’in Uzay 2001 Macerası filmindeki HAL gibi eserler ortaya çıktı⁵⁵. 1969 yılında ise, YSA’nın önemli araştırmacılarından Marvin Minsky ve Seymour Papert’in yayımladıkları “Perceptrons” adlı kitapta YSA’nın sanıldığı gibi doğrusal olmayan problemlerin çözüm yapamadığını iddia ettiler. Örnek olarak ise XOR probleminin algılayıcılar ile çözülemediğini matematiksel olarak gösterdiler. Bu kitap, YSA’ya olan güvenin azalmasına neden oldu. Bunun sonucunda, konu ile ilgili çalışmalar durdu ve kuruluşlar finansal desteğini kesti. Çalışmalar, XOR problemi çözülünceye kadar durakladı⁵⁶.

YSA’nın o dönemdeki kötü ününe rağmen bazı araştırmacılar çalışmalarına devam ettiler: Shun-ichi Amari, James Anderson, Michael Arbib, Kunihiko Fukushima, Stephen Grossberg, Teuvo Kohonen, Arthur Little, John Hopfield ve Paul Werbos. 1980’li yıllara gelindiğinde teknolojinin gelişmesinin de yardımıyla bu

⁵³ Kishan Mehrotra , Chilukuri K. Mohan ve Sanjay Ranka , Elements of Artificial Neural Networks, The MIT Press,London,1997,.s.22

⁵⁴ Anderson ve McNeill, a.g.e.,s.64.

⁵⁵ A.e.,s.18.

⁵⁶ Öztemel,a.g.e.,s.38

araştırmacıların çalışmaları sonuç vermeye başladı. 1982’de John Hopfield, Hopfield ağı ile YSA’nın genelleştirilebileceği ve geleneksel çözüm yöntemleri ile çözülmesi zor olan problemlere çözüm getirilebileceğini gösterdi. Çalışmasını mühendislerin kolaylıkla anlayabileceği şekilde sunduğundan YSA’ya olan ilgi arttı. Çalışmaların neticesi Hinton ve arkadaşlarının geliştirdikleri Boltzman makinesi oldu. Bu dönemde Rummelhart ve arkadaşlarının yayınladıkları 2 ciltlik kitapta, çok katmanlı algılayıcıların ilk temelleri atıldı. Çok katmanlı algılayıcılar ile XOR problemi çözmekle birlikte, Hopfield ağlarının ve Boltzman makinesinin kısıtlamalarını da çözdü. Böylece YSA’ya olan güven yeniden artmış ve konu ile ilgili çalışmalar hız kazandı. Yine 1982 yılında Kohonen, kendi adıyla anılan kendi kendine öğrenebilen (Self Organizing Feature Maps - SOM) ağlar adlı çalışması yayınlamıştır⁵⁷. 1986’da Rumelhart ve McClelland karmaşık ve çok katmanlı ağlar için geri yayılım öğrenme algoritmasını (backpropagation) ortaya koymuşlardır⁵⁸.

1988’de Broomhead ve Lowe, Radyal Tabanlı Fonksiyonlar (Radial Basis Functions-RBF) modelini geliştirmişlerdir. Bu ağı, çok katmanlı algılayıcılara bir alternatif olarak sundular. Aynı yıl Specht, bu ağları daha çok geliştirerek, Probabilistik Ağlar (PNN) ve 1991’de Genel Regresyon Ağları (GRNN)’nı geliştirdi⁵⁹.

Doksanlı yıllarla birlikte, YSA çalışmaları artmış, sayısız gelişme meydana gelmiştir. Bilgisayar teknolojisinin gelişmesi, bu çalışmaların en büyük destekçisi olmuştur.

2.2. Yapay Sinir Ağlarının Yapısı ve Temel Elemanları:

2.2.1. Biyolojik Sinir Ağları

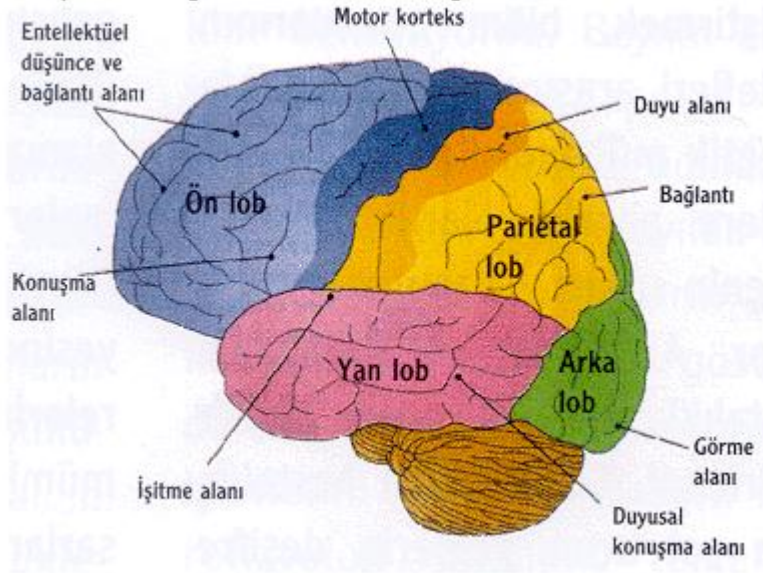
Yaklaşık 2200 cm² ve 1.5 kg. olan insan beyninin 3’de 2 si kıvrımlardan oluşur. Bu kıvrımlar arasında yaklaşık 100 milyar sinir hücresi ve bunlar arasındaki ilişkiler ağı bulunur. Vücudumuzdaki kaslara, organlara ve salgı bezlerine bilgiler göndererek

⁵⁷ A.e.,s.39.

⁵⁸ Mehrotra, Mohan, Ranka,a.g.e.,s.65.

⁵⁹ Öztemel,a.g.e.,s.40

onların çalışmasını kontrol eden sinir hücrelerine "nöron" denilir⁶⁰. Nöron sinir sisteminin temelidir.



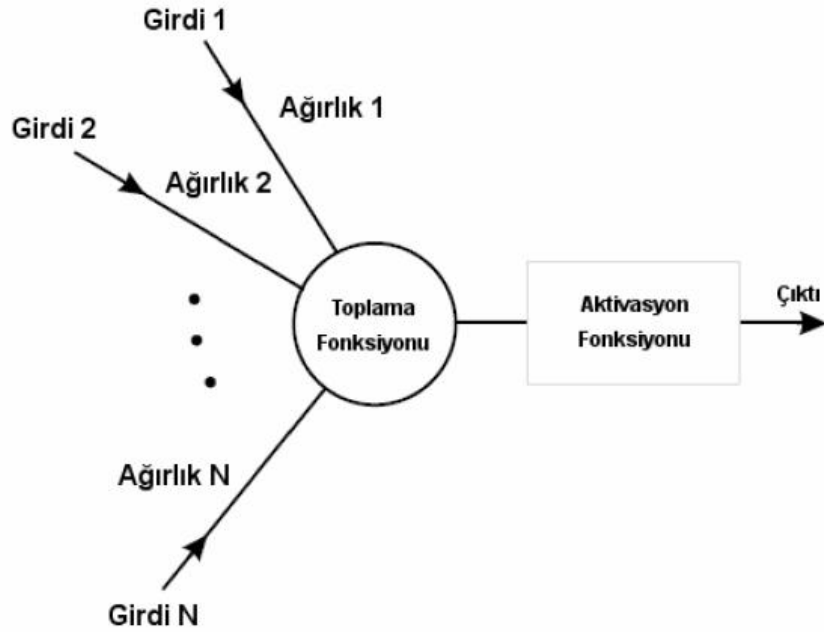
Şekil 2.1: Beyin Yapısı

Sinir hücresi; çekirdek (nucleus), dendritler, aksonlar (axon) ve sinapsler (synapse) olmak üzere 4 temel öğeden meydana gelir. Dendritler, diğer hücrelerden aldığı bilgileri hücrenin çekirdeğine iletir. Aksonlar ise elektriksel darbeler şeklindeki bilgiyi hücreden dışarı taşıyan birimdir. Aksonların bitimi, ince yollara ayrılabilir ve bu yollar, diğer hücreler için dendritleri oluşturur. Bu bağlantıyı sağlayan elemana da “sinaps” denir.

2.2.2. Yapay Sinir Hücresi (Proses Elemanı)

Sinir sisteminin temel elemanı nöronlar olduğu gibi, YSA’nın temel elemanı da yapay nöronlardır. Yapay nöronların 5 temel elemanı vardır: Girdiler, Ağırlıklar, Toplama fonksiyonu, Aktivasyon Fonksiyonu ve Çıktılar.

⁶⁰Bilim Teknik Beyin Özel Sayısı Eylül 2003,(Çevrimiçi) www.bilimteknik.com,2010.



Şekil 2.2: Yapay Sinir Hücresinin Yapısı⁶¹

2.2.2.1. Girdiler

Yapay sinir hücresine dış dünyadan ve kendinden gelen geri besleme bilgilerini ara katmanlara transfer ederler. Girdi katmanı bu bilgiler üzerinde herhangi bir işlem uygulamaz, olduğu gibi iletir.

2.2.2.2. Ağırlıklar

Ağırlıklar bir yapay hücreye gelen bilginin önemini ve etkisini gösterir. Ağırlıkların büyük ya da küçük olması önemli ya da önemsiz olduğu anlamına gelmez. Bir ağırlığın değerinin sıfır olması, o ağ için en önemli olay olabilir. Eksi değerler önemsiz demek değildir. Bu nedenle artı veya eksi olması, pozitif veya negatif olduğunu sıfır olması ise herhangi bir etkisinin olmadığını gösterir. Ağırlıklar değişkenler ya da sabit değerler de olabilirler⁶².

⁶¹ Kevin Gurney, Computers and Symbols versus Nets and Neurons, UCL Draft Papers, U.K, 1996.s.4

⁶² Öztemel, a.g.e., s.49

2.2.2.3. Toplama Fonksiyonu

Toplama fonksiyonu, bir hücreye gelen net girdiyi hesaplar. Bunun için bir çok toplama fonksiyonu içinden en uygun olanı seçilir. En çok kullanılan toplama fonksiyonu, girdi katmanına gelen bilgileri (A_i), her bir girdinin kendi ağırlık değeri (G_i) ile çarpılıp toplanarak net girdi (S) bulunur:

$$S = \sum_i^n A_i G_i \quad (2.1)$$

Net Giriş	Açıklama
Çarpım: $Net\ Girdi = \prod_i G_i A_i$	Ağırlık değerleri girdiler ile çarpılır ve daha sonra bulunan değerler birbirleri ile çarpılarak net girdi hesaplanır.
Maksimum: $Net\ Girdi = Max(G_i A_i), i = 1.....N$	N tane girdi içinden ağırlıklar ile çarpıldıktan sonra en büyüğü yapay sinir hücresinin net girdisi olarak kabul edilir.
Minimum: $Net\ Girdi = Min (G_i A_i), i = 1.....N$	N tane girdi içinden ağırlıklar ile çarpıldıktan sonra en küçüğü yapay sinir hücresinin net girdisi olarak kabul edilir.
Çoğunluk: $Net\ Girdi = \sum_i sgn(G_i A_i)$	N tane girdi içinden ağırlıklar ile çarpıldıktan sonra pozitif ve negatif olanların sayısı bulunur. Büyük olan sayı hücrenin net girdisi olarak kabul edilir.
Kümülatif Toplam: $Net\ Girdi = Net(eski) + \sum (G_i A_i)$	Hücreye gelen bilgiler ağırlıklı olarak toplanır ve daha önce gelen bilgilere eklenerek hücrenin net girdisi bulunur.

Tablo 2.1: Toplama Fonksiyonu Örnekleri⁶³

Toplama fonksiyonu örneklerinden de görüldüğü gibi, kimi durumlarda girdilerin değeri önemliyken kimi durumlarda girdilerin sayısı önemlidir. Hangi durum için hangi toplama fonksiyonunun en uygun olduğunu belirleyen bir formül yoktur. Genellikle deneme yanılma yolu ile en uygun toplama fonksiyonu bulunur. Sinir

⁶³ Öztemel,a.g.e.,s.50.

ağındaki bütün işlemci elemanların aynı toplama fonksiyonuna sahip olması zorunluluğu yoktur. Bazen aynı yapay sinir ağı içindeki işlemci elemanların bir grubu aynı, diğerleri ise başka toplama fonksiyonlarını kullanabilirler. Bu tamamen tasarımcının kendi kararına bağlıdır⁶⁴.

2.2.2.4. Aktivasyon Fonksiyonu

Transfer, sıkıştırma, eşik veya işlemci fonksiyonu olarak da adlandırılan aktivasyon fonksiyonu, hücreye gelen net girdiyi işleyerek hücre çıktısını üretir. Toplama fonksiyonunda olduğu gibi, aktivasyon fonksiyonu için de çeşitli formüller kullanılır ve en uygun olanı yine deneme yanılma yoluyla bulunur. Bir ağıdaki tüm hücrelerin aktivasyon fonksiyonu da birbirinden farklı olabilir. Ayrıca, çok katmanlı algılayıcı gibi bazı modeller, bu fonksiyonun türevi alınabilir olmasını şart koşmaktadır⁶⁵.

Aktivasyon fonksiyonu olarak doğrusal fonksiyonlar girdi çıktı ile orantılı olduğundan genelde tercih edilmez. Bu durum, ilk YSA denemelerinin başarısızlıkla sonuçlanmasının temel nedenidir⁶⁶. En çok kullanılan aktivasyon fonksiyonu, sigmoid fonksiyondur.

Bazı aktivasyon fonksiyonları şöyledir:

a. Lineer Fonksiyon:

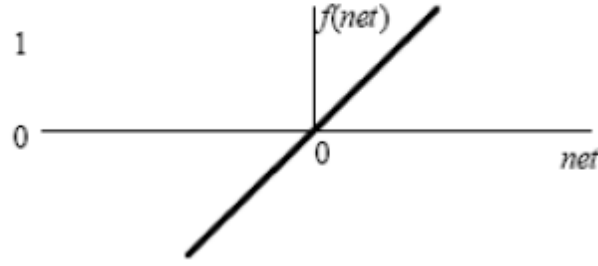
Gelen girdiler hücrenin çıktısı olarak kabul edilir.

$$f(net) = net \quad (2.2)$$

⁶⁴ A.e.,s.50

⁶⁵ Öztemel,a.g.e.,s.51

⁶⁶ M.L.Minsky, and S.S Papert.,Perceptrons, MA:MIT Press, Cambridge,1969.



Şekil 2.3: Lineer Aktivasyon Fonksiyonu

b. Step fonksiyonu:

Gelen net girdi değerinin belirlenen bir eşik değeri altında veya üstünde olmasına göre hücrenin çıktısı 1 veya 0 değerlerini alır.

$$f(net) = \begin{cases} 0 & \text{if } net \leq 0 \\ 1 & \text{if } net > 0 \end{cases} \quad (2.3)$$



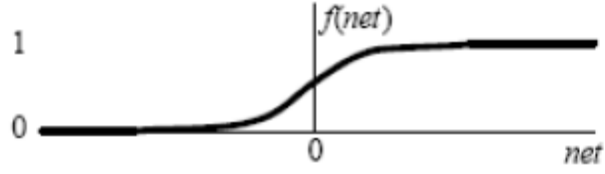
Şekil 2.4: Step Aktivasyon Fonksiyonu

c. Sigmoid Fonksiyonu:

Sigmoid aktivasyon fonksiyonu, sürekli, doğrusal olmayan ve türevi alınabilir olmasından dolayı en çok kullanılan aktivasyon fonksiyonudur. Girdinin her değeri için 0 ile 1 arasında bir değer hesaplar⁶⁷.

$$f(net) = \frac{1}{1 + e^{-net}} \quad (2.4)$$

⁶⁷ SARAÇ, Tuğba, Yapay Sinir Ağları, Seminer Projesi, Gazi Üniversitesi Endüstri Mühendisliği Anabilim Dalı, 2004,s.24.

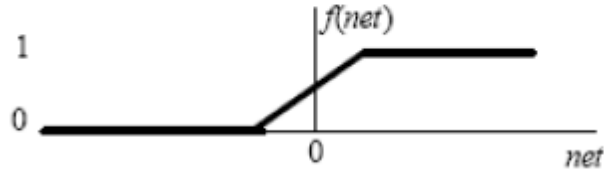


Şekil 2.5: Sigmoid Aktivasyon Fonksiyonu

d. Eşik Değer Fonksiyonu:

Gelen bilgilerin, 0 veya 1'den büyük veya küçük olmasına göre bir değer alır. 0 ve 1 arasında değerler alabilir.

$$F(NE\!T) = \begin{cases} 0 & \text{if } NE\!T \leq 0 \\ NE\!T & \text{if } 0 < NE\!T < 1 \\ 1 & \text{if } NE\!T \geq 1 \end{cases} \quad (2.5)$$

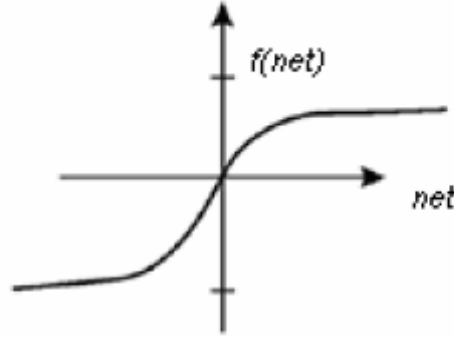


Şekil 2.6: Eşik Değer Aktivasyon Fonksiyonu

e. Hiperbolik Tanjant Fonksiyonu:

Gelen net girdi değerinin tanjant fonksiyonundan geçirilmesi ile hesaplanır.

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



Şekil 2.7: Hiperbolik Tanjant Aktivasyon Fonksiyonu

2.2.2.5. Hücrenin Çıktısı

Yapay sinir hücresinin çıktısı, aktivasyon fonksiyonu tarafından hesaplanan çıktı değeridir. Üretilen çıktı dış dünyaya veya başka bir hücreye gönderilir. Bunun yanında hücre kendi çıktısını kendine girdi olarak da gönderebilir.

2.2.3. Yapay Sinir Ağlarının Yapısı

Yapay sinir hücreleri bir araya gelerek YSA'yı oluştururlar. Bu sinir ağı yapısı oluşurken hücreler birbirine paralel 3 katman halinde dizilir: Girdi Katmanı, Ara Katman (Gizli Katman) ve Çıkış Katmanı.

2.2.3.1. Girdi Katmanı

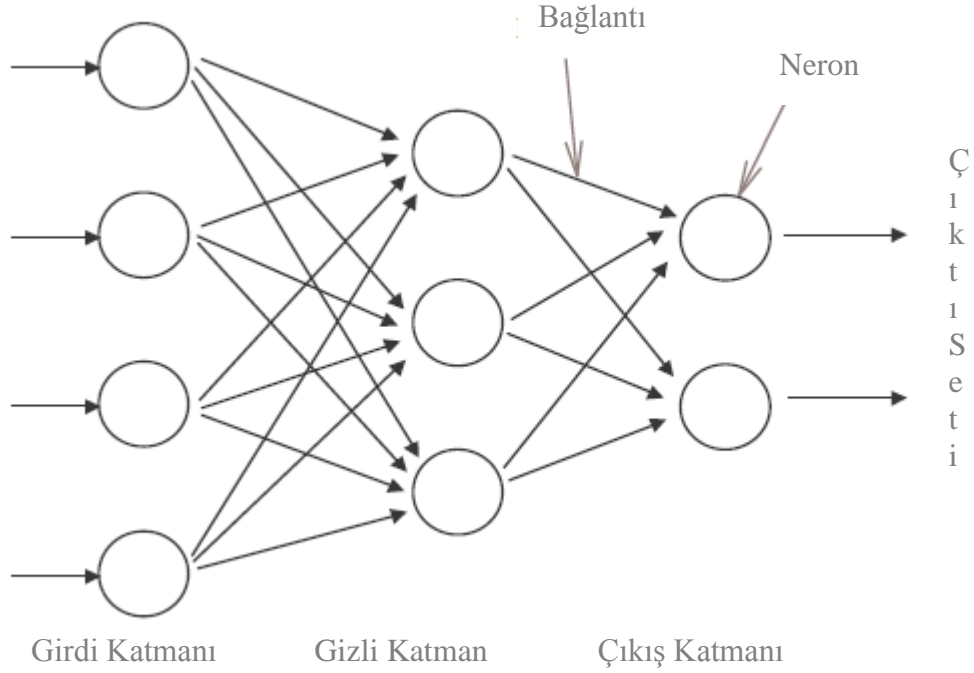
Bu katmanda bulunan proses elemanları, dış ortamdan bilgileri alarak ara katmanlara iletirler. Bazı YSA'da bu katmanda herhangi bir bilgi işleme olmamaktadır.

2.2.3.2. Ara (Gizli) Katmanlar

Bilgilerin işlenerek çıktı katmanına gönderildiği katmandır. Bir yapay sinir ağında birden fazla ara katman olabilir.

2.2.3.3. Çıktı Katmanı

Bu katmanda bulunan proses elemanları, ara katmandan gelen bilgileri işleyerek, ağın çıktısını üretir.



Şekil 2.8: Bir Yapay Sinir Ağı Örneği

2.3. Yapay Sinir Ağlarının Çalışması

YSA'nın çalışma prensibi, ağa gösterilen girdilere göre, ağın onu çeşitli işlemlerden geçirerek çıktı seti oluşturmasıdır. Bunun için ağa girdiler göstererek ağın doğru çıktı oluşturması (ağın eğitilmesi) gerçekleştirilmelidir. Ağa gösterilen girdiler, vektör halinde ağa sunulmalıdır. Ağ buna göre, çıktı vektörü oluşturur.

YSA'nın, girdi vektörlerini, çıktı vektörlerine nasıl dönüştürdüğü bilinmemektedir. Bu nedenle YSA'nın çalışma şekli "kara kutu" olarak nitelendirilmektedir. Bu da ağa olan güveni azaltmaktadır.

2.3.1. Yapay Sinir Ağlarının Eğitilmesi ve Testi

YSA'da **ağın eğitilmesi**, proses elemanları arasındaki ağırlık değerlerinin değiştirilmesi işlemidir. Başlangıçta rasgele atanan bu ağırlıklar, ağa örnekler gösterildikçe değişir. Amaç, ağa yeterli sayıda örnek göstererek, ağın doğru çıktıyı üretecek ağırlıkları bulmasıdır. Ağın doğru ağırlıkları bulması, ağın sonuçlar

hakkında genelleme yapabildiğini gösterir. Genelleme, ağın eğitiminde kullanılmamış fakat aynı olayın bir parçası olan örnekler hakkında doğru tahmin yapabilme özelliğidir. Ağın bu genelleştirme özelliği kazanması olayına “ağın öğrenmesi” adı verilir. Ağın ağırlıklarının değişmesi işlemi belli kurallara göre gerçekleşmektedir. Bu kurallara öğrenme kuralları adı verilir. Ayrıca, öğrenme kurallarına göre kullanılan öğrenme stratejileri vardır.

Ağın eğitimi için kullanılan veriye “eğitim seti” adı verilir. Eğitim sırasında, eğitim seti ağa bir çok kez gösterilerek, en doğru ağırlıklar bulunmaya çalışılır. Eğitim setinin ağa bir kez gösterilmesi iterasyon (döngü, epoch) olarak ifade edilir. İterasyon sayısının uygun seçilmesi ağın performansında çok etkilidir. Gereğinden az veya çok seçilmesi ağın performansını düşürür. Eğitimin süresi genellikle iterasyonla ölçülür. YSA için üretilmiş programlarda, eğitim seti yanında Doğrulama Seti (Validation Set) oluşturulur. Bu veri seti, ağın her iterasyonda ne kadar öğrendiğini test eder.

YSA’da öğrenme iki adımdan oluşur: Birinci adımda ağa gösterilen girdi için ağın üreteceği çıktı belirlenir. Bu çıktı değerine göre, ikinci adımda ağın bağlantılarının ağırlıkları değiştirilir. Ağın çıktısının belirlenmesi ve ağırlıkların değiştirilmesi öğrenme kuralına göre farklılık göstermektedir.

Eğitim işleminin tamamlanmasından sonra, ağın performansının ölçümü için ağ test edilir. Ağı, test etmek için ağa eğitim sırasında gösterilmeyen, veri setinden test amaçlı olarak ayrılan örnekler kullanılır. Bu veri setine “test seti” adı verilir. Test işleminde ağın ağırlık değerleri değiştirilmez. Örnekler ağa gösterilmekte ve ağ eğitimi sırasında belirlenen ağırlık değerlerini kullanarak daha önce görmediği bu örnekler için çıktılar üretir. Elde edilen çıktıların doğruluk dereceleri ağın performansını belirler. YSA’nın, bilinen örnekleri öğrenerek, bilinmeyen örnekler hakkında genelleme yapabilme özelliğine “Adaptif Öğrenme” denir⁶⁸.

⁶⁸ Öztemel,a.g.e.,s.55-56.

2.3.2. Öğrenme Stratejileri

Öğrenme yeteneği, insan zekasının temel özelliklerindendir. YSA’da öğrenme, Simon (1983)’a göre, “zaman içinde yeni bilgilerin keşfedilmesi yoluyla davranışların iyileştirilmesi sürecidir”⁶⁹. Burada bahsedilen iyileşme, YSA’daki genelleme yeteneğidir. Bilgisayarlar da insanlar gibi tecrübe kazanarak öğrenirler.

YSA’da öğrenme, örneklerin ağı gösterilmesi ile sağlanmaktadır. Öğrenme, ağ, genelleme yeteneği kazanana yani gerçek değerler ile çıktı değerleri arasındaki fark olan hata, kabul edilir düzeye gelene kadar devam eder. YSA, örneklerden öğrenirken değişik öğrenme stratejileri kullanmaktadır. Genel olarak 3 çeşit öğrenme stratejisi kullanılır.

2.3.2.1. Öğretmenli (Supervised) Öğrenme

Bu stratejide, her örnek için hem girdiler hem de o girdiler sonucunda, olması gerek çıktılar ağı gösterilir. Ağın görevi, olayın girdileri ile çıktıları arasındaki ilişkiyi öğrenmektir. Olması gereken değerler ile çıktı değerleri arasındaki farkın yani hatanın karelerinin ortalaması en küçük olacak şekilde ağırlıklar sürekli olarak güncellenir ve işlemlere devam edilir. Delta kuralı, geri yayılım (back propagation) öğretmenli eğitimin kurallarından bazılarıdır.

2.3.2.2. Destekleyici (Reinforcement) Öğrenme

Bu öğrenme kuralında da öğretmen vardır. Fakat burada ağı gerçek sonucu sunmak yerine, ağın ürettiği çıktıya göre sonucun doğruluğunu ya da yanlışlığını belirten bir sinyal üretir. Ağ, öğretmenden gelen bu sinyal doğrultusunda öğrenir. LVQ ağı destekleyici öğrenmeye örnek olarak söylenebilir.

⁶⁹ A.e.,s.21

2.3.2.3. Öğretmensiz (Unsupervised) Öğrenme

Bu strateji de öğrenmeye yardımcı bir öğretmen yoktur. Ağa sadece giriş değerleri gösterilir. Gösterilen örnek değerlere göre ağ, parametreler arasındaki ilişkileri kendi kendine öğrenir. Yapay sinir ağı daha sonra bağlantı ağırlıklarını aynı özellikleri gösteren örüntüler (patterns) oluşturmak üzere ayarlar. Daha sonra kullanıcının elde edilen sınıfların ne anlama geldiğini kendisinin yapması gerekir. Danışmansız öğrenme genellikle sınıflandırma problemlerinin çözümünde kullanılmaktadır. ART (Adaptive Resonance Theory), Signal Hebbian Öğrenme Kuralı ve Diferansiyel Hebbian Öğrenme Kuralı danışmansız öğrenmeye örnek olarak verilebilir⁷⁰.

2.3.2.4. Karma Stratejiler

Üç stratejiden, bir kaçını birlikte kullanan ağlar da vardır. Kısmen öğretmenli ve kısmen öğretmensiz olarak öğrenme yapan ağlardır. Radyal Tabanlı Yapay Sinir Ağları (Radial Basis Networks - RBN) ve Olasılık Tabanlı Yapay Sinir Ağları (Probability Based Neural Networks - PBNN) bunlara örnek olarak verilebilir.

2.3.3. Öğrenme Kuralları

YSA'da öğrenme, öğrenme stratejilerinin yanında öğrenme kuralları da vardır.

2.3.3.1. Çevrimiçi (On-Line) Öğrenme Kuralları

Bu kurala göre öğrenen sistemler, gerçek zamanda çalışırken bir taraftan fonksiyonlarını yerine getirmekte, bir taraftan da öğrenmeye devam etmektedirler. **ART** ve **Kohonen's SOM** ağlarında kullanılan **Kohonen Öğrenme Kuralı** bu öğrenme kuralına örnektir⁷¹.

⁷⁰ Saraç,a.g.e.,s.30.

⁷¹ Öztemel,a.g.e.,s.26

2.3.3.2. Çevrimdışı (Offline) Öğrenme

Bu öğrenme kuralında ağ, kullanıma alınmadan önce örnekler üzerinde eğitilir. Bu kuralı kullanan sistemler eğitildikten sonra gerçek hayatta kullanıma alındığında artık öğrenme olmamaktadır. **Delta Öğrenme Kuralı** bu tür öğrenmeye örnektir.

2.3.3.3. Öğrenme Kurallarından Bazıları

YSA modellerinin eğitilmesinde kullanılan birçok öğrenme kuralı vardır. Bu kuralların birçoğunun temeli, Hebb Kuralı ve onun geliştirilmiş versiyonlarıdır. Bununla beraber temelde dört öğrenme kuralından bahsedilebilir: Hebb Kuralı, Hopfield Kuralı, Delta Kuralı ve Kohonen Kuralı.

2.3.3.3.1 Hebb Kuralı

1949 yılında geliştirilen bu kural, bilinen en eski öğrenme kuralıdır. Bu kurala göre, bir sinir hücresi diğer bir sinir hücresinden girdi alırsa ve her iki hücre de yüksek derecede aktif ise (matematiksel olarak aynı işareti taşıyorsa) her iki hücrenin arasındaki bağlantının ağırlığı artırılmalıdır. Yani bir hücre aktifse, bağlı bulunduğu hücreyi aktif, pasifse pasif yapmaya çalışmaktadır⁷².

2.3.3.3.2 Hopfield Kuralı

Bu kural Hebb kuralına benzemektedir. Diğerinden farklı olarak ağ elemanlarının bağlantılarının ne kadar kuvvetlendirilmesi ya da zayıflatılması gerektiğini de belirler. Eğer sonuçların ve girdilerin her ikisi de aktif/pasif ise öğrenme katsayısı kadar ağırlık değerlerini de kuvvetlendirir/zayıflatır. Ağırlıkların kuvvetlendirilmesi ya da zayıflatılması öğrenme katsayısı yardımıyla gerçekleştirilir. Bu katsayı genellikle 0 ile 1 arasında kullanıcı tarafından belirlenen sabit bir pozitif değerdir.

⁷² Öztemel,a.g.e.,s.26.

2.3.3.3.3 Delta Kuralı

Widrow ve Hoff tarafından geliştirilen bu kural, Hebb Kuralının gelişmiş şeklidir. Ağıın gerçek çıktısı ile beklenen çıktısı arasındaki farkı azaltmak için YSA'nın bağlantı ağırlık değerlerinin sürekli değiştirilmesi ilkesine dayanır. Amaç, sonuç ile elde edilen çıktı arasındaki hata karelerinin ortalamasını en aza indirebilmektir. Bu nedenle En Küçük Kareler Kuralı (Least Mean Square Rule-LMS) olarak da adlandırılır. Hatalar en son katmandan geriye doğru ardışık iki katman arasındaki bağlantı ağırlıklarına dağıtılır. Bu işleme hatanın geriye dağıtılması anlamında geri besleme işlemi denir. Geriye yayılım algoritması bu kurala göre çalışmaktadır.

2.3.3.3.4 Kohonen Kuralı

Kohonen (1982) tarafından geliştirilen bu kural biyolojik sistemlerdeki öğrenme sisteminden esinlenilmiştir. Bu sinir hücreleri ağırlıkları değiştirmek için birbirleri ile yarışır. En büyük çıktıyı üreten hücre kazanan hücre olur ve bağlantı ağırlıklarını değiştirir. Kazanan hücre yakınındaki hücelere göre daha kuvvetli hale gelmektedir. Bu kuralda bir hedef değerler dizisi olmasına gerek yoktur. Bu nedenle kendi kendine yani öğretmensiz bir öğrenme kuralıdır.

2.4 Yapay Sinir Ağlarının Avantajları

YSA, özellikle doğrusal olmayan problemleri, geleneksel yöntemlere göre çok daha iyi çözümledikleri için üstünlük sahibidirler. Temelde, YSA'nın karakteristik özellikleri ağıın yapısına göre farklılaşmaktadır. Bu kısımda bahsedilen özellikler, YSA'nın genel özellikleridir.

a. Doğrusal Olmayan Yapı

YSA'nın temel işlemi olan hücrenin yapısı doğrusal değildir. Bu nedenle bu doğrusal olmayan hücrelerden meydana gelen YSA'da doğrusal olmayan bir yapı gösterir. Gerçek hayattaki problemler çoğunlukla doğrusal olmayan yapı gösterdiği için YSA bu soruların çözümünde önemli bir yöntemdir.

b. Paralellik

Günümüzde kullanılan bilgi işlem yöntemlerinin çoğu seri, sıralı işlemlerden oluşmaktadır. YSA ise paralel işlem yöntemini kullanır. Seri işlemcilerde herhangi bir birimin yavaş olması tüm sistemi yavaşlatırken, YSA'ların sahip olduğu paralel işlemciler sayesinde yavaş bir birim sistemin çalışması sırasında herhangi bir soruna yol açmamaktadır. Bu durum YSA'ların daha hızlı ve güvenilir olmasını sağlamaktadır.

c. Genelleme

Ağ, problemi öğrendikten sonra eğitim sırasında ona öğretilmeyen örnekler için de genelleme yapabilme özelliğine sahiptir. Hatalı, eksik veriler için de doğruya yakın sonuç üretebilir. YSA'lar, ilk defa karşılaştıkları veriler, veya eksik veriler hakkında karar verip genelleme yapabildikleri için iyi birer Desen (Örüntü) Tanımlayıcısı (Pattern Recognition Engine) ve Sağlam Sınıflandırıcılardır (Robust Classifier)⁷³.

d. Uyarlanabilirlik

YSA, problemdeki verilere göre ağırlıklarını değiştirir. Yani, belirli bir problemi çözmek amacıyla eğitilen YSA, problemdeki değişimlere ağırlıklarını değiştirip, öğrenmesini sürdürür. Değişim sürekli ise gerçek zamanda da eğitimine devam edebilir. Bu özellik sayesinde YSA, uyarlamalı (adaptive) örnek tanıma, sinyal işleme, sistem tanılama ve denetim gibi alanlarda etkin olarak kullanılabilir⁷⁴.

e. Hata Toleransı

YSA, işlemci elemanları arasında, diğer sistemlerden farklı olarak, paralel bağlantı vardır. Paralel bağlantının bir sonucu olarak, ağın sahip olduğu bilgi tüm bağlantılara dağılmış halde bulunmaktadır. Böylece, giriş verisinde bulunabilecek herhangi bir gürültü, bütün ağırlıklara dağıtılmış halde bulunduğundan çok büyük

⁷³Hasan Yurtoğlu, "Yapay Sinir Ağları Metodolojisi ile Öngörü Modellemesi: Bazi Makroekonomik Değişkenler İçin Türkiye Örneği,Uzmanlık Tezi, DPT Uzmanlık Tezleri Yayın No: 2683, Ekonomik Modeller ve Stratejik Araştırmalar Genel Müdürlüğü, 2005,s.36.

⁷⁴ Saraç,a.g.e.,s.14.

hatalara neden olmaz, hata tolere edilebilir. Bu hatayı tolere edebilme özellikleri sayesinde, en karmaşık problemlere bile uygulanabilmekte ve geleneksel yöntemlere göre çok daha iyi sonuçlar vermektedirler.

f. Öğrenebilme

YSA'ların en önemli özelliği insan beyninin çalışma sistemi olan öğrenme yöntemini kullanmasıdır. YSA'da öğrenme, özellikleri verilen örnekler yoluyla sağlanmaktadır ve ağ, örnekleri kullanarak probleme ilişkin genelleme yapabilecek yeteneğe ulaşmaktadır. Bu özelliği sayesinde geleneksel yöntemler için karmaşık olan sorunlara çözüm getirebilmektedirler. İnsanların kolayca yapabildiği ancak geleneksel yöntemler için imkansız olan basit işlemleri de başarıyla yapmaktadırlar. Ayrıca, YSA'lar sürekli öğrenmektedirler. Kendisine gösterilen yeni örnekleri öğrenebilmeleri ve yeni örneklerle adapte olabilmeleri sayesinde sürekli olarak yeni olayları öğrenebilmesi mümkündür.

g. Bilginin saklanması

Geleneksel hesaplama yöntemlerinde bilgi, veri tabanlarında veya program kodlarının içinde saklanmaktadır. YSA' da ise bilgi ağın bağlantılarında, ağa dağılmış durumda bulunmaktadır. Bütün ağ, öğrendiği olayın tamamını göstermektedir. Bu nedenle YSA'nın dağıtılmış bellekte bilgi saklayabildikleri söylenebilmektedir.

h. Yerel Bilgi İşleme ve Esneklik

YSA'da her bir işlem birimi, geleneksel sistemlerden gibi çözülecek problemin tümü ile ilgilenmek yerine, sadece problemin gerekli parçası ile ilgilenmektedir. YSA'lar, her biri ilgili problemin bir parçası ile ilgilenen çok sayıda işlemciden oluşması ve bağlantılar arasındaki ağırlıkların ayarlanabilmesi gibi özelliklerinden dolayı esnek bir yapıya sahiptir. Bu esnek yapı sayesinde ağın bir kısmının zarar görmesi nedeni ile model işlevini yitirmemekte, sadece modelin performansı düşmektedir. Ayrıca, toplam işlem yükünü paylaşan işlemci elemanların birbirleri arasındaki yoğun

bağlantı yapısı sinirsel hesaplamanın temel güç kaynağıdır. Bu işlem yapısı sayesinde, YSA yöntemi en karmaşık problemlere bile uygulanabilmekte ve tatminkar sonuçlar sağlayabilmektedir⁷⁵.

i. Eksik Verilerle Çalışma

YSA, geleneksel sistemlerin aksine, eğitildikten sonra eksik bilgiler ile çalışabilmekte ve gelen yeni örneklerde eksik bilgi olmasına rağmen sonuç üretebilmektedirler. Eksik bilgi nedeniyle, modelin performansı ise eksik bilginin önemine bağlıdır.

j. Donanım ve Hız

YSA paralel yapısı nedeniyle Büyük Ölçekli Entegre Devre Teknolojisi (VLSI) ile gerçekleştirilebilir. Bu özellik, YSA'ların hızlı bilgi işleme yeteneğini arttırmaktadır ve gerçek zamanlı uygulamalarda yaygın olarak kullanılmasının ana sebeplerindendir.

k. Sınırsız Sayıda Değişken ve Parametre Kullanımı

YSA geleneksel yöntemlerin aksine sınırsız sayıda değişken ve parametre ile çalışabilmektedir. Bu sayede mükemmel bir tahmin doğruluğu ile genel çözümler sağlanabilmektedir⁷⁶.

l. Gerçek Zamanlı İşlem

YSA yapısı paralel olarak çalıştığından gerçek zamanlı işlem yapabilir.

⁷⁵ Yurtoğlu, a.g.e., s.35

⁷⁶ A.e., s.36

2.5 Yapay Sinir Ağlarının Dezavantajları

YSA'ların çok sayıda avantajı olduğu gibi, bazı eksiklikleri de vardır. Bunlardan en önemlileri şöyle özetlenebilir⁷⁷:

- YSA'lar donanıma bağımlıdır. Paralel işlem yapmalarından dolayı hızlı çalışan paralel işlemcilerle ihtiyaç duyarlar. Bunun yanında, bir ağın nasıl oluşturulması gerektiğini belirleyecek kurallar yoktur.
- Probleme uygun ağ yapısının belirlenmesi genellikle deneme yanılma yöntemi ile yapılmaktadır. Bu nedenle, YSA ile kesin doğru bir sonuç çıkarılamaz, doğruluğu kabul edilebilir sonuçlar elde edilir.
- Ağın yapısının oluşturulmasında, öğrenme katsayısı, öğrenme-test kümesi, katman sayısı gibi değerlerin belirlenmesinde herhangi bir kural olmaması bir diğer dezavantajdır. Her problem kendi içinde farklılık gösterdiğinden genelleme yapmak da çok güçtür. Ağın performansı, kullanıcının deneyimine bağlıdır.
- YSA sadece nümerik verilerle çalışabilmektedir. Bu nedenle problemlerin nümerik değerlere dönüştürülüp ölçeklenmesi oldukça büyük bir problemidir. Bazı durumlarda kullanıcının deneyimi bile yeterli olmayabilir. Günümüzde çözilemeyen YSA problemlerinin ana nedenlerinden biri nümerikleştirme problemidir.
- Ağın ne kadar eğitileceğinin bilinmemesi de bir diğer problemidir. Genellikle hatanın kabul edilebilir değerler içerisinde kalmasıyla eğitim tamamlanmış olarak kabul edilmektedir. Fakat, ağın optimum öğrenmesini gerçekleştirdiği söylenememektedir.
- YSA'ların en önemli de problemi, davranışlarının açıklanamamasıdır. Problemlere çözüm üretmekte, ama bunu nasıl ve neden yaptığını açıklayamamaktadır. Bu ise ağın sonucuna olan güvenilirliği azaltmaktadır.

⁷⁷ Öztemel,a.g.e.,s.34.

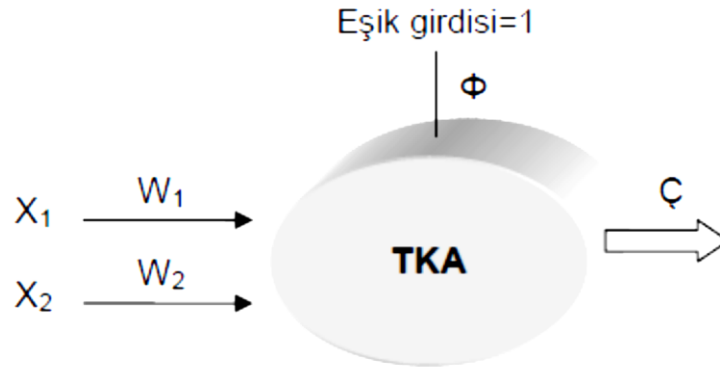
3. YAPAY SİNİR AĞI MODELLERİ

3.1. İlk Yapay Sinir Ağları

İlk yapay sinir ağı modelleri, günümüzde kullanılan gelişmiş modellerin temelini oluşturdıkları için önemlidirler. Bu modeller, tek katmanlı algılayıcılar, perseptron ve ADALINE/MADALINE modelleridir.

3.1.1. Tek Katmanlı Algılayıcılar (TKA)

Tek katmanlı YSA, girdi ve çıktı katmanını olmak üzere iki katmandan oluşur. Ağın, birden fazla girdisi olabilir, fakat tek bir çıktısı (Ç) vardır. Çıktı üniteleri bütün girdi ünitelerine (X) bağlanmaktadır ve her bağlantının bir ağırlığı (W) vardır. Her girdi bir ağırlık değeriyle çıktıya bağlıdır. Bu ağlarda proses elemanlarının değerlerinin ve dolayısıyla ağın çıktısının sıfır olmasını önleyen bir de eşik değeri (Φ) vardır ve değeri daima 1'dir.



Şekil 3.1: İki Girdi ve Bir Çıktı'dan oluşan TKA modeli

Tek katmanlı algılayıcılarda ağın çıktısı, ağırlıklandırılmış girdi değerleri eşik değeri ile toplanıp aktivasyon fonksiyonundan geçirilerek ağın çıktısı hesaplanır. Bu işlem şu şekilde ifade edilir:

$$y = F\left(\sum_{i=1}^m w_i x_i + \theta\right) \quad (3.1)$$

Tek katmanlı algılayıcılarda, çıktı doğrusal bir fonksiyondur. Ağ, eşik değer fonksiyonu yardımıyla gösterilen örnekleri iki sınıfa ayıracak bir doğru üretmeye çalışır. Ağın çıktısı 1 ve -1 değerlerini alarak iki sınıf oluşturur. Çıktı eğer 1 ise birinci sınıf, -1 ise ikinci sınıf olarak kabul edilir.

$$F(s) = \begin{cases} 1 & \text{Eger } s > 0 \\ -1 & \text{Aksi takdirde} \end{cases} \quad (3.2)$$

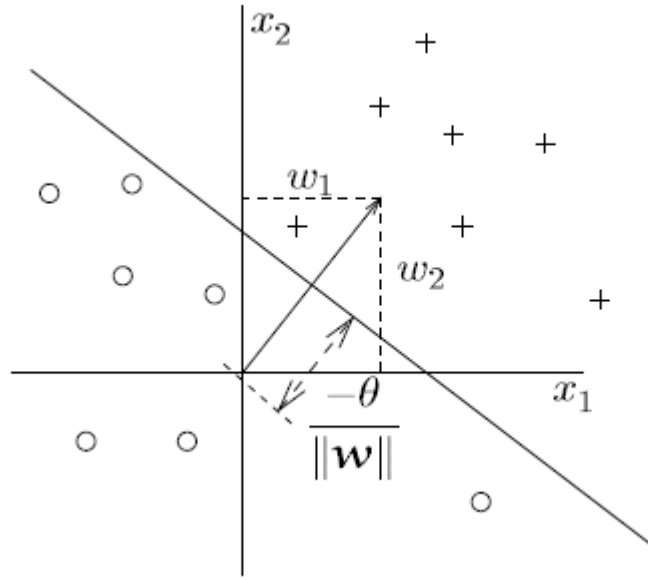
İki sınıfı birbirinden ayıran doğrunun denklemi;

$$w_1 x_1 + w_2 x_2 + \theta = 0 \quad (3.3)$$

formülü ile hesaplanır. Bu denklemden x_1 ve x_2 değerleri bulunarak doğru çizilir.

$$x_1 = -(w_2 / w_1)x_2 - \theta / w_1 \quad (3.4)$$

$$x_2 = -(w_1 / w_2)x_1 - \theta / w_2 \quad (3.5)$$



Şekil 3.2: Ağırlık ve Sınıfları Birbirinden Ayıran Doğrunun Geometrik Gösterimi⁷⁸

Tek katmanlı algılayıcılarda öğrenme, verileri ikiye ayıran en iyi doğrunun bulunmasıdır. Bunun için ağırlık değerlerinin değiştirilmesi gerekmektedir. Bunun için t zaman biriminde ağırlık değerleri Δw kadar değiştirilirse:

$$w_i(t+1) = w_i(t) + \Delta w_i(t) \quad (3.6)$$

ile hesaplanacaktır. Öğrenme esnasında bu değişim her iterasyonda uygulanarak en uygun doğru bulunmaya çalışılır. Ancak bu işlem yeterli olmayabileceğinden eşik değeri de değiştirilir. Buradaki amaç, doğrunun sınıflar arasında kaymasına yardımcı olmaktır. Böylece aktivasyon fonksiyonunun konumu belirlenmiş olur. Öğrenme sırasında eşik değeri ağırlıklarda olduğu gibi her iterasyonda $\Delta \theta$ kadar değiştirilir. t anındaki eşik değeri:

$$\theta(t+1) = \theta(t) + \Delta \theta(t) \quad (3.7)$$

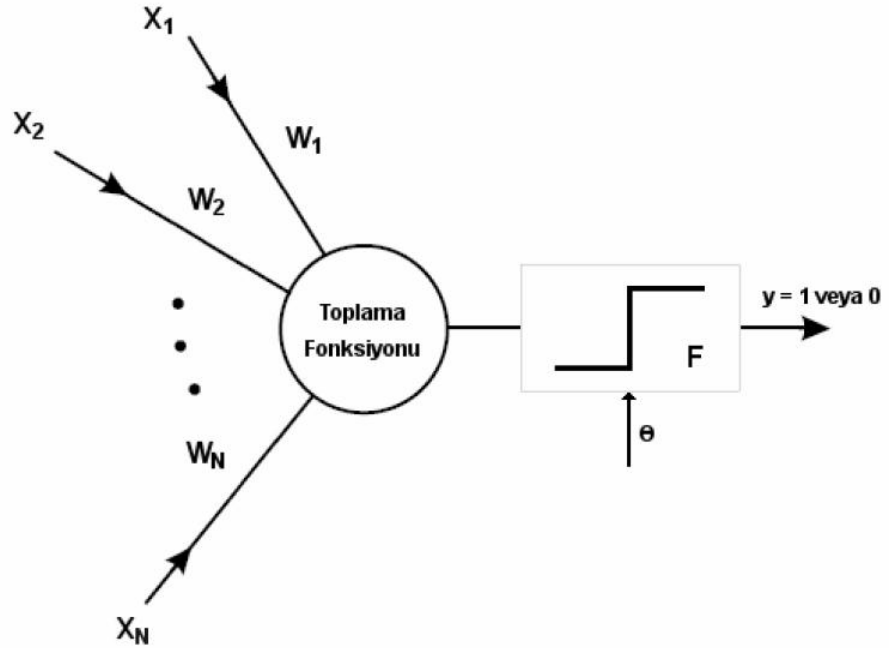
ile hesaplanır.

⁷⁸ Ben KRÖSE ve Patrick Van Der SMAGT, An Introduction to Neural Networks, Eight Ed., The University of Amsterdam, 1996.s.24.

Tek katmanlı YSA'da önemli iki modelden bahsedilebilir. Bu modeller basit algılayıcılar (perseptron) ve Adaptif Doğrusal Eleman (Adaline/Madaline)'dir⁷⁹.

3.1.1.1. Basit Algılayıcılar (Perseptron)

Tek katmanlı algılayıcı (Single Layer Perceptron), F. Rosenblatt tarafından 1950'li yılların sonlarında örüntü (şekil) sınıflandırma amacıyla geliştirilmiştir. Perseptronlar oldukça sınırlı olmalarına rağmen, kendinden sonra geliştirilen modellere temel olmaları açısından önemlidirler. Bir araya gelmiş sinir hücrelerinin miktarına bağlı olarak, basit algılayıcılar ile değişik sayıda sınıflandırma problemi çözebilir. Doğru bir sınıflandırma için sınıfların düzlemsel olarak ayrılması gerektiği gösterilmiştir⁸⁰. Basit algılayıcılar bir sinir hücresinin birden fazla girdiyi alarak bir çıktı üretmesi kuralı ile çalışmaktadır. Ağırlık çıktısı bir veya sıfırdan oluşan mantıksal değerlerdir. Çıktı değeri ise adım fonksiyonu ile hesaplanır.



Şekil 3.3: Bir basit algılayıcı yapısı⁸¹

⁷⁹ Öztemel, a.g.e., s.61

⁸⁰ Minsky and Papert, a.g.e.

⁸¹ Gurney, a.g.e.

Bir basit algılayıcı, eğitilebilen, yani ağırlıkları (W) değiştirilebilen tek bir hücreden oluşur. Girdiler ve her girdiye ait çıktılar ağa gösterilirler. Öğrenme kuralına göre basit algılayıcı, çıktı değerini hesaplar. Ağ, öğrenme kuralına göre, eşik değer ve ağırlıkları değiştirerek en uygun çıktı değerini bulmaya çalışır⁸².

Basit Algılayıcıların öğrenmeleri dört adımda gerçekleşir⁸³.

1. Ağa girdi setini ve girdiler karşılık gelen çıktı değerleri gösterilir. Girdi değeri birden fazla olabilir. Fakat çıktı değeri 0 veya 1 değerini alır.
2. İşlemci elemana gelen net girdi:

$$s = \sum_{i=1}^m w_i x_i \quad (3.8)$$

formülü ile hesaplanır.

3. İşlemcinin çıktısı hesaplanır. Net girdi eşik değerinden büyükse 1, küçükse 0 değerini alır.

$$y = \begin{cases} 1 & \text{Eğer } s > \theta \\ 0 & \text{Eğer } s \leq \theta \end{cases} \quad (3.9)$$

Eğer gerçekleşen çıktı ile beklenen çıktı ise ağ öğrenmiştir. Fakat çıktılar birbirine eşit değilse iki durum söz konusudur.

- a. Ağın beklenen çıktı değeri 0, fakat net girdi eşik değerinin üstünde olan 1 değeridir. Bu durumda ağırlık değeri azaltılır. Bu durum, λ öğrenme katsayısı olmak üzere vektörel olarak şöyle ifade edilir:

$$W_n = W_0 - \lambda X \quad (3.10)$$

⁸²Öztemel, a.g.e, s.62

⁸³ A.e.

Sabit bir deęer olan λ , aęırlıkların deęiřim miktarlarını belirlemektedir.

- b. Beklenen ıktının 1, aęın ıktısının 0 olduęu durumdur. Bu durumda aęırlık deęeri arttırılır. Bu durumun geometrik gsterimi řoyledir:

$$W_n = W_0 + \lambda X \quad (3.11)$$

- 4. İlk  adım, btn girdi setindeki rnekler iin doęru sınıflandırmalar yapılıncaya kadar tekrarlanır.

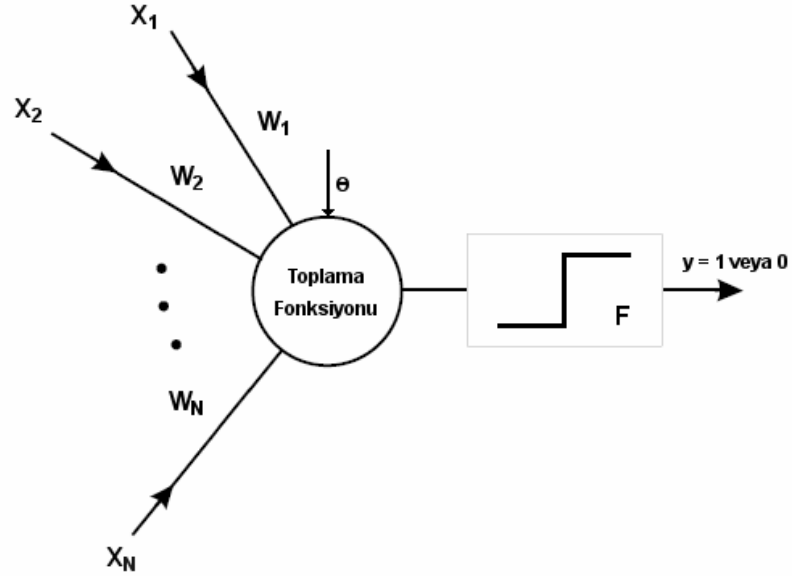
3.1.1.2. ADALINE/MADALINE Modeli

Bu modeller, 1959 yılında Bernard Widrow ve Marcian Hoff tarafından geliřtirilmiřtir. ADALINE uyabilen doęrusal elemanlar (Adaptive Linear Elements), MADALINE ise oklu uyabilen doęrusal elemanlar (Multiple Adaptive Linear Elements) olarak tanımlanır⁸⁴.

ADALINE aęı, en kk ortalamaların karesi yntemine gre alıřmaktadır. ęrenme kuralına delta kuralı da denir. ęrenme kuralı, aęın ıktısının beklenen ıktı deęerine gre hatasını en aza indirecek řekilde aęın aęırlıklarının deęiřtirilmesi kuralına gre alıřır. ADALINE modeli tek katmanlı algılayıcıya benzemektedir ve aralarındaki fark ise ęrenme kuralında grlmektedir⁸⁵.

⁸⁴ Anderson ve McNeill, a.g.e.,s.17.

⁸⁵ ztemel,a.g.e.,s.69.



Şekil 3.4: Bir ADALINE yapısı⁸⁶

Net girdisi NET ve çıktısı (Ç) olan ADALINE ünitesi şu şekilde öğrenir:

$$y = \sum_j w_j x_j + \theta \quad (3.12)$$

Eğer $NET \geq 0$ ise Çıktı (Ç)=1

Eğer $NET \leq 0$ ise Çıktı (Ç)=-1

x_j girdi, w_j her girdinin ağırlık değeri; θ ise çıktının 0'dan farklı bir değer olmasını sağlayan eşik değeridir. Beklenen değer B olduğunda hata değeri:

$$E = B - \text{Ç} \quad (3.13)$$

olacaktır. Delta kuralına göre, en küçük hata fonksiyonunu verecek ağırlıklar dereceli azalma (gradient descent) yöntemi ile bulunur. Bu yöntemle göre, her bir ağırlık değerinin hataya bağlı kısmi türevlerinin negatif (-) değerleri kullanılarak hata azaltılır. ADALINE ağırlıklarının çıktı değerini üreten aktivasyon fonksiyonu adım fonksiyonudur.

⁸⁶Gurney,a.g.e.

ADALINE modelinde herhangi bir anında hatayı azaltmak için kullanılan kural:

$$w_i(t) = w_i(t-1) + \lambda E x_i \quad (3.14)$$

Burada $w_i(t)$ ağırlığın t zamanındaki yeni değerlerini, $w_i(t-1)$ ağırlığın değişmeden önceki değerini, λ öğrenme katsayısını, E beklenen değer ile çıktı arasındaki hatayı ve x_i 'de girdi değerlerini göstermektedir.

Benzer şekilde θ eşik değeri de yine zaman içerisinde değiştirilerek olması gereken eşik değeri bulunur. Formülü ise;

$$\theta_i(t) = \theta_i(t-1) + \lambda E \quad (3.15)$$

'dir⁸⁷.

Birden fazla ADALINE ünitesinin bir araya gelerek oluşturdukları yapay sinir ağına MADALINE ağı denilmektedir. MADALINE ağı ADALINE ile aynı öğrenme kuralını kullanır. MADALINE ağıları genel olarak iki katmandan oluşmaktadır. Her katmanda değişik sayıda ADALINE ünitesi bulunur. Ağın çıktısı yine 1 ve -1 değerleri ile gösterilir⁸⁸.

Tek katmanlı algılayıcılar, doğrusal olmayan yapıları çözemedikleri için çok katmanlı algılayıcılar geliştirilmiştir.

3.2. Çok Katmanlı Algılayıcı ve Geriye Yayımlı Yapay Sinir Ağı

İlk dönemdeki tek katmanlı YSA, doğrusal problemlerin çözümünde başarılıyken, doğrusal olmayan problemlerin çözümünde başarılı değildi. Basit algılayıcı ve ADALINE ile çözülemeyen, XOR problemi YSA'ya olan güveni azalttı ve çalışmalar durma aşamasına geldi. Rumelhart, Hinton ve Williams 1986 yılında çok

⁸⁷ Baş,a.g.e.,s.46-47

⁸⁸ Öztemel,a.g.e.,s.73.

katmanlı algılayıcıyı geliştirerek, XOR problemini çözmüş ve YSA gerektiği önemi yeniden kazandırmıştır⁸⁹.

Bu model, ağı beklenen çıktısı ile ürettiği çıktı arasındaki hatayı en aza indirmek için hatayı geriye yaymasından dolayı, hata yayma modeli ya da geriye yayım modeli (backpropagation model) olarak da adlandırılmaktadır⁹⁰.

Çok katmanlı algılayıcıların YSA'ya tekrar önem verilmesini sağlamasıyla, akademik çalışmalar artmış, mühendislik problemlerinin hemen hepsine çözüm ürettiği anlaşılmıştır. Özellikle sınıflandırma, tanıma ve genelleme yapmayı gerektiren problemlerde çok önemli bir çözüm aracıdır.

Çok katmanlı algılayıcılar, delta kuralının aktivasyon fonksiyonları ve çok katmanlı ağlar için genelleştirilmiş hali olan doğrusal olmayan Genelleştirilmiş Delta Kuralı (Generalised Delta Rule) adı verilen öğrenme kuralını kullanır. Genelleştirilmiş Delta Kuralı, en küçük kareler yöntemine dayalı bir öğrenme kuralıdır. Öğrenim sırasında hem girdiler hem de o girdilere karşılık gelen çıktılar gösterildiğinden delta ve genişletilmiş delta kuralları ileri beslemeli danışmanlı öğrenme kurallarıdır.

3.2.1. Çok Katmanlı Algılayıcıların Yapısı

Çok katmanlı YSA, girdi katmanı, ara katman ve çıktı katmanı olmak üzere üç katmandan oluşmaktadır.

Girdi katmanı, dış ortamdan gelen bilgileri alarak ara katmana göndermekle görevlidir. Gelen her bilgi herhangi bir işlem yapılmadan olduğu gibi ara katmana gönderilir. Çok katmanlı algılayıcılara birden fazla girdi gelebilir ancak her işlemci elemanın tek girdisi ve tek çıktısı olur. Bu çıktı bir sonraki katmandaki tüm işlemci elemanlara gönderilir. Girdi katmanındaki her bir işlemci eleman, bir sonraki

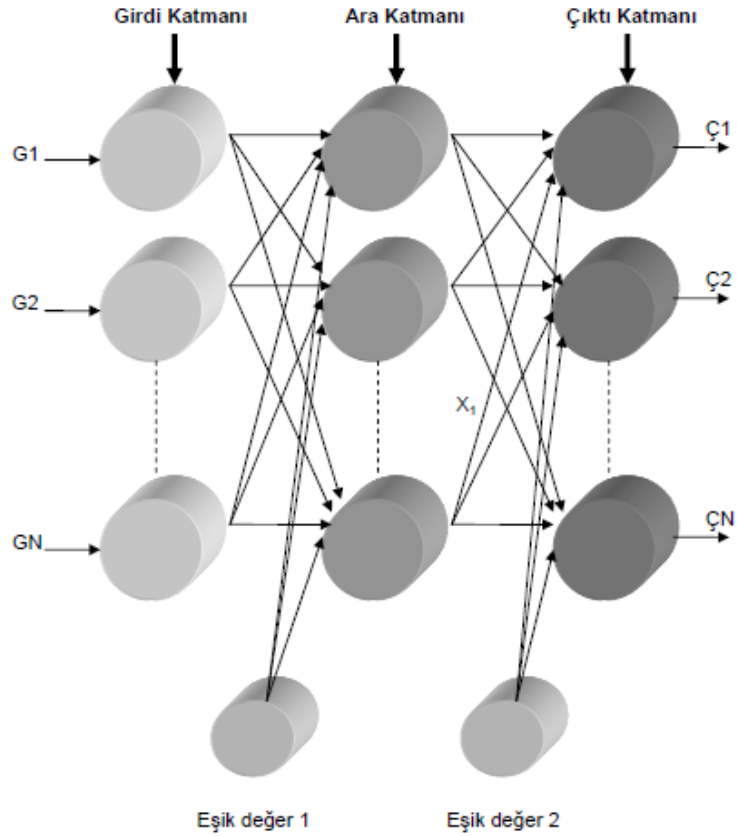
⁸⁹Kröse ve Smagt,a.g.e.,s.33.

⁹⁰ Öztemel,a.g.e.,s.76.

katmandaki tüm işlemci elemanlara bağlıdır. Girdi katmanındaki nöron sayısı uygulanan problemin giriş veri sayısı kadardır.

Ara (gizli) katman, girdi katmanından gelen bilgileri işleyerek bir sonraki katmana gönderir. Bir çok katmanlı algılayıcıda birden fazla ara katman ve her katmanda birden fazla işlemci bulunabilir. Katman sayısı ve her bir katmandaki işlemci sayısı deneme yanılma yoluyla tasarımcı tarafından bulunur. Ara katmandaki her bir işlemci eleman bir sonraki katmandaki tüm işlemci elemanlara bağlıdır.

Çıktı katmanı, ara katmandan gelen verileri işleyerek dış ortama gönderir. Bütün çok katmanlı algılayıcılarda bir çıktı katmanı bulunmakta, ancak çıktı katmanında birden fazla işlemci eleman olabilmektedir. Her işlemci elemanı bir önceki katmanda bulunan bütün işlemci elemanlara bağlıdır.



Şekil 3.5:Çok katmanlı Algılayıcı Modeli (Öztemel, 2006)

Ağa, hem örnekler hem de örneklerden elde edilmesi gereken çıktılar verilmektedir. Ağ kendisine gösterilen örneklerden genellemeler yaparak problem uzayını temsil eden bir çözüm uzayı üretmektedir. Daha sonra gösterilen benzer örnekler için bu çözüm uzayı sonuçlar ve çözümler üretebilmektedir⁹¹.

Çok katmanlı algılayıcı yapısına sahip ve öğrenme kuralı olarak geri yayılım algoritmasını kullanan ağlara, ileri beslemeli geri yayılım yapay sinir ağı adı verilmektedir. Ortaya çıkışından itibaren hem çok etkili hem de kullanımının kolay olmasından dolayı en çok kullanılan yapay sinir ağı modelidir.

İleri beslemeli geri yayılım ağlarında katman sayısı ve katmanlardaki işlemci eleman sayısı ağın başarılı sonuçlar üretebilmesi için önem taşımaktadır. Ancak bu sayıların neler olacağına ilişkin kesin çözüm yoktur. Bunun yerine uygulamalar sonucunda ortaya çıkmış ve araştırmacılar tarafından kabul görmüş bazı kurallar bulunmaktadır. Bu kurallar şöyledir⁹²:

Kural – 1 : Girdi ve çıktı verileri arasındaki ilişkinin karmaşıklık derecesi arttıkça işlemci eleman sayısı da arttırılmalıdır.

Kural – 2 : Eğer modellenen problem birçok aşamaya ayrılabilirse, fazla sayıda ara katman kullanılmalıdır. Eğer az sayıda ara katman kullanılırsa ağ başaramaz. Gereğinden fazla ara katman kullanılması durumunda ise ağ ezberlemektedir (memorization). Bu da ağın, yeni örnekler için genelleme yeteneğini azaltmaktadır.

Kural – 3 : Yapay sinir ağının eğitilmesinde kullanılan örnek setinin büyüklüğü, ara katmanlardaki işlemci elemanların sayısı için bir üst limit kriteri oluşturmaktadır. Bu üst limiti bulabilmek için önce eğitim setindeki girdi-çıkış sayısı bulunmalıdır. Bulunan bu sayı ağıdaki toplam girdi ve çıktı işlemci elemanlarının sayısına bölünmesiyle elde edilen sonuç, bir ölçeklendirme katsayısı olarak kullanılabilir. Bu katsayı genellikle 5 ile 10 arasında bir değerdir. Fazla hata içeren veri setleri için bu

⁹¹ Öztemel, a.g.e., s.44.

⁹² Anderson ve McNeill, a.g.e., s.33-34

katsayı 20 ile 50 arasında değerler alabilir. Veri setinin hemen hemen hiç hata içermemesi durumunda bu katsayı 2 seviyesine kadar düşebilir.

Bu yöntemle, ölçeklendirme katsayısını kesin olarak ulaşılamasa da bir fikir edinmek mümkün olmaktadır. Ayrıca, genelleme yeteneğinin kaybolabilmesi ve dolayısıyla yeni veriler tanıtıldığında ağın kullanışsız kalması sonucu doğurabileceğinden, bir ara katmandaki işlemci eleman sayısının çok fazla olmaması yararlı olacaktır⁹³.

3.2.2. Geri Yayılım Ağlarının Öğrenmesi

Çok katmanlı ağlar, danışmanlı öğrenme stratejisine göre çalışır. En küçük kareler yöntemine dayanan Genelleştirilmiş Delta Öğrenme Kuralında ağın öğrenebilmesi için eğitim seti adı verilen örneklerden oluşan bir kümeye daha ihtiyaç vardır. Genelleştirilmiş Delta Öğrenme Kuralı iki aşamadan oluşur⁹⁴:

1.aşama : İleri doğru hesaplama: Ağın çıktısını hesaplanır.

2.aşama : Geriye doğru hesaplama: Ağırlıklar değiştirilir.

İleri doğru hesaplama aşamasında, girdi katmanına örnekler gösterilerek, işleme tabii tutulmadan ara katmana iletilirler. Bu eşitlik, $y_k^i = x_k$ olarak gösterilebilir. Yani girdi katmanındaki k. işlemci elemanın çıktısı y_k^i olarak gösterilir.

Ara katmandaki her işlemci eleman, girdi katmanındaki bütün işlemci elemanlardan gelen bilgileri bağlantı ağırlıklarıyla (w_1, w_2) ağırlıklandırarak işleme alır. Önce ara katmandaki işlemci elemanlara gelen net girdi (s_j^a) şöyle hesaplanır:

$$s_j^a = \sum_{k=1}^n w_{kj} y_k^i \quad (3.16)$$

⁹³ Baş,a.g.e.,s.51.

⁹⁴ Öztemel,a.g.e.,s.77

w_{kj} k. girdi katmanı elemanını, j. Ara katman elemanına bağlayan ağırlık değerini; y_k^i ise girdi katmanındaki k. işlemci elemanın çıktısını gösterir. j. ara katman elemanının çıktısı ise bu net girdinin aktivasyon fonksiyonundan geçirilmesiyle hesaplanır. Aktivasyon fonksiyonu olarak türevi alınabilen bir fonksiyon kullanılmalıdır, genellikle sigmoid fonksiyonu tercih edilir. Sigmoid fonksiyon kullanılması durumunda çıktı şöyledir:

$$y_j^a = \frac{1}{1 + e^{-(s_j^a + \beta_j^a)}} \quad (3.17)$$

Burada β_j ara katmanda bulunan j. elemana bağlanan eşik değer elemanının ağırlığını göstermektedir. Bu ağırlık değeri sigmoid fonksiyonunun yönünü belirler. Eğitim sırasında ağ bu değeri kendisi belirler.

Ara ve çıktı katmanların işlemci elemanlarının çıktıları, kendilerine gelen net girdinin hesaplanması ve sigmoid fonksiyonundan geçirilmesi sonucu belirlenmektedirler. Çıktı katmanından çıkan değerler, yani çıktılar, elde edilince ağın ileri hesaplama aşaması tamamlanmış olur⁹⁵.

Geriye doğru hesaplama aşamasında ağın gösterilen girdilere göre oluşturduğu çıktılar, beklenen çıktı değerleri ile karşılaştırılır. Amaç bu çıktıların arasındaki farkın yani hatanın en aza indirilmesidir. Bu nedenle, geriye hesaplamada bu hata ağın ağırlık değerlerine dağıtılarak bir sonraki denemede hatanın azaltılması sağlanır. Çıktı katmanındaki m. işlemci eleman için oluşan hata;

$$E_m = d_m - y_m \quad (3.18)$$

⁹⁵ Öztemel, a.g.e., s.78.

olur. Bu hata, bir işlemci eleman için oluşan hatadır. Çıktı katmanı için ise bütün hataların toplanması gerekmektedir. Bazı hataların eksi çıkmasından dolayı, hatanın yanlış hesaplanmasına engel olmak için, ağırlıkların kareleri hesaplanarak, karakökü alınır. Bu hata şöyle hesaplanır:

$$E = \frac{1}{2} \sum_m E_m^2 \quad (3.19)$$

Toplam hatayı en aza indirebilmek için, hatanın işlemci elemanlarına dağıtılması gerekmektedir. Bu ise işlemcilerin ağırlıklarını değiştirerek gerçekleşir. Ağırlıkların değiştirmek için iki durum mevcuttur: ara katman ile çıktı katmanı arasındaki ağırlıkları değiştirmek ya da ara katmanlar veya ara katmanı ile girdi katmanı arasındaki ağırlıkları değiştirmektir.

Ara katman ile çıktı katmanı arasındaki ağırlıkların değiştirilmesi işlemi şöyledir: ara katmandaki j. işlemci elemanı çıktı katmanındaki m. işlemci elemana bağlayan bağlantının ağırlığındaki değişim miktarı Δw^a olmak üzere herhangi bir t zamanındaki ağırlığın değişim miktarı:

$$\Delta w_{jm}^a(t) = \lambda \delta_m y_j^a + \alpha \Delta w_{jm}^a(t-1) \quad (3.20)$$

Burada λ öğrenme oranını, α ise momentum katsayısını göstermektedir. Öğrenme oranı, ağırlıkların bir sonraki düzeltmede hangi oranda değiştirileceğini belirlemektedir. Momentum katsayısı ise ağırlık öğrenme sırasında yerel minimuma takılmasını önler. Bu değer küçük olması yerel çözümlerden kurtulmayı, büyük olması ise tek bir çözüme ulaşmayı zorlaştırabilmektedir. Yerel minimum ise ağırlık en iyi çözüme ulaşmaya çalışırken, farklı bir çözüme takılması ve daha iyi bir çözümün bulunamamasıdır. O nedenle tasarımcılar ağırlıkların ürettikleri çözümlerde ϵ (tolerans

değeri) kadar hatayı kabul etmektedirler. Tolerans değerinin altındaki herhangi bir noktada problem öğrenilmiş kabul edilir⁹⁶.

Eşitlikteki δ_m , işlemci elemanın hatasını göstermekte ve şu şekilde hesaplanmaktadır:

$$\delta_m = f'(s).E_m \quad (3.21)$$

Buradaki $f'(s)$,aktivasyon fonksiyonunun türevidir. Sigmoid fonksiyonu kullanıldığında:

$$\delta_m = y_m(1 - y_m).E_m \quad (3.22)$$

olur. Değişim miktarı hesaplandıktan sonra ağırlıkların t. iterasyondaki değerleri şöyledir:

$$w_{jm}^a(t) = w_{jm}^a(t-1) + \Delta w_{jm}^a(t) \quad (3.23)$$

Benzer şekilde eşik değer elemanın da ağırlıklarını değiştirmek gereklidir. Onun için önce değişim miktarı hesaplanır. Eğer çıktı katmanındaki işlemci elemanların eşik değer ağırlıkları β o ile gösterilirse, bu elemanın çıktısının sabit ve 1 olması nedeni ile değişim miktarı,

$$\Delta\beta_m^0(t) = \lambda\delta_m + \alpha\Delta\beta_m^0(t-1) \quad (3.24)$$

olmaktadır. Eşik değerin t. iterasyondaki ağırlığının yeni değeri ise:

$$\beta_m^0(t) = \beta_m^0(t-1) + \Delta\beta_m^0(t) \quad (3.25)$$

⁹⁶ Öztemel,a.g.e.,s.78

formülü ile hesaplanmaktadır.

Geriye doğru hesaplamada ikinci durum ara katmanlar arasındaki veya ara ve girdi katmanları arasındaki ağırlıkları değiştirilmesidir. Ara ve çıktı katmanları arasındaki ağırlıkların değiştirilmesinde her bir ağırlık için sadece çıktı katmanındaki bir işlemci elemanın hatası dikkate alınırken, girdi katmanı ile ara katman arasındaki (veya iki ara katman arasındaki) ağırlıkların değiştirilmesinde çıktı katmanındaki bütün işlemci elemanların hatasından payını alması gerekmektedir. Bu ağırlıklardaki değişim (mesela girdi ile ara katmanları arasındaki ağırlıkların değişimi) Δw^i ile gösterilirse değişim miktarı:

$$\Delta w_{kj}^i(t) = \lambda \delta_j^a y_k^i + \alpha \Delta w_{kj}^i(t-1) \quad (3.26)$$

ile hesaplanır. δ_j^a ise:

$$\delta_j^a = f'(s) \sum_m \delta_m w_{jm}^a \quad (3.27)$$

Sigmoid aktivasyon fonksiyonu kullanıldığında hata değeri ise:

$$\delta_j^a = y_j^a (1 - y_j^a) \sum_m \delta_m w_{jm}^a \quad (3.28)$$

formülü ile hesaplanır. Değişim miktarı hesaplandıktan sonra t. iterasyondaki yeni değerler:

$$w_{kj}^i(t) = w_{kj}^i(t-1) + \Delta w_{kj}^i(t) \quad (3.29)$$

olur. Ara katman eşik değeri ağırlıkları β^a ile gösterilirse değişim miktarı:

$$\Delta \beta_j^a(t) = \lambda \delta_j^a + \alpha \Delta \beta_j^a(t-1) \quad (3.30)$$

Ağırlıkları t . iterasyondaki değerleri ise şöyle hesaplanır:

$$\beta_j^a(t) = \beta_j^a(t-1) + \Delta\beta_j^a(t) \quad (3.31)$$

Böylece ağırlıkların hepsi değiştirilmiş olacaktır. Bir iterasyon hem ileri hem geri hesaplamaları yapılarak bitirilip ikinci örneğe aynı işlemler uygulanır. Bu işlemler, öğrenme gerçekleşene kadar sürdürülür.

3.3. Yapay Sinir Ağları Destekleyici Öğrenme: Doğrusal Vektör Parçalama/Niceleme Modeli (LVQ)

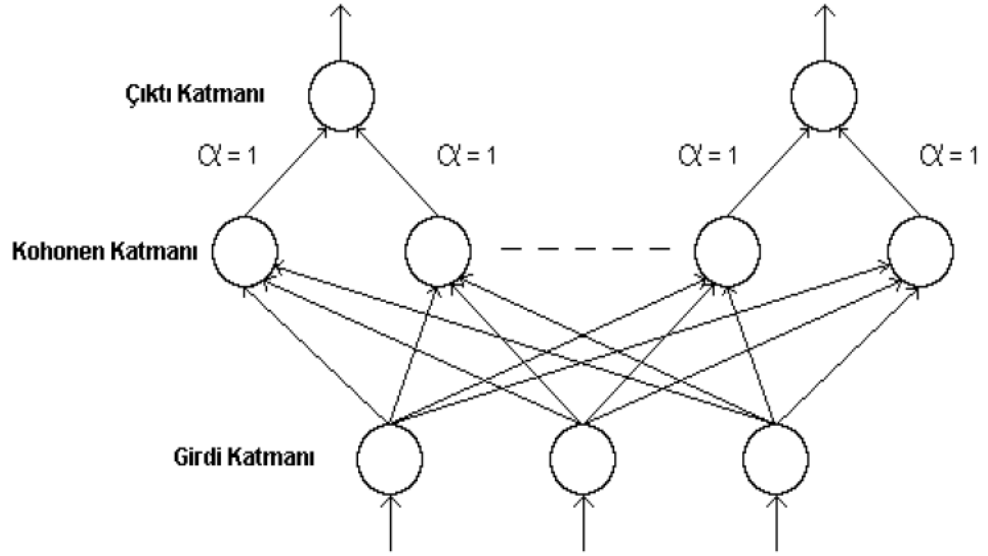
LVQ (Learning Vector Quantization), 1984 yılında Kohonen tarafından geliştirilmiştir. Bu model ve Kohonen'in LVQ ağından önce geliştirdiği SOM (Self-Organizing Maps) modeli katmanı üzerine temellendirmiştir. Kohonen katmanı n boyutlu bir vektörü bir vektörler setine uydurmaktadır (mapping)⁹⁷.

Daha çok sınıflandırma problemlerinin çözümünde kullanılan LVQ'da amaç bir vektörün belirli sayıda vektör ile gösterilmesidir. LVQ ağının öğrenmesi, girdi vektörünün hangi vektör seti tarafından temsil edildiğinin bulunması anlamına gelmektedir. Bu vektör setine referans vektörleri denirse LVQ ağının görevi, öğrenme yolu ile bu referans vektörleri belirlemektir. Yani, girdi vektörlerinin üyesi olabilecekleri vektör sınıfını belirlemektir⁹⁸.

LVQ ağları üç katmandan oluşmaktadır. İlk katman girdi katmanıdır. Örneklerin ağa gösterildiği bu katmanda bilgi işleme gerçekleşmez, gelen bilgiler, girdi vektörünü oluşturur. İkinci katman, Kohonen katmanıdır. Bu katmanda girdi setine en yakın olan ağırlık vektörü belirlenir. Bu katmandaki her eleman bir referans vektörünü gösterir. Girdi vektörü, girdi katmanı ile Kohonen katmanı arasındaki ağırlıkların oluşturduğu referans vektörlerine haritalanmaktadır. Üçüncü katman olan çıktı katmanında ise girdinin ait olduğu sınıf belirlenir.

⁹⁷ Anderson ve McNeill, a.g.e., s.41

⁹⁸ Öztemel, a.g.e., s.115



Şekil 3.5: LVQ Ağının Topolojik Yapısı

LVQ ağları, girdi ve Kohonen katmanları arasında tam bağlantılı, Kohonen ve çıktı katmanları arasında ise kısmi bağlantılı olmaktadır. Girdi katmanındaki her işlemci eleman Kohonen katmanındaki tüm işlemci elemanlarla bağlantılıdır. Kohonen katmanındaki işlemci elemanlar ise çıktı katmanındaki bir tek işlemci elemanla bağlantılıdır. Kohonen katmanı ile çıktı katmanı arasındaki ağırlıklar (α) sabit olup 1'e eşittir ve bu ağırlıklar değişmez. Ağın eğitimi sadece Kohonen katmanı ile girdi katmanı arasındaki ağırlıkların yani referans vektörlerin değerlerinin değiştirilmesi ile gerçekleştirilir. Kohonen katmanında ne kadar işlemci eleman varsa o kadar referans vektörü oluşturulur⁹⁹.

LVQ, öğretmensiz öğrenme türlerinden olan yarışmacı öğrenme (competitive learning) sınıfına giren Kohonen öğrenme kuralını kullanmaktadır. Eğitim sırasında girdilerin sınıflandırılması en yakın komşu (nearest neighbour) kuralına göre gerçekleşir. Eğitimde girdi vektörü ile referans vektörü arasındaki en kısa mesafe aranır ve girdi vektörünün en kısa mesafede bulunan vektör grubuna ait olduğu varsayılmaktadır. Çıktı değerlerinin belirlenmesinde “kazanan her şeyi alır (winner

⁹⁹ Öztemel,a.g.e.,s.117.

takes all)” stratejisi uygulanır. Ağ eğitilirken her iterasyonda ağın ürettiği çıktının değeri yerine sadece doğru olup olmadığı belirlenir. Sadece girdi vektörüne yakın olan vektörün (kazanan vektör) değerleri (ağın bu vektöre ait ağırlıkları) değiştirilir¹⁰⁰.

Kohonen katmanında işlemci elemanları birbiri ile yarışır. Kohonen ve çıktı katmanlarındaki proses elemanlarının çıktıları ikili (binary) değerlerdir. Sadece kazanan işlemci elemanının çıktısı 1 olup, diğerleri 0’dır. Hangi işlemci elemanının yarışı kazandığına öğrenme kuralı karar verir. Kohonen katmanında hangi işlemcinin çıktısı 1 olursa bağlı bulunduğu çıktı katmanındaki işlemci elemanında çıktısı 1 olur ve girdi vektörü o çıktının gösterdiği sınıfın üyesi kabul edilir¹⁰¹.

LVQ ağının öğrenme kuralı Kohonen öğrenme kuralıdır. Bu yöntemle göre, Kohonen tabakasındaki işlemci elemanlar birbirleriyle yarışır. Yarışma girdi vektörü ile referans vektörleri arasındaki öklid (euclid) mesafesinin hesaplanmasına dayanmaktadır. Hangi işlemci elemanın referans vektörü girdi vektörüne en yakın ise yarışmayı o kazanır. Girdi vektörü X ile referans vektörü a arasındaki mesafe d ile gösterilirse, i . işlemci elemanın mesafesi:

$$d_i = \|A_i - X\| = \sqrt{\sum_j (a_{ij} - x_j)^2} \quad (3.32)$$

‘dir. Bu formülde, a_{ij} ağırlık vektörünün j ., x_j ise girdi vektörünün j . değerini gösterir. Girdi vektörü ile referans vektörlerinin aralarındaki mesafe tek tek hesaplandıktan sonra hangi işlemci elemanın referans vektörü, girdi vektörüne en yakın ise yarışmayı kazanır. Öğrenme sırasında, sadece girdi katmanını bu işlemci elemana bağlayan ağırlık değerleri değiştirilir. Diğerleri değiştirilmez. Kazanan işlemci eleman için iki durum söz konusu olmaktadır.

¹⁰⁰ Kröse ve Van Der Smagt, a.g.e., s.63-64.

¹⁰¹ Öztemel, a.g.e., s.117.

İlk durumda kazanan işlemci eleman doğru sınıfın bir üyesidir. Bu durumda ilgili ağırlıklar girdi vektörüne biraz daha yaklaştırılmaktadır. Bu, aynı örnek ağı tekrar gösterildiğinde yine aynı işlemci elemanın kazanması için yapılmaktadır. Bu durumda ağırlıkların değiştirilmesi:

$$A_n = A_0 + \lambda(X - A_0) \quad (3.33)$$

Formülü ile bulunur. Burada λ , öğrenme katsayısıdır. Öğrenme katsayısı zaman içinde 0 değerini alacak şekilde monoton olarak azalır. Bunun nedeni, girdi vektörünün referans vektörüne çok yaklaştığında durması ve aksi yönde tekrar uzaklaşmamasıdır. Aksi takdirde ters yönde tekrar uzaklaşma meydana gelecektir.

İkinci durumda ise, kazanan işlemci eleman yanlış sınıftandır. Bu durumda ağırlık vektörü girdi vektöründen uzaklaşmaktadır. Bunun amacı, bir daha aynı örnek geldiğinde aynı işlemci elemanın kazanmamasıdır.

$$A_n = A_0 + \lambda(X - A_0) \quad (3.34)$$

Öğrenme katsayısının zaman içinde sıfır değerini alana kadar azalması burada da geçerli olmaktadır.

LVQ'da öğrenme esnasında yalnızca referans vektörlerinin ağırlıkları değiştirilir. Girdi katmanı ile çıktı katmanı arasındaki ağırlıklar (α) sabit ve 1 değerine eşittir. Kohonen katmanındaki her işlemci elemanın çıktısı y^k olmak üzere:

$$y_i^k = \begin{cases} 1 & \text{Eğer i. işlemci eleman yarışı kazanırsa} \\ 0 & \text{Aksi halde} \end{cases} \quad (3.35)$$

Kohonen katmanındaki işlemci elemanların çıktıları, bu işlemci elemanları çıktı katmanına bağlayan ağırlık değerleri ile çarpılarak ağı çıktısı hesaplanır:

$$y_i = \sum_j y_j^k \alpha_{ki} \quad (3.36)$$

Bu formül, Kohonen katmanında yarışmayı kazanan işlemci elemana bağlı olan çıktı elemanın değerinin 1, diğerlerinin değerinin 0 olması anlamına gelmektedir. Ağın çıktıları belirlendikten sonra çıktının doğru sınıflandırılıp sınıflandırılmadığı sorgulanır. Buna göre Kohonen katmanındaki yarışmayı kazanan işlemci elemanı girdi katmanına bağlayan ağırlıklar değiştirilmektedir.

Eğitim setindeki tüm örnekler doğru sınıflandırılncaya kadar bu işlemler tekrarlanır. Hepsi doğru sınıflandırılınca öğrenme gerçekleştirilmiş olur. Anlatılan bu model standart bir LVQ modelidir.

LVQ modelinin bazı dezavantajları bulunmaktadır. Bunlardan ilki öğrenme katsayısının zamanında 0 değerini almaması durumunda ağın, doğru ağırlık değerlerinden uzaklaşmasıdır. Ayrıca bazı problemlerde sürekli aynı referans vektörü yarışmayı kazanmakta olduğundan ağın esnekliği ortadan kalmaktadır. Diğer dezavantaj ise sınıflandırmayı yaparken iki sınıfın tam ortasında veya sınırlara çok yakın bulunan vektörlerin hangi sınıfa gireceklerinin belirlenememesidir. Bu sorunları ortadan kaldırmak için LVQ2, Ceza Mekanizmalı LVQ ve LVQ-X modelleri geliştirilmiştir¹⁰².

3.3.1. LVQ2 Ağı

Kohonen tarafından geliştirilmiştir. LVQ2, LVQ modelinin uygulanmasından sonra elde edilen çözümün iyileştirilmesi için kullanılır. Sınır değerlerdeki vektörlerin yanlış sınıflandırılması önlenmeye çalışılır. Aynı anda iki referans vektörünün ağırlıkları değiştirilir. Bu vektörlerin ağırlıklarının değiştirilmesi için iki koşulun sağlanması gerekmektedir.

¹⁰²Fırat Bayır, Yapay Sinir Ağları ve Tahmin Modellemesi Üzerine Bir Uygulama, İstanbul Üniversitesi Sosyal Bilimler Enstitüsü Yüksek Lisans Tezi, 2006, s.77.

1. Girdi vektörüne en yakın iki vektörden, birincisi (en yakın olan) yanlış, ikincisi doğru sınıftadır.
2. Girdi vektörü, bu iki vektörün arasında merkezi olarak belirlenmiş bir aralık içerisinde kalmaktadır.

Bu durumda iki vektörün de ağırlıkları değiştirilir. Bu ağırlık vektörlerinin yeni değerleri A_{1N} ve A_{2N} şöyle hesaplanır:

$$\begin{aligned} A_{1N} &= A_{10} - \lambda(X - A_{10}) \\ A_{2N} &= A_{20} - \lambda(X - A_{20}) \end{aligned} \tag{3.37}$$

Burada A_{10} ve A_{20} ağırlık vektörlerinin değişmeden önceki değerlerini göstermektedir.

3.3.2. Ceza Mekanizmalı LVQ

Standart LVQ ağlarında bazı ağırlık vektörleri sürekli kazanmakta ve bu da ağıın esnekliğini bozmakta, diğer ağırlık vektörleri referans vektörü olma niteliklerini kaybetmektedir. Belirlenen her ağırlık vektörü girdinin bir yönünü göstermesi gerekirken sadece bir tanesi tüm sorumluluğı üstlenmektedir.

Bu problemten kurtulmak için 1988 yılında DeSieno tarafından Ceza Mekanizmalı LVQ modeli geliştirilmiştir. Bu modelde sürekli kazanan ağırlık vektörü cezalandırılmakta ve peş peşe kazanması engellenmektedir. Sürekli kazanan vektöre kaç defa kazandığı ile ilgili olarak uzaklığına bir değer eklenir, yani uzaklaştırılır. Böylece, diğer vektörlere de kazanma şansı tanınmaktadır¹⁰³.

¹⁰³ Bayır,a.g.e.,s.78.

3.3.3. LVQ-X Modeli:

Aynı anda iki ağırlık vektörünün değerleri değiştirilir. Böylece ağırlık hem öğrenme hızı hem de genelleme yeteneği artırılmaktadır. Bu öğrenme kuralı Öztemel tarafından geliştirilmiştir. Her iterasyonda global kazanan ve yerel kazanan olmak üzere iki işlemci eleman belirlenir. Mutlak kazanan girdi vektörüne en yakın ağırlık vektörüne sahip olan işlemci, yerel kazanan ise, doğru sınıf içerisinde girdi vektörüne en yakın ağırlık vektörüne sahip işlemcidir.

Eğer global kazanan işlemci eleman doğru sınıf içerisinde değilse onun ağırlık vektörü o girdi vektöründen uzaklaştırılır. Aynı zamanda doğru sınıf içerisindeki en yakın olan işlemci elemanın (yerel kazanan) ağırlık vektörü de girdi vektörüne yaklaştırılır. Bu doğru işlemci elemana daha sonraki iterasyonlarda kazanma şansı vermektedir.

Eğer global ve yerel kazanan aynı işlemci eleman ise o zaman tek bir ağırlık vektörü değiştirilmekte ve bu ağırlık vektörü girdi vektörüne yaklaştırılmaktadır¹⁰⁴.

3.4. Yapay Sinir Ağı Modeli (Öğretmensiz Öğrenme) Adaptif Rezonans Teori (ART) Ağları

Hem ileri hem geri beslemeli bir ağ olan ART ağları, Grossberg'in 1976 yılında beyin fonksiyonlarını açıklamaya yönelik çalışmaları sonucunda ortaya çıkmıştır. Bu model, biyolojik sisteme ait şu üç temel üzerine kurulmuştur¹⁰⁵.

- a) **Normalizasyon** : Biyolojik sistemler bulundukları çevredeki değişimlere duyarlıdır ve bu değişikliklere uyum sağlayabilme özelliğine sahiptirler. Örneğin, gürültülü bir ortamda insanın bir süre sonra gürültüden rahatsız

¹⁰⁴ Öztemel, a.g.e., s.125.

¹⁰⁵ Kröse ve Van Der Smagt, a.g.e., s.69.

olmaması onun ortama uyum sağladığını ve çevresindeki olayları normalize ettiğini göstermektedir.

- b) **Ayrıştırılabilme:** İnsanların ayırt edilmesi zor olan ayrıntıları ayrıştırabilme yeteneği bazen hayat kurtaran bir özelliktir. Örneğin insan, dinlenen bir aslanla saldırmak üzere olan bir aslan arasındaki farkı anlayabilecek bir sisteme sahiptir.
- c) **Ayrıntıların Saklandığı Kısa Dönemli Hafıza:** Belirlenen farklılıklar ve çevresel değişimler davranışlara dönüşmeden önce hafızada saklanmakta, sonrasında ise eyleme dönüşmektedir. Bilgilerin geçici olarak tutulduğu ve zaman içinde yok olarak yerine yeni bilgilerin saklandığı yer kısa dönemli hafızadır. Ancak sürekli aynı şeylerin tekrar edilmesi sonucunda bilgiler kısa dönemli hafıza yerine uzun dönemli hafızada saklanabilmektedir. Uzun dönemli hafıza bilgilerin sürekli tutulduğu ve kolay kolay unutulmadığı hafıza türüdür.

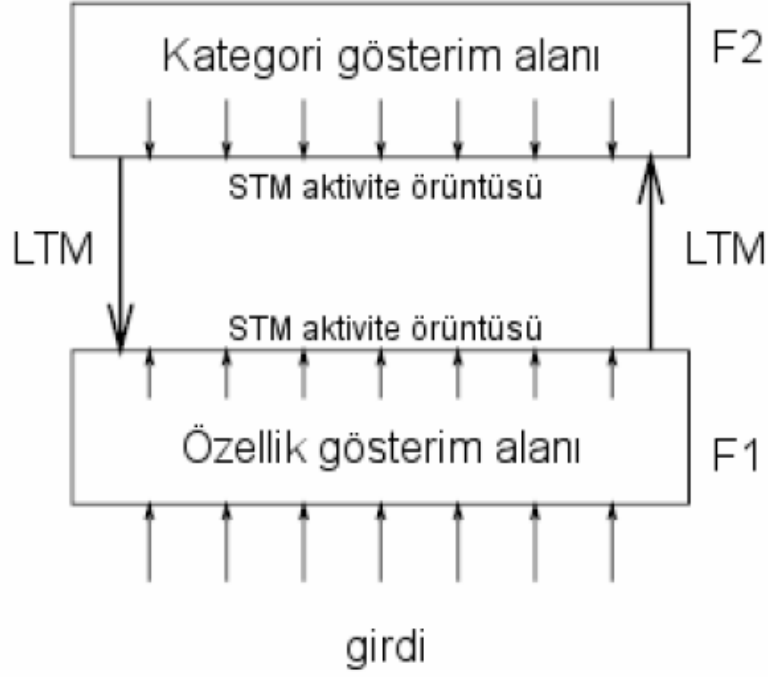
Grossberg ve arkadaşları beynin bu özelliklerinden yola çıkarak, beynin kullandığı sezgisel yaklaşımları matematik modele dönüştürüp ART ağlarını oluşturmuşlardır. Sınıflandırma problemleri için geliştirilmiş olan ART ağları, öğretmensiz öğrenme ağlarına da öncülük etmiştir. Ayrıca, gerçek zamanda çalışabilir ve çevrimiçi öğrenebilirler.

ART, adını öğrenme ve hatırlama arasındaki karşılıklı etkileşimi gerçekleştiren yöntemden yani rezonanstan almıştır. ART ağlarında işlemci elemanların çıktıları katmanlar arasında sürekli olarak ileri geri hareket etmektedir. Bu esnada eğer örnek girdi belirlenmiş bir sınıfa uyuyorsa ağ kararlı hale gelmekte ve ART ağı rezonanstadır denilmektedir¹⁰⁶.

ART ağları, genel olarak iki katmandan oluşur. Bu katmanlar F1 ve F2 olarak isimlendirilmiştir. F1 katmanı girdinin özelliklerini, F2 katmanı kategorileri (ayrıştırılmış sınıfları) gösterir. Bu iki katman birbirlerine LTM ile

¹⁰⁶Sevinç Gülseçen, Yapay Sinir Ağları, İşletme Alanında Uygulaması ve Bir Örnek Çalışma, İstanbul Üniversitesi Sosyal Bilimler Enstitüsü Basılmamış Doktora Tezi, 1993, s.89

bağlanırlar. Girdi bilgileri F1 katmanından alınarak, F2 katmanında sınıflandırma yapılmaktadır. Şekil4.10, ART ağının şematik yapısını örneklemektedir.



Şekil 3.6: ART ağlarının genel yapısı¹⁰⁷

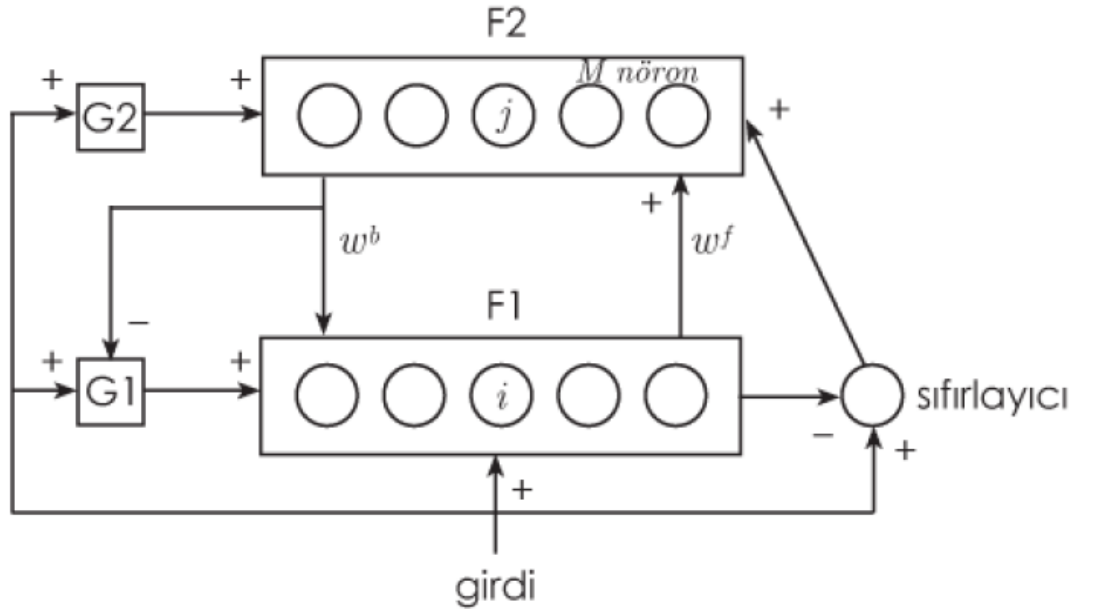
ART ağlarında girdiler sınıflandırılmadan önce, girdilerin özellikleri incelenerek F1 katmanının aktivasyonu belirlenir. LTM’de ki bağlantı değerleri ile girdiler kategorilere ayrılarak F2 katmanına gönderilir. F2 katmanındaki sınıflandırma ile F1 katmanından gelen sınıflandırma birbirleri ile eşleştirilerek, eğer örnek belirlenmiş bir sınıfa uyuyorsa o kategoriye ait olarak kabul edilir. Aksi takdirde, ya yeni bir sınıf oluşturulmakta ya da girdinin sınıflandırması yapılmaktadır.

ART ağlarının çalışması, aşağıdan yukarı (F1’den F2’ye) bilgi işleme ve yukarıdan aşağı (F2’den F1’e) bilgi işleme olmak üzere iki şekilde olmaktadır. Aşağıdan yukarı bilgi işleme durumunda bir girdi örüntüsü ağa gösterilir. Bu örüntü hem F1 katmanında STM’de X aktivite örüntüsünü oluşturur hem de oryantasyon sistemini

¹⁰⁷Kröse, van der Smagt,a.g.e.,s.70.

veya diğ er bir deyiřle reset'i (yeniden yerleřtirme mod ul n ) aktif etmek  zere bir iřaret (signal) g nderir. Benzer řekilde oluřturulan X  r nt s  hem reset'e bir engelleyici (inhibitory) iřaret g ndermekte hem de F1 katmanından bir  ıkt   r nt s  oluřturmaktadır. Bu  ıkt   r nt s  F2 katmanına giden bir girdi  r nt s ne d n řt r l r. Bu girdi  r nt s  ise F2 katmanının  ıktısı olan  r nt y  oluřturur. Bu aynı zamanda ağıın  ıktısıdır. Bu řekilde ařağıdan yukarı (F1 katmanından F2 katmanına) bilgi iřleme tamamlanmıř olmaktadır.

Yukarıdan ařağı bilgi iřleme de benzer řekilde ger ekleřmektedir. Bu durumda, F2 katmanında oluřturulan  ıkt   r nt s  yukarıdan ařağı bir sinyal g nderir. Bu sinyal daha sonra beklenen  r nt ye d n řt r l r. Aynı zamanda kontrol fakt r  (kazan ) i in engelleyici (inhibitory) bir iřaret g nderir. Bundan sonra beklenen  r nt n n girdi  r nt s  ile eřlenip eřlenemeyeceğıne bakılır. Eėer b yle bir eřleřme m mk n deėilse F1 katmanında yeni bir STM  r nt s  (X^*) oluřturulur. Bu  r nt  oryantasyon sistemindeki engelleyici iřaretin etkisini azaltmaktadır.



řekil 3.7: ART Ağıının  alıřma řekli

Oluřturulan X^* sinyali oryantasyon sistemindeki engelleyici iřaretin etkisini azaltarak reset'in F2 katmanına bir sinyal g ndermesini saėlar. Bu iřaret F2 katmanında Y^*  r nt s n  oluřturur. B ylece ağıa g sterilen girdi  r nt s  i in

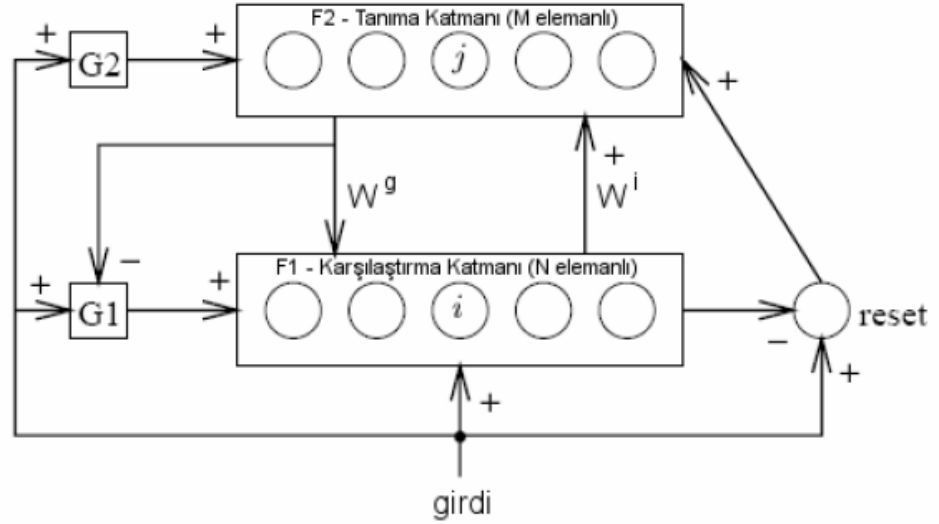
doğru sınıfı gösteren Y^* çıktısı üretilir. Eğer üretilen örüntü ile girdi örüntüsü eşleşirse o zaman sadece yukarıdan aşağı o girdinin sınıfını gösteren ağırlıklar değiştirilir. Bu değiştirme öğrenme kuralına göre gerçekleştirilmektedir¹⁰⁸.

1976 yılında ortaya çıkan ART ağlarının, günümüze kadar farklı çeşitleri ortaya çıkmıştır. ART1, ART2, ART3, ARTMAP, Fuzzy ART gibi ağları saymak mümkündür. Bu ağların aynı temele dayanmakla birlikte öğrenme kurallarında çok az farklılıklar göstermektedir. Bu çalışmada ART1 ve ART2 ağlarından bahsedilecektir.

3.4.1. ART1 Ağı:

ART1 ağı, sadece ikili (binary) girdiler ile çalışan en basit ART ağıdır. Geliştirilen ilk ART ağı olduğu söylenebilir. İki katmandan oluşan, ART1 ağında, F1 katmanı karşılaştırma katmanı, F2 katmanı tanıma katmanı olarak adlandırılır. F1 katmanındaki bütün işlemci elemanlar F2 katmanındaki bütün işlemci elemanlarla bağlantılıdır. Bu bağlantılar sürekli değerlerden oluşan ve ile gösterilen LTM bağlantılarıdır. Bu bağlantıların özellikleri ileri doğru bağlantılar olmalarıdır. Aynı zamanda F2 katmanından F1 katmanına geriye doğru ikili değerlerden meydana gelen ve w^g ile gösterilen kazanç modülü ve reset vardır.

¹⁰⁸ Baş,a.g.e.,s.70.



Şekil 3.8: ART1 Modeli ¹⁰⁹

Karşılaştırma katmanındaki her işlemci elemanın 3 girdisi bulunur. Bunlar:

- Girdi örneğinin bir elemanı (aşağıdan yukarı)
- Geri besleme değeri (yukarıdan aşağı)
- Kazanç değeri (G1'den gelen değer)

F1 katmanındaki işlemci eleman bu üç girdiden en az ikisinin aktif olması durumunda 1 değerini oluşturmaktadır. Buna 2/3 kuralı denilir.

F2 katmanındaki işlemci elemanlar kendilerine gelen net girdiyi hesaplarlar ve birbirleri ile yarışır. Genellikle en fazla çıktıyı üreten işlemci eleman yarışmayı kazanır.

G1 modülü girdi örüntüsünün değerlerini hafızada bulunan değerler ile eşleştirmede kullanılmaktadır. Gelen bilgilerin aşağıdan yukarı veya yukarıdan aşağı olup olmadığının bilinmesini sağlar. Aşağıdan gelen değerler bu kazanç değerini aktif, yukarıdan gelenler ise pasif yaparlar.

G2 girdi örneğindeki bütün elemanlar arasında mantıksal operatörlerdir, sadece yeni bir sınıfın oluşmasında kullanılmaktadırlar.

¹⁰⁹Kröse, van der Smagt, a.g.e., s.71

Reset ise girdi vektörü ile F1 katmanı çıktısı (çıkı vektörü) arasındaki farkın belirli bir değeri aşması durumunda F2 katmanına bir sinyal göndermektedir. İki vektörün arasındaki fark benzerlik katsayısı (vigilance) denilen bir değer ile karşılaştırılır. Bu katsayı girdi vektörünün hafızada bulunan çıkı vektörlerine uygunluğunu belirler. Aradaki fark benzerlik katsayısından küçükse o zaman benzerlik yok demektir. Bu durumda girdi vektörü için yeni bir sınıf oluşturmak ve hafızaya almak gerekir.

ART1 ağının eğitilmesi sırasında önce, ağırlıklara başlangıç değerleri atanmalıdır. F1 katmanından ağa sunulan örnek sayısı N, F2 katmanında çıkı işlemci eleman sayısı M ise, $M \geq N$ olmalıdır. Çünkü N adet örneğin hepsinin birbirinden farklı olması durumunda ağın her biri için bir sınıf ayıracak durumda olması gerekir. Bazı uygulamalarda sınıf sayısı belirli bir sayı ile sınırlandırılmaktadır. Bu durumda ağın görevi, girdi setini belirlenen sayıda sınıfa ayırır. Bunu sağlamak için de benzerlik katsayısı önemlidir. ART ağlarının başlangıç değerlerinin atanmasında da iki durum söz konusudur:

1. F2 ve F1 arasındaki geriye doğru ağırlıkların bütün değerleri başlangıçta 1 değerini alırlar.

$$w_{ji}^g = 1 \quad (3.38)$$

$j: 1, 2, 3, \dots, M \quad i: 1, 2, 3, \dots, n$

2. F1 ve F2 arasındaki ileri doğru ağırlıkların başlangıç değerleri, F1 katmanındaki işlemci eleman sayısı n olmak üzere,

$$w_{ij}^i = \frac{1}{1+n} \quad (3.39)$$

Başlangıç değerlerinin atanmasından sonra benzerlik katsayısı belirlenir. Benzerlik katsayısı ρ ile gösterilir ve değeri 0 ile 1 arasındadır. Bu katsayı, iki vektörün aynı sınıfın elemanı sayılabilmesi için birbirlerine ne kadar benzemesi gerektiğini belirler. Bu değerin 0.8 olması yani benzerliğin %80 olması, girdi vektörünün çıkı vektörü ile aynı sınıfın elemanı olması için yeterlidir. Bunun altında bir benzerlik aynı grubun elemanı olmak için yeterli değildir.

Benzerlik katsayısı ne kadar büyük olursa, sınıf sayısı da o kadar çoğalmaktadır. Bu sayı düşük olduğunda, sınıf sayısı da az olmaktadır. Fakat bu durumda, ağın öğrenme performansının düşük olabilme ihtimali vardır. Sınıf sayısının az veya çok olmasından çok öğrenilmesi istenilen olayı doğru temsil edebilen sınıf sayısını oluşturacak şekilde benzetim katsayısı belirlenmektedir.

Bir sonraki aşamada girdi setindeki örnek $X(x_1, x_2 \dots \dots x_n)$ vektörü olarak (her bir elemanı x_i olarak tanımlanmış şekilde) ağa gösterilir.

Girdi setindeki örneğin ağa gösterildikten sonra F2 katmanındaki her işlemci elemanın çıktı değeri:

$$y'_j = \sum_i^N w_{ij}^i(t)x_i \quad (3.40)$$

j:1,2,3.....M i:1,2,3,.....n

formülü ile hesaplanır. Kazanan eleman, en büyük çıktıya sahip işlemci elemandır. Bu elemanın sahip olduğu ağırlık vektörüne en uygun sınıf (kategori) gösterim vektörü denilmektedir. Bunun k. işlemci eleman olduğu varsayılırsa kazanan elemanın çıktısı:

$$y_k^* = \max(y'_j) \quad (3.41)$$

Olmaktadır. Burada * işareti kazanan elemanı ifade etmektedir. Bu elemanın ağırlık vektörü girdi vektörü ile karşılaştırılarak benzerlik katsayısına göre uygunluk sınaması yapılır.

Bu aşamada, kazanan elemanın ağırlık vektörü ile girdi vektörünü temsil edip edemeyeceğine karar verilir. Bunun için önce girdi vektöründe bulunan 1 sayısı s1 belirlenmelidir:

$$s1 = |X| = \sum x_i \quad (3.42)$$

Daha sonra kategori gösterim vektörü (kazanan elemanın ağırlık vektörü) ile girdi vektörünün uyuştugu 1 sayısı (s2) bulunmalıdır:

$$s2 = |w_k^g \cdot X| = \sum w_{jk}^g x_i \quad (3.43)$$

Eğer $\frac{s2}{s1} \geq \rho$ ise iki vektör birbirinin benzeri kabul edilmektedir. Yani kategori gösterim vektörü girdi vektörünü temsil ediyor denmektedir. Bu durumda ağırlıklar şu şekilde değiştirilmelidir:

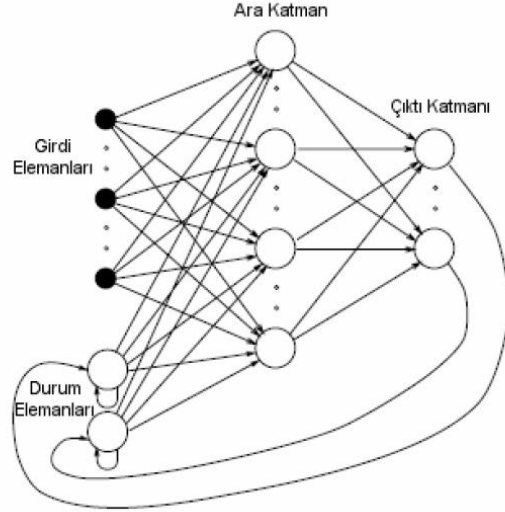
$$w_{jk}^g(t+1) = w_{jk}^g(t)x_i \quad (3.44)$$

$$w_{ik}^j(t+1) = \frac{w_{jk}^g(t)x_i}{0.5 + \sum_i^N w_{ki}^g(t)x_i} \quad (3.45)$$

Eğer $\frac{s2}{s1} < \rho$ ise o zaman en büyük çıktıyı veren işlemci elemanın çıktısı 0 olarak atanır. Ondan sonraki en büyük çıktıyı veren işlemci eleman seçilerek F2 katmanındaki işlemci elemanların çıktılarının hesaplanmasından itibaren işlemler tekrar edilir. Girdi vektörü bu sınıfın elemanı olarak atanmaz ise üçüncü en büyük çıktıyı veren işlemci elemanın sınıf gösterim vektörü ile benzerliğine bakılarak benzerlik olması durumunda ağırlıklar değiştirilmelidir. Başlangıçta birinci örnek birinci sınıfın temsilcisi olarak atanır. İkinci örnek birinci ile aynı sınıftan ise (benzer ise) birinci sınıfın, değilse ikinci sınıfın temsilcisi olarak sayılır. Böylece örneklerin hepsi ya ağırlıkları değiştirilerek var olan sınıflardan birine girmekte ya da yeni bir sınıfın temsilcisi olmaktadır. En kötü olasılık ile N örnek için N adet sınıf oluşturulabilir (N=M). Bu şekilde bu işlemler, bütün elemanlar sınıflandırılincaya kadar devam eder.

3.5. Jordan Ağları

İlk geri beslemeli (yinelemeli) ağlardan biri olan Jordan ağları, çok katmanlı perceptronlara benzer yapıdadırlar. Girdi, çıktı ve ara elemanlara ek olarak durum elemanları (state units) adı verilen özel işlemci elemanlar da bulunmaktadır. Durum elemanları, çıktı katmanından aldıkları aktivasyon değerlerini bir sonraki iterasyona girdi olarak taşımakla görevli elemanlardır. Durum elemanlarının sayısı ile ilgili bir kısıt yoktur. Diğer katmanlardaki işlemci elemanlar, çok katmanlı perceptronlardaki işlemci elemanlara benzer şekilde çalışırlar. Ara katman elemanları hem doğrusal hem de doğrusal olmayan aktivasyon fonksiyonlarına sahip olabilirler. Durum elemanları ile çıktı elemanları arasındaki bağlantı ağırlıkları sabittir. Durum katmanındaki her işlemci elemanın kendisine bağlantısı bulunmaktadır. Çok katmanlı perceptronların kullandığı öğrenme kuralları, Jordan ağının eğitiminde de kullanılabilir¹¹⁰ (Kröse, van der Smagt, 1996).



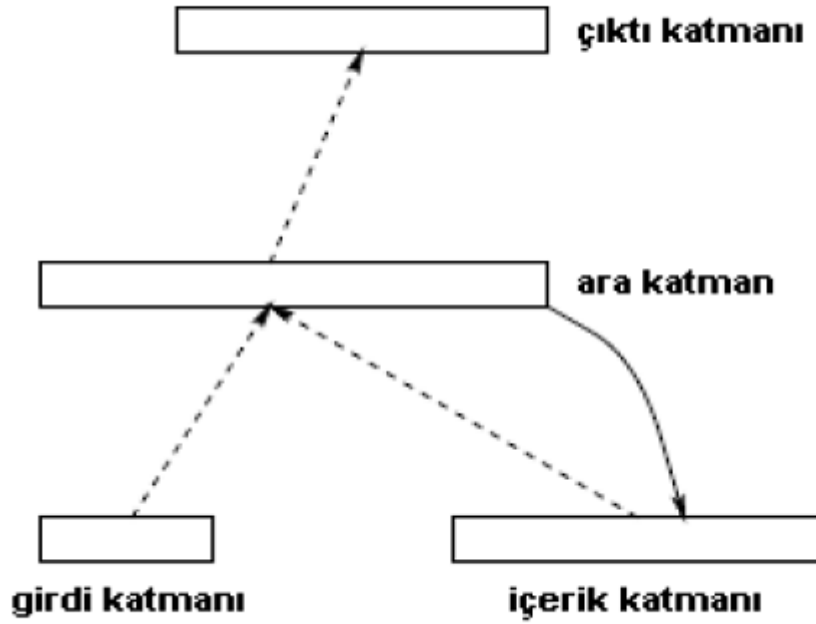
Şekil 3.9: Jordan Ağı¹¹¹

¹¹⁰Kröse ve Van Der Smagt,a.g.e.,s.48

¹¹¹ A.e.,s.48.

3.6. Elman Ağı:

Elman ağı, 1990 yılında Elman tarafından geliştirilmiştir. Elman ağı, girdi katmanı, çıktı katmanı, ara katmanlar ve bunlara ek olarak içerik katmanından (context layer) oluşmaktadır. İçerik elemanları, ara katmandan aldıkları aktivasyon değerlerini bir sonraki iterasyona girdi olarak taşırlar. Elman ağları, Jordan ağına oldukça benzemekle birlikte iki önemli farka sahiptir. Bunlardan ilki geri besleme yaptıkları aktivasyon değerlerini çıktı katmanından değil ara katmandan almaları, ikincisi ise içerik elemanlarının kendilerine bağlantılarının olmamasıdır. Elman ağlarında da ara elemanlar ve içerik elemanları arasındaki bağlantı ağırlıkları sabittir ve +1'dir¹¹².



Şekil 3.10: Elman Ağı¹¹³

Elman ağının öğrenmesi, Genelleştirilmiş Delta öğrenme kuralına göre gerçekleşir. Ara katmanda bulunan işlemci elemanlara gelen net girdi değeri, girdi katmanındaki elemanın değeri (u) ile ağırlığının (W^g) çarpılıp toplanması sonucu bulunan

¹¹² Baş, a.g.e., s.81.

¹¹³ Kröse ve Van Der Smagt, a.g.e., s.49.

değerlere içerik elemanından gelen bağlantı değerlerinin ara katmanlarının (W^i) bir önceki aktivasyon değerleri ile çarpılıp toplanması sonucu bulunur. Bu değer bir fonksiyondan geçirilerek (f) ara katman elemanının çıktısı (X) bulunur. Yani herhangi bir t zaman diliminde kullanılan aktivasyon fonksiyonunun sigmoid olması durumunda ara katman elemanlarının çıktıları şöyle hesaplanır:

$$x_i(t) = \frac{1}{1 + e^{-s_i(t)}} \quad (3.46)$$

Burada hesaplanan net girdi s ise ara katmanlardan gelen geri beslemeler dikkate alınarak hesaplanır. Bilinen geri yayılım algoritmasında hesaplanan net girdi değerindeki farklılık buradadır. Yani Elman ağında,

$$s_k = W^g u(t) + W^i X(t-1) \quad (3.47)$$

Elman ağlarında, çok katmanlı algılayıcılar gibi, kullanılan aktivasyon fonksiyonlarının türevi alınabilir fonksiyonlar olması gerekmekte ve genelleştirilmiş delta kuralına göre hem işlemci elemanların çıktıları hesaplanmakta , hem de ağırlıkların değişim miktarları belirlenmektedir. Ağıın çıktısı ise çıktı katmanına gelen net değer doğrusal fonksiyondan geçirilmesi ile belirlenir. Yani, çıktı katmanındaki işlemci elemanların aktivasyon fonksiyonları doğrusal olduğu için, herhangi bir t zamanında çıktı katmanındaki çıktı elemanının değeri, W^a ağırlık değerleri ile ara katman elemanlarının çıktıları (X) kullanılarak:

$$y(t) = W^a(t)X(t) \quad (3.48)$$

formülüyle hesaplanır. t. zaman diliminde çıktı katmanındaki j. elemanın çıktısı:

$$y_j(t) = \sum_{i=1}^M w_i^a(t) x_i(t) \quad (3.49)$$

‘dir. Böylece ağın ileri doğru bilgi işlemesi tamamlanmış olur. Elman ağına sunulan herbir örnek için beklenen çıktı belirli olduğu için, t. zaman diliminde j. çıktı elemanında oluşan hata (E_j) :

$$E_j = d_j(t) - y_j(t) \quad (3.50)$$

‘dir. Bu hata, genelleştirilmiş delta kuralına göre değişebilen ağırlıklara dağıtılmaktadır. Çıktı fonksiyonunun sigmoid olması durumunda t. zaman diliminde ağırlıklara dağıtılacak olan hata (δ):

$$\delta(t) = y(t) - [1 - y(t)]E(t) \quad (3.51)$$

Ağırlıkların değişimi için önce değişimin miktarı hesaplanarak bu değişim miktarı ağırlıklara eklenir. Genelleştirilmiş Delta Kuralı’na göre ağırlıkların değişimi için verilmiş olan denklemler Elman ağı için de geçerlidir. Burada dikkat edilmesi gereken nokta, geri besleme ağırlıklarının değerlerinin sabit olduğu ve değiştirilmeyeceğidir. Yani ağırlıkların değerleri değiştirilirken geri besleme ağırlıklarını dikkate almamak gerekir. Bu ağırlıklar ileri doğru bilgi işlerken içerik elemanlarının girdisini oluşturmada (geri besleme değerlerini girdi olarak içerik elemanlarına taşımada) kullanılırlar. Ayrıca ağın ağırlıklarını değiştirirken geri besleme bağlantı ağırlıklarının dikkate alınmaması ve içerik elemanlarının girdi elemanları olarak düşünülmesi durumunda Elman ağı çok katmanlı algılayıcı ile aynı olur¹¹⁴.

¹¹⁴ Öztemel,a.g.e.,s.169

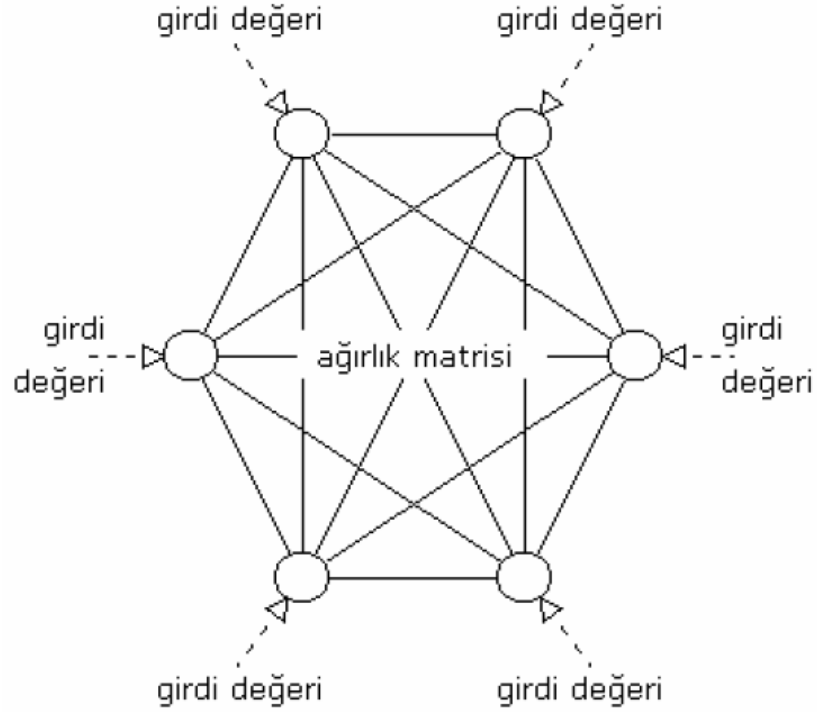
3.7. Hopfield Ağı

1982 yılında John Hopfield tarafından geliştirilen Hopfield ağı, tek katmanlı ve geri dönüşümlü bir ağıdır. Hopfield ağı, beyine benzer bir şekilde belirli hafızaları veya desenleri depolayan basit bir sinir ağıdır. Basit yapısı ve matematiksel tutarlılığı nedeniyle YSA çalışmalarında önemli yeri olan ve tercih edilen bir modeldir. Her probleme uygulanamamasına rağmen, daha karmaşık modelleri anlamak için giriş bilgisi sağlamaktadır¹¹⁵.

Hopfield ağı, genellikle ikili (0 veya 1) ve çift kutuplu (-1 veya +1) girişleri kabul etmektedir. Hopfield ağlarında her işlemci eleman, diğer işlemci elemanlarla tam bağlantı halindedir ve hem girdi hem de çıktı elemanıdır. Bağlantılar çift yönlüdür (bilgi her iki yönde de akmaktadır) vesimetriktir¹¹⁶.

¹¹⁵Gülseçen,a.g.e.,s.78.

¹¹⁶ Bayır,a.g.e.,s.67



Şekil 3.11: Hopfield ağınnın yapısı¹¹⁷

Ağın bağlantı değerleri bir enerji fonksiyonu olarak saklanır. Her iki yönde akan bilgiye uygulanan ve her bağlantı için hesaplanan ve değişmeyen bir ağırlık değeri vardır. Her işlemci eleman diğer işlemci elemanlardan gelen girdilerin toplamını bir eşik değeri ile karşılaştırmakta ve diğer işlemci elemanlara çıktı olarak iletmektedir.

Hopfield ağında örnek örüntü seçilmekte ve ağın ağırlıklarının başlangıç değerlerini saptamak için kullanılmaktadır. Bu bir kere yapıldıktan sonra herhangi bir örüntü ağına sunulmakta ve bu da giriş örüntüsüne en çok benzeyen örüntülerden biriyle sonuçlandırılmaktadır. Çıktı örüntüsü, birimlerin durumlarına bakılarak ağdan okunabilmektedir¹¹⁸.

¹¹⁷ Jochen Fröhlich, Neural Net Overview (1997), (Çevrimiçi), <http://fbim.fh-regensburg.de/~saj39122/jfroehl/diplom/e-1.html>, 2010.

¹¹⁸ Çetin Elmas, Yapay Sinir Ağları: Kuram, Mimari, Eğitim, Uygulama”, Seçkin Yayıncılık, Ankara, 2003.

Hopfield ağıları çıktı değerlerinin kesikli ve sürekli oluşlarına göre ikiye ayrılır. Kesikli Hopfield ağıları, çağrışımlı bellek (associative memory) olarak kullanılır. Sürekli Hopfield ağıları ise kombinatoriyel optimizasyon problemlerinin çözümünde kullanılır¹¹⁹.

Kesikli Hopfield Ağları, Birleşik (ilişkili) Ağlar (Associative Networks) sınıfına girmektedir. Birleşik Ağlar beynimizin en önemli fonksiyonlarından biri olan Birleşik Bellek kavramına dayanır. Bir ismin bir kişiyi, bir telefon numarasını, bir başka kişiyi çağrıştırması beynin Birleşik Bellek özelliğinden kaynaklanmaktadır. Birleşik Bellek kavramından hareketle bir grup yapay sinir ağı modelini kapsayan Birleşik Ağlar ortaya çıkmıştır. Birleşik Ağlar kabaca “ bir vektör setini başka bir vektör setine bağlayan ağ” olarak tanımlanabilir. Kesikli Hopfield ağı ve Grossberg ağı bu tür ağlardır¹²⁰.

Kesikli ve Sürekli Hopfield Ağlarının çalışma biçimleri aynıdır ancak kullanılan aktivasyon fonksiyonları farklıdır. Kesikli Hopfield Ağları aktivasyon fonksiyonu olarak signum ¹²¹ fonksiyonunu kullanırken Sürekli Hopfield Ağları sigmoid fonksiyonunu kullanmaktadır.

Bu ağdaki her elemanın iki değeri bulunur. İşlemci elemanın değeri +1 (on) veya -1 (off) olabilmektedir. Bir işlemci elemanın t . zamandaki girdisi $y_k(t)$ olarak gösterilirse:

$$x_k(t) = \sum_{j \neq k}^N w_{jk} y_j(t-1) - \theta_k \quad (3.52)$$

olur. Formülde kullanılan w , ağırlık değerini, $y(t-1)$ işlemci elemanın bir önceki zaman dilimindeki çıktı değerini, θ ise sabit eşik değerini göstermektedir. Aynı işlemci elemanın çıktısı $y(t)$ ise:

¹¹⁹ Öztemel,a.g.e.,s.171.

¹²⁰ Gülseçen,a.g.e.,s.77.

¹²¹ Signum fonksiyonu, tanımlanan değerın işaretine göre, -1, 0 ve +1 sonuçlarını veren bir fonksiyondur. Tanımlanacak değer 0'dan küçükse: -1, 0'a eşitse: 0 ve 0'dan büyükse: +1 sonucunu verir.

$$y(t) = \text{sgn}(x(t)) \quad (3.53)$$

formülü ile hesaplanır. Formüldeki sgn, signum fonksiyonunu belirtmektedir.

$$y_k(t) = \begin{cases} +1 & \text{Eger } x_k(t) > u_k \\ -1 & \text{Eger } x_k(t) < u_k \\ y_k(t-1) & \text{Aksi halde} \end{cases} \quad (3.54)$$

Burada kullanılan u değeri de bir eşik değeridir. Pratikte 0 değeri seçilmekle birlikte böyle bir zorunluluk bulunmamaktadır.

Hopfield ağınnın eğitilmesinde iki aşama vardır: ağırlıkların belirlenme ve saklama ile bilgilere ulaşma.

Ağın eğitimi, ağırlıkları belirleme aşamasında, bir defada öğrenme prensibine göre şöyle hesaplanır:

$$w_{ij} = \begin{cases} \frac{1}{N} \sum_{p=1}^M \chi_i^p \chi_j^p & \text{Eger } i \neq j \text{ ise} \\ 0 & \text{Aksi halde} \end{cases} \quad (3.55)$$

Bu formüldeki M öğrenilecek (saklanacak) örnek sayısını, χ ise bir örneğin i . ve j . elemanının değerini göstermektedir. Bu ağırlıklar hesaplandıktan sonra sabitlenir. Bu ağırlıklar hesaplandıktan sonra sabitlenir. Burada, oluşturulan ağırlık matrisi simetrik olmasından dolayı, w_{ij} ile w_{ji} ağırlıkları aynıdır.

Ağın kullanılabilmesi için kararlı (Stable) hale gelmiş olması gerekir. Bunun için ağa daha önce görmediği yani eğitim setinde olmayan bir örnek gösterilir. Bu örnek eksik bilgiler içerebilir. Ağın görevi bu eksiklikleri gidermek ve örneğin tamamını

hafızadan bulmaktır. Bunun için örnek ağa sunulmakta ve ağ durağan hale gelene kadar iterasyonlar yapılmaktadır. Ağ durağan hale geldikten sonra ürettiği çıktı, ağın kendisine gösterilen girdiye cevabı olarak görülür. Girdi örneği X ($X_1, X_2, X_3, \dots, X_N$) başlangıç değerlerine atanmak üzere ağın iterasyonları yukarıda da gösterilen formüle göre devam etmektedir:

$$y_k(t+1) = \text{sgn}\left(\sum_{j=1, j \neq k}^N w_{jk} y_j(t) - \theta_k\right) \quad (3.56)$$

Bu formülün çalışabilmesi için girdi vektörü, ağın başlangıç çıktı değerleri olarak atanmaktadır. Yani,

$$X(0) = X = (X_1, X_2, X_3 \dots \dots X_N)$$

olarak alınmaktadır. Ağın durağan hale gelebilmesi için enerji fonksiyonunun değeri en azlanır. Bu enerji fonksiyonu:

$$\mathcal{E}(t) = -\frac{1}{2} \sum_{i=1}^N \sum_{j=1, j \neq i}^N w_{ij} (y_i(t) y_j(t) + \sum_{i=1}^N y_i(t) \theta_i) \quad (3.57)$$

şeklinde verilmektedir. Ağ çalışırken bu enerji fonksiyonu ya azalır ya da değişmez. Dolayısıyla zaman içinde ağın minimum hata düzeyine ulaşması(durağan hale gelmesi) her durumda mümkün olmaktadır¹²².

Hopfield ağlarının ana kullanım amacı İlişkili Hafızadır (Associative Memory). Bunun yanında Hopfield ağları optimizasyon problemlerinde ve veri ilişkilendirmede yaygın olarak kullanılmaktadır. Hopfield ağlarının en önemli uygulamalarından biri,

¹²² Öztemel, a.g.e., s.173.

gezgin satıcı problemidir. Bu problemde bir satıcı N adet şehre gitmek zorundadır. Bir şehre bir defa uğramak koşulu ile en kısa zamanda bütün şehirleri gezebilmesi için izlemesi gereken rotanın bulunması istenmektedir¹²³.

3.8 Kohonen Ağları

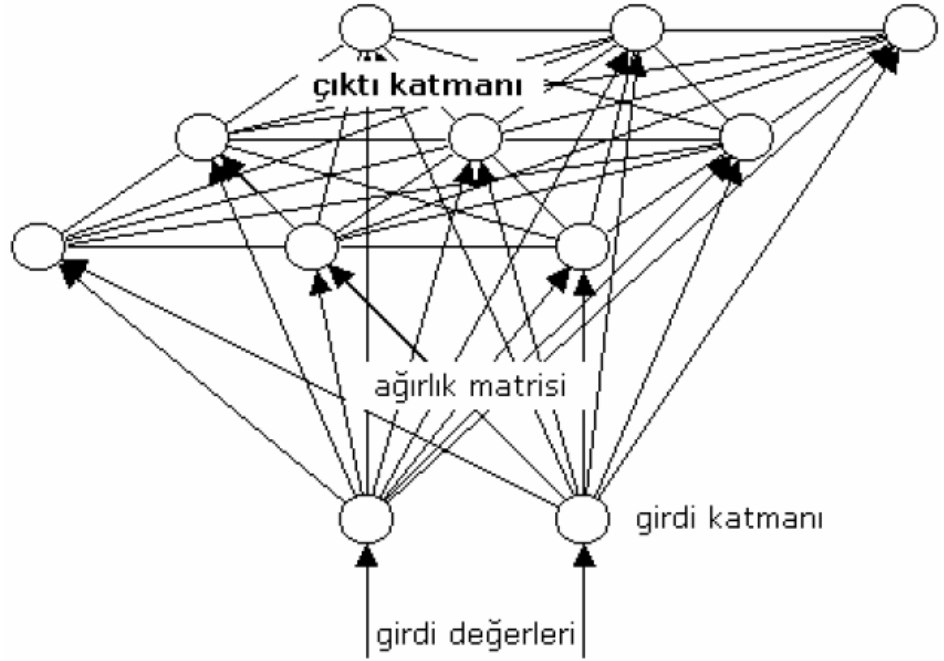
SOM (Self Organization Feature Map Network-Özörgütlemeli Özellik Haritası) olarak da adlandırılan Kohonen ağı, beynin Neocortex tabakasında yaygın olan duyuşal haritalardan esinlenilerek 1972 yılında Kohonen tarafından geliştirilmiştir¹²⁴. Genel olarak sınıflandırma yapmak için kullanılan Kohonen ağlarının, girdi vektörlerinin dağılımını öğrenebilme ve girdi vektörlerini sınıflandırma yetenekleri yüksektir. Bu ağların diğer ağlardan en önemli farkı, eğitim sırasında öğretmene ihtiyaç duymamalarıdır.

Yapısal olarak diğer ağlardan farklı olan bu ağ, girdi katmanı ile iki boyutlu çıktı katmanından oluşmaktadır. Çıktı katmanı iki boyutlu bir düzlemi, işlemci elemanlar ise bu düzlem üzerine dağılmış vektörleri gösterirler. Girdi elemanları çıktı elemanlarının tamamıyla bağlantılıdır. Yarışmacı öğrenmeyi kullanan Kohonen ağında kazanan işlemci eleman 1 diğer elemanlar ise 0 değerini alır. Bu stratejiye Kazanan Hepsini Alır (Winner-Takes-All) stratejisi denir. Eğitimi sırasında hem yarışmayı kazanan işlemci elemanın hem de komşusu olan işlemci elemanların ağırlıkları değiştirilir¹²⁵.

¹²³ Bayır,a.g.e.,s.69-70.

¹²⁴ Baş,a.g.e.,s.64.

¹²⁵ Öztemel,a.g.e.,s.174-175



Şekil 3.12: Kohonen Ağı¹²⁶

Bu ağın eğitimi sırasında, herhangi bir t zamanında örnek setinden bir örnek ağa gösterilir. Girdi vektörü X ve ağırlık vektörü W normalize edilmiş olmalıdır. Çıkı elemanlarından kazanan elemanı bulmak için iki yöntem bulunmaktadır.

Birinci yöntemle göre, her elemanın çıkı sı ağırlıklarla girdilerin çarpımının toplamı ile bulunur:

$$y_i = \sum_i W_{i0} x_i \quad (3.58)$$

Bu çıkı değlerinden en yüksek değere sahip olan işlemci eleman yarışmayı kazanır. Bu işlemci elemanın k. eleman olması durumunda:

$$y_k = 1$$

¹²⁶Fröhlich,a.g.e.

$$y_i = 0 \quad i = 1, 2, \dots \text{ ve } i \neq k$$

olur. İkinci yöntemde ise, Öklit mesafesi (d) kullanılarak elde edilen, girdi vektörüne en yakın ağırlık vektörüne sahip işlemci eleman kazanan elemandır. İki vektör arasındaki mesafe:

$$d_j = \|X - W_j\| \quad (3.59)$$

ile hesaplanır. Her çıktı elemanı için bu mesafeler hesaplanmakta ve en küçük mesafe değerine sahip işlemci eleman kazanan eleman olarak belirlenir. Kazanan işlemci elemanın belirlenmesinin ardından bu elemanın ve komşularının ağırlıkları şu şekilde hesaplanır:

$$W(t+1) = W(t) + \lambda g(i,k)(X(t) - W(t)) \quad (3.60)$$

Burada λ öğrenme katsayısıdır. Öğrenme sırasında değeri zamanla küçülür. $g(i,k)$, komşuluk fonksiyonudur ve i ile k elemanlarının komşuluklarını belirlemektedir. $i=k$ olduğunda, $g(i,k)=1$ olur. Bu fonksiyon zaman içinde azalan bir fonksiyondur. Genel olarak $g(i,k)$:

$$g(i,k) = (\exp(-\|d_i - d_k\|^2 / (2\sigma^2))) \quad (3.61)$$

şeklinde formüle edilir. Formülde, d_i ve d_k i . ve k . elemanlarının pozisyonunu gösteren vektörler, σ ise komşuluk alanının genişliğini gösteren ölçüdür. Bu genişlik zaman içerisinde azalmaktadır.

Kazanan işlemci elemanın, onunla birlikte ağırlıklarının değiştirileceği, komşularını belirlemek için iki yöntem bulunmaktadır. Bunların ilki, kazanan işlemci elemanın etrafındaki elemanları bir kare/dikdörtgen içine almak ve bunun içinde kalanların ağırlıklarını değiştirmektir. İkinci yöntem ise, kazanan işlemci elemanın etrafındaki

elemanlar bir çokgen içine alınarak içinde bulunan elemanların ağırlıklarının değiştirilmesidir.

Kohonen ağı, ses ve yazı tanıma, eş zamanlı çeviri gibi işlemlerde oldukça başarılıdır. Renk, desen ayrımlarını yapabilir ve görünen, hareket eden nesneleri ayrıştırabilirler. Bu özellikleri sayesinde robotik sistemlerde ve sensör uygulamalarında da kullanılmaktadırlar¹²⁷.

¹²⁷ Bayır,a.g.e.,s.73.

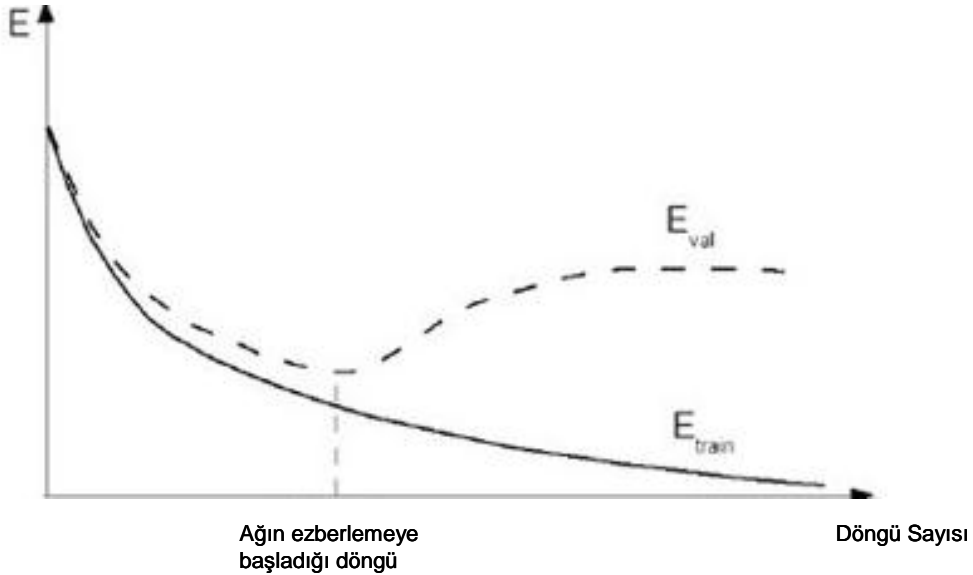
4. ÇOK KATMANLI YAPAY SİNİR AĞI TASARIMI

4.1 Eğitim ve Test Kümelerinin Seçimi:

Ağa sunulacak, eğitim ve test kümelerindeki örnekler ağın performansında çok etkilidir. Eğitim kümesindeki örnekler, ağın problem uzayını çok iyi yansıtmalıdır. Çünkü, ağ kendisine gösterilen örnekler üzerinden genelleme yapabileceğinden, test kümesinde, ağın eğitim kümesi ile ilişkili olmayan örnekler için ağın doğru kestirim yapmasını beklemek mümkün değildir.

Ayrıca, eğitim setindeki örnek sayısı ne kadar fazla olursa, fonksiyon üzerindeki bilinen nokta sayısı fazla olacağından, ağ doğruya daha yakın tahmin yapabilir.

Veri setinde, eğitim ve test kümesi yanında, bir de onaylama (cross validation) kümesi seçilebilir. Bu onaylama kümesi, modelin ezberlemeye başlamadan önce eğitimini durdurmak için kullanılır. Onaylama kümesi hatası artmaya başladığı an en iyi genel çözüm bulunduğu için eğitim durdurulur.



Şekil 4.1Eğitim Kümesi ve Onaylama Kümelerinin Hatalarının Karşılaştırılması

Veri setinin hangi oranlarda bölüneceğine dair kesin bir kural yoktur. Fakat genellikle eldeki verilerin %70'i eğitim, %20'si onaylama ve %10'u test için ayrılır¹²⁸.

4.2 Katman ve Nöron Sayısının Seçimi

YSA'da, ağın katman sayısı, ve katmanlardaki nöron sayısı için kesin bir kural, yoktur. En uygun yapı, deneme yanılma yöntemi ile bu bulunur. Genellikle, başlangıçta bir ağ yapısı seçilerek, zamanla ağ büyütülerek, ya da küçültülerek en uygun ağa ulaşılır. En uygun ağ yapısının bulunmasında, tasarımcının bu konudaki deneyimi önemli bir etkidir.

Katman sayısı ve katmanlardaki nöron sayısı arttıkça yapay sinir ağının işlem ve öğrenme yeteneği düşmekte ve ağın ezberlemesine (Memorization) neden olmaktadır. Gereğinden az kullanımı ise ağın örneklerden yeteri kadar iyi genelleme yapamayabilir.

Giriş katmanındaki nöron sayısı, bağımsız değişken sayısı kadardır. Ağın giriş katmanında her bir değişkene ait bir nöron mevcuttur. Çıkış katmanındaki nöron sayısı da, hedeflenen çıktı sayısı kadardır. Genellikle bir çıktı sonucu hedeflendiğinden, 1'dir.

Gizli katman sayısı ise probleme, veri setine bağlı olarak değişmektedir. Genellikle bir ya da iki ara katman (gizli katman) yeterlidir. Daha fazla ara katman kullanmak ağın hızını düşürür ve ağın ezberlemesine yol açabilir. Eğer 3 katmanlı bir yapı ile uygun sonuç bulunamazsa ara katman sayısı 4'e kadar çıkartılabilir. Daha önce yapılan çalışmalara göre, ara katman sayısını, 4'den fazla seçmek ağın performansını düşürmektedir.

Ara katmanlardaki nöron sayısı içinde kesin bir kural yoktur. Fakat daha önce yapılan çalışmalara göre bazı genellemeler yapılmıştır. İstenen ideal sonucu elde

¹²⁸ JingTaoYAO, Chew LimTAN, Guidelines for Financial Forecasting with Neural Networks, Nisan 2001

etmek için n girdili bir ağda, katman başına $2n$, $2n + 1$, $2n+2$ nöron sayısı ile ulaşılabilirliğini öne sürmektedir. Ayrıca, 3 katmanlı, n giriş nöronlu ve m çıkış nöronlu bir ağ için gizli katmandaki nöron sayısı karekök ($n \times m$) kadar olabilir gibi bir varsayım da vardır. Gizli katmandaki hücre sayısı problemin türüne ve veri sayısına göre yukarıdaki formülün 1,5 ila 2 katı arasında değişebilir. Buna geometrik piramit kuralı denmektedir. Baily ve Thompson 3 katmanlı bir ağ için gizli hücre sayısını giriş katmanındaki hücre sayısının %75'i olarak önermektedirler. Katz ise optimum sayının giriş katmanındaki hücre sayısının 1,5 ila 3 katı olarak kullanılmasını önermektedir¹²⁹.

4.3 Girdilerin Ölçeklendirilmesi

Veri seti oluşturulurken, değişkenler farklı ölçekler kullanılarak bir araya getirilmiş olunabilir. Ayrıca, veride çok büyük ya da çok küçük aşırı uç değerler olabilir. Verideki bu aşırı salınımları engellemek ve sistem performansını arttırmak için girdiler ölçeklendirilir. Bunun için logaritmik fonksiyonlar kullanıldığı gibi, genellikle verilerin $[0,1]$ veya $[-1,+1]$ aralıkları arasında normalize edilir. Ölçekleme verilerin geçerli eksen sisteminde sıkıştırılması anlamı taşıdığından veri kalitesi aşırı salınımlar içeren problemlerin YSA modellerini olumsuz yönde etkileyebilir¹³⁰.

4.4 Fonksiyon Seçimi

Aktivasyon fonksiyonu, hücrenin çıkışı hesapladığı fonksiyondur. Aktivasyon fonksiyonunun seçimi verilerin özelliklerine ve ağdan öğrenilmesi istenen verinin türüne ve yapısına bağlıdır. En çok kullanılan aktivasyon fonksiyonu ise sigmoid aktivasyon fonksiyonudur. Diğer en çok kullanılan fonksiyonlar ise, doğrusal, işaret, eşik, hiperbolik tanjant ve lojistik fonksiyonlarıdır. Doğrusal olmayan problemlerde doğrusal olmayan aktivasyon fonksiyonları kullanılmaktadır.

¹²⁹Iebling Kaastra, Milton Boyd: "Designing a Neural Network for Forecasting Financial and Economic Time Series", Neurocomputing,(215,236),1994,s.225.

¹³⁰Saraç,a.g.e,s.35-36.

4.5 Performans Fonksiyonunun Belirlenmesi

Performans fonksiyonları, istenilen çıktı değerleri ile ağın ürettiği değerler arasındaki farkın kümülatif değerlerini hesaplamaktadır. Bu performans değerleri ile, eğitim kümesinin gerçeği ne kadar temsil ettiği (hatayı minimize ettiği) gözlemlenmekte ve ağırlıklar bu değerlere göre değiştirilmektedir. YSA’da hatayı minimize etmek için en çok kullanılan yöntem hata karelerinin ortalaması (Mean Square Error (MSE))’dır. Ayrıca, ortalama hata (Mean Error (ME)), Hata Kareleri Ortalamasının Karekökü (Root Mean Square(RMSE)) ve Ortalama Mutlak Hata (Mean Absolute Error (MAE)) kullanılan diğer performans kriterleridir.

4.6 Öğrenme Oranı ve Momentum Katsayısının Belirlenmesi

Öğrenme oranı/katsayısı ağırlıkların değişim miktarını belirler. Öğrenme oranının büyük seçilmesi salınım (oscillation) yol açmakta ve bir sonuca ulaşmasını zorlaştırmaktadır. Küçük bir değer seçilmesi ise öğrenme süresinin uzamasına ve ağın yerel çözümlere takılmasına neden olmaktadır.

Bir problemin çözümünde, en az hatayı veren çözüm olan mutlak minimuma ulaşmak her zaman pratikte mümkün değildir. Yapay sinir ağı, eğitimi sırasında bu çözüme ulaşmaya çalışmakta fakat ağ farklı bir çözüme takılabilmekte ve daha fazla iyileştirmek mümkün olmamaktadır¹³¹.

Momentum katsayısı ise ağırlıktaki önceki değişimlerin geçerli ağırlık değişimini nasıl etkileyeceğini belirler. Eğer momentum terimi kullanılmazsa ağın düşük öğrenme katsayısı ile yerel minimuma ulaşması çok uzun sürmekte, büyük öğrenme katsayısında ise salınımdan dolayı yerel minimuma ulaşamamaktadır. Momentum kullanıldığında ise yerel minimuma hızlı bir şekilde ulaşılabilir.

¹³¹Mehrotra, Mohan ve Ranka,a.g.e.,s.82-83.

Özetle momentum katsayısı, özellikle yerel çözümlere takılan ağların bir sıçrama ile daha iyi sonuçlar bulmasını ve ağın daha hızlı sonuca ulaşmasını sağlamak amacıyla kullanılmaktadır. Bu değerin küçük olması yerel çözümlerden kurtulmayı, büyük bir değer olması ise tek bir çözüme ulaşmayı güçleştirebilmektedir.

4.7 Ağın Devir (Epoch) Sayısı

Eğitim setinin ağ içinde bir kez işleminden geçirilmesine devir (epoch) denir. Devir sayısı, ağın öğrenmesini doğrudan etkileyeceğinden, dikkatli seçilmesi gerekir. Eğitim sürecinde çok fazla ya da çok az iterasyon kullanımından sakınılmalıdır. Gerekenden az sayıda iterasyon, yetersiz bir eğitimle sonuçlanacaktır. Gerekenden fazla iterasyon ise ağın eğitimi için gerekli süreyi uzatacak ve aşırı uygunluk problemine (overfitting) de neden olarak ağın test seti üzerindeki performansını azaltacaktır.

5. LOJİSTİK REGRESYON

Lojistik regresyon (LR) analizi, bağımlı değişkenin kategorik olduğu durumlarda kullanılan regresyon modelidir. LR analizinin temel amacı diğer regresyon yöntemlerinde olduğu gibi bağımsız değişkenler ile bağımlı değişken arasındaki ilişkiyi incelemektir. Başka bir deyişle amaç minimum uygun sayıda değişken ile sonuç değişkeni ve açıklayıcı değişkenler arasındaki ilişkiyi tanımlayan kabul edilebilir modeli kurmaktır. Lojistik regresyon yönteminde bağımlı değişkenin sürekli olması gibi bir varsayım yoktur, özellikle bağımlı değişkenin iki veya daha çok değer aldığı durumlarda kullanılır¹³².

Lojistik modelin biyolojik deneylerin analizi için kullanımı ilk olarak Berkson (1944) tarafından önerilmiş, Cox (1970) bu modeli gözden geçirerek çeşitli uygulamalarını yapmış, özet gelişmeler ise ilk Andersson (1979, 1983) tarafından verilmiştir. Ayrıca verilerin lojistik modele uyumu ile ilgili birçok çalışmalar da yapılmıştır. Bunlar arasında Aranda-Ordaz (1981) ve Johnson(1985) tarafından yapılan çalışmalar en önemlileridir. Pregibon (1981) iki grup lojistik modelde etkin (influential), aykırı (outlier) gözlemleri ve belirleme ölçütlerini (diagnostic), Lesaffre (1986), Lesaffre ve Albert (1989) ise çoklu grup lojistik modellerde etkin ve aykırı gözlemlerle belirleme ölçütlerini incelemişlerdir.

Lojistik regresyon modellerinin yaygın bir şekilde kullanılır hale gelmesi, katsayı tahmin yöntemlerinin geliştirilmesi ve lojistik regresyon modellerinin daha ayrıntılı incelenmesine sebep olmuştur. Cornfield (1962), lojistik regresyondaki katsayı tahmin işlemlerinde diskriminant fonksiyonu yaklaşımını ilk kez kullanarak popüler hale getirmiştir. Lee (1984) basit dönüşümlü (cross-over) deneme planları için liner lojistik modeller üzerinde durmuştur. Bonney (1987) lojistik regresyon modelinin kullanımı ve geliştirilmesi üzerinde çalışmıştır. Robert ve ark. (1987) lojistik regresyonda standart Kikare, olabilirlik oran (G2), “pseudo” en çok olabilirlik

¹³²Sultan Deniz Ulupınar, 2001 Kriz Dönemi, Öncesi ve Sonrasında Türk TicariBankalarının Karlılıklarının Lojistik Regresyon Analizi ile İncelenmesi, İstatistik BilimDalı Yüksek Lisans Tezi, Marmara Üniversitesi, İstanbul, 2007, s.39.

tahminleri, uyum mükemmelliği ve hipotez testleri üzerine çalışmalar yapmışlardır. Duffy (1990) lojistik regresyonda hata terimlerinin dağılışı ve parametre değerlerinin gerçek değerlere yaklaşımını incelemiştir. Başarır (1990) klinik verilerde çok değişkenli lojistik regresyon analizi ve ayırsama sorunu üzerinde çalışmıştır. Hsu ve Leonard (1995) lojistik regresyon fonksiyonlarında Bayes tahminlerinin elde edilmesi işlemleri üzerine çalışmışlar ve lojistik regresyonda Monte Carlo dönüşümünün kullanılabileceğini göstermişlerdir. Akkaya ve Pazarlıoğlu (1998) lojistik regresyon modellerinin ekonomi alanında kullanımını örneklerle incelemiştir. Cox ve ark. (1998) kardiovasküler hastalıklar ve hipertansiyon arasındaki ilişkiyi incelemiştir¹³³.

LR, ilk zamanlarında sadece sağlık bilimleri çalışmalarında kullanılsa da, son yıllarda işletme alanında da sıklıkla kullanılmaktadır. Son dönemlerde bazı uygulamalarda, kredi değerlendirme kuruluşları kendilerine gelen başvuruların kredilendirme olasılığını lojistik regresyon modelini kullanarak belirlemektedir¹³⁴.

LR ile Doğrusal Regresyon arasındaki en önemli fark, bağımlı değişkenin kategorik olmasıdır. Ayrıca, Doğrusal Regresyonda, bağımsız değişken(ler)in, normal dağılıma şartı aranırken, lojistik regresyonda böyle bir koşul yoktur ve Doğrusal Regresyonda, bağımlı değişkenin değeri tahmin edilirken, LR’de bağımlı değişkenin alabileceği değerlerden birinin gerçekleşme olasılığı tahmin edilir.

LR Analizi, değişkenler arasında çoklu doğrusal bağlantı olmadığını varsayar. Gözlem sayısının az olması çoklu bağlantı olasılığını artırır. Eğer çoklu doğrusal bağlantı problemi varsa katsayılar yanlış tahmin edilebilir, katsayıların standart hataları artabilir, t testi geçersiz ve modelin tahmin gücü düşük olabilir.

¹³³ Hüdaverdi Bircan, "Lojistik Regresyon Analizi: Tıp Verileri Üzerine Bir Uygulama" Kocaeli Üniversitesi Sosyal Bilimler Enstitüsü, 2004/2: s. 185-208, 2004, s. 186.

¹³⁴ Alan Agresti, Categorical Data Analysis, A John Wiley & Sons, Inc, Publication, Second Edition, 2002, s. 165.

Lojistik regresyon analizinde 3 yöntem vardır¹³⁵. Bu modellerden İkili (Binary) Lojistik Regresyon Modeli, kategorik bağımlı değişkenin iki cevap içerdiği (örn: var-yok) durumlarda kullanılmaktadır. Sıralı (Ordinal) Lojistik Regresyon, bağımlı değişkenin sıralı ölçekli olduğu, en az üç kategoride gözlenen değer içerdiği (örn: hafif, orta, ağır) durumlarda kullanılan bir yöntemdir. Nominal Lojistik Regresyon, cevap değişkeninin isimsel olduğu durumlarda (örn: tıp, eczacılık, mühendislik) uygulanan bir yöntemdir. Bu yöntem, Ordinal Lojistik Regresyon yöntemine benzer ancak burada bağımlı değişkenin aldığı cevapların sıralı olması şart değildir. Bu çalışmada, İkili Lojistik Regresyon analizi yapılmış ve bu yöntem açıklanmıştır.

5.1. İkili Lojistik Regresyon Modeli

Lojistik regresyon modeli şöyle yazılmaktadır:

$$E(y_i) = L_i = \log\left(\frac{P_i}{1-P_i}\right) = \sum_{k=0}^p \beta_k x_{ik} \quad (5.1)$$

Bu modelde sonuç değişkeninin iki değer alması nedeni ile hata terimi sıfır ortalama ve $P(1-P)$ varyanslıdır. Hata terimi bu parametrelerle binom dağılımlı olup, analiz bu teorik temele dayanmaktadır. Lojistik fonksiyon aynı zamanda:

$$P_i = \frac{1}{1+e^{-(\alpha+\beta x)}} \quad (5.2)$$

ile de gösterilir. Bu eşitliğe lojistik dağılım fonksiyonu adı verilir. $\alpha+\beta x=Z$ olarak kabul edilirse budurumda,

$$\frac{P_i}{1-P_i} = \frac{1+e^{Z_i}}{1+e^{-Z_i}} = e^{Z_i} \quad (5.3)$$

¹³⁵ Kazım Özdemir, Paket Programlar ile İstatistiksel Veri Analizi-1, Eskişehir, Kaan Kitabevi, 2004, s.591

Eşitliği elde edilir. Bu eşitlik odds (bahis) oranı olarak adlandırılır¹³⁶. Bir olayın gerçekleşme olasılığı p ise, gerçekleşmeme olasılığı $1-p$ ’ dir. Olayın oddsu yani bir olayın gerçekleşme ve gerçekleşmeme olasılıklarının oranı “ O ” olarak tanımlanır:

$$O = \frac{p}{1-p} = \frac{\text{olayın gerçekleşme olasılığı}}{\text{olayın gerçekleşmeme olasılığı}} \quad (5.4)$$

Odds oranının 1’ den küçük olması, olasılığın 0,5’ den küçük olmasına; oddsun 1’ den büyük olması ise olasılığın 0,5’ den büyük olmasına işaret eder. Odds oranının en alt sınırı olasılıkta olduğu gibi sıfırdır, fakat odds oranında olasılıktaki gibi bir üst sınır bulunmamaktadır¹³⁷.

Odds değeri $-\infty$ ile $+\infty$ arasında değişen bir değerdir. Odds’ un doğal logaritması alındığında, diğer bir deyişle logiti elde edildiğinde; odds, 1’ den 0’ a yaklaştıkça mutlak değer olarak giderek artan ve pozitif yönde 1’ den sonsuza yaklaştıkça da yine giderek artan bir değer olur. Eğer y bağımsız değişkeni için odds değerinin doğal logaritması kullanılırsa, tahminlenen olasılığın, minimum ve maksimum değerleri aşması gibi bir problem kalmayacaktır¹³⁸.

İki düzeyli bağımlı değişkenler için lojistik modelin katsayılarını tahmin etmek amacıyla kullanılan yöntemler, En Çok Olabilirlik(Maximum Likelihood, ML), Yeniden Ağırlıklandırılmış İteratif En Küçük Kareler (Reweighted Iterative Least Square, RILS) ve Minimum Lojit Ki – Kare(Minimum Logit Chi-Square, MLCS) yöntemleridir¹³⁹. Bu tez çalışmasında En Çok Olabilirlik yöntemi kullanılmıştır.

¹³⁶ Serhat Burmaoğlu, Birleşmiş Milletler Kalkınma Programı Beşeri Kalkınma Endeksi Verilerini Kullanarak Diskriminant Analizi, Lojistik Regresyon Analizi ve Yapay Sinir Ağlarının Sınıflandırma Başarılarının Değerlendirilmesi, Atatürk Üniversitesi Sosyal Bilimler Enstitüsü İşletme Anabilim Dalı Doktora Tezi, 2009.

¹³⁷ Allison, P. D., Logistic Regression Using the SAS System: Theory and Application, SAS Institute Inc, USA, 2001, s.12.

¹³⁸ Menard, a.g.e., s.13.

¹³⁹ Hüseyin Tatlıdıl, Uygulamalı Çok Değişkenli İstatistiksel Analiz, Ankara, Akademi Matbaası, 1996.

Genel olarak en çok olabilirlik yöntemi, gözlenen veri kümesini elde etmenin olasılığını maksimum yapan bilinmeyen parametrelerin değerlerini verir. Bu metodu uygulamak için öncelikle, en çok olabilirlik fonksiyonunun oluşturulması gerekmektedir. Bu fonksiyon gözlenen verilerin olasılıklarını, bilinmeyen parametrelerin bir fonksiyonu olarak açıklar. Bu parametrelerin en çok olabilirlik tahmin edicileri, fonksiyonu maksimum yapan değerleri bulacak şekilde seçilir. Böylece sonuçta elde edilen tahminleyiciler, gözlenen verilerle çok yakın değerlere sahiptir¹⁴⁰.

İki değeri olan bir bağımlı değişken için başarı olasılığı $p_i = p(y_i = 1/X)$ ve başarısızlık olasılığı $1 - p_i$ olmak üzere, her bir gözlemin p_i parametresi ile Bernoulli dağılımına sahip olduğu varsayılırsa $i = 1, 2, \dots, n$ için olasılık, $P(Y_i/X_i) = P_i^{y_i}(1 - P_i)^{1-y_i}$ şeklinde yazılabilir. Aynı denklem n gözlem için yazılacak olursa,

$$P\left(\frac{Y}{X}\right) = P_i^{y_i}(1 - P_i)^{1-y_i} \text{ olarak yazılır. } n \text{ gözlem için ise,}$$

$$L\left(\frac{Y}{X}\right) = P\left(\frac{Y}{X}\right) = \prod_1^n P_i^{y_i}(1 - P_i)^{1-y_i} \quad (5.5)$$

Bu denklem, en çok olabilirlik fonksiyon denklemidir.

En çok olabilirlik yönteminde, p tane bağımsız değişkene ait β tahmini, bağımlı değişkeni gözleme olasılığını yani $L(Y/X)$ olabilirlik fonksiyonunu maksimum yapacak şekilde seçilir. Lojistik regresyon modelinin en çok olabilirlik fonksiyonu, $L(Y/X)$ denkleminde P_i yerine açık ifadesi konularak elde edilir. Buradan, olabilirlik fonksiyonunun logaritması alınır:

$$\ln[L(Y/X, \beta)] = \sum_1^n [y_i \ln P_i + (1 - y_i) \ln(1 - P_i)] \quad (5.6)$$

Denklemini elde edilir.

Denklem 5.2'de β 'ya göre birinci türevi alınıp sıfıra eşitlenirse:

¹⁴⁰ Bircan, a.g.e., s.191.

$$\sum_1^n (y_i - P_i) x_{ij} = 0 \quad (5.7)$$

Bu eşitlikler olabilirlik eşitlikleri (likelihood equations) olarak adlandırılır. Bu denklemin çözülmesi ile β 'nın tahmin değeri olan $\hat{\beta}$ 'ya ulaşılır. P_i nin üstel olması nedeniyle 5.3 denkleminin doğrusal değildir. Bu nedenle, $\hat{\beta}$ 'yı bulmak için bu denklem iteratif olarak çözümlenmelidir. Bu çözümde, $\hat{\beta}$ lara bir başlangıç değeri verilerek ilk tahminleri elde edilir. Her adımda ε kadar küçük miktarda ayarlamalar yapılarak türevler alınır ve en çok olabilirlik tahminleri bulunur. İterasyon yakınsama sağlanıncaya dek devam eder. Yakınsama, ε düzeltme terimlerinin iterasyon değerlerini değiştirmedeği noktada gerçekleşir.

5.2. Modelin Katsayılarının Testi ve Yorumlanması

5.2.1. Olabilirlik Oran Testi

Lojistik regresyonda, bağımsız değişkenin modelde olduğu durumdaki ve modelde olmadığı durumdaki tahmin edilen değerler kıyaslanır. Lojistik regresyonda bu karşılaştırma log en çok olabilirlik fonksiyonu ile yapılır¹⁴¹.

$$D = -2 \ln \left[\frac{\text{Şu andaki modelin olabilirliği}}{\text{Doymuş modelin olabilirliği}} \right] \quad (5.8)$$

$(-2 \ln)$ katının alınması, matematiksel olduğu kadar dağılımı bilinen bir değer elde etmektir. Bu değer hipotez testi amacıyla kullanılmaktadır. Böyle bir teste olabilirlik oran testi adı verilmektedir. Bu denklem, açık olarak yazılacak olursa,

$$D = -2 \sum_{i=1}^n \left\{ y_i \ln \left(\frac{\hat{\pi}_i}{y_i} \right) + (1 - y_i) \ln \left(\frac{1 - \hat{\pi}_i}{1 - y_i} \right) \right\} \quad (5.9)$$

¹⁴¹Menard,S.,Applied Logistic Regression Analysis, Sage Publications, California, 2002.

Elde edilir. Bağımsız bir değişkenin önemine karar vermek için denklemde bağımsız değişkenin olduğu ve olmadığı durumlardaki D değerleri karşılaştırılır. Bağımsız değişkeni kapsamamasından dolayı ortaya çıkan D'deki değişim aşağıdaki gibidir:

$$G = D(\text{Değişkensiz Model için}) - D(\text{Değişkenli Model için}) \quad (5.10)$$

G, doğrusal regresyonda kullanılan F testindeki pay kısmı ile aynı rolü üstlenir. G'yi hesaplamak için farkı alınacak D değerlerinin her ikisi için de doymuş modelin olabilirlikleri ortak olduğundan G istatistiği:

$$G = -2 \ln \left[\frac{\text{Değişkensiz modelin olabilirliği}}{\text{Değişkenli modelin olabilirliği}} \right] \quad (5.11)$$

Formülü ile hesaplanır. Tek bağımsız değişkenli özel durumlarda, değişkenin modelde olmadığı durumdaki β_0 'ın en çok olabilirlik tahmini $\ln(n_1/n_0)$ 'dır. $n_1 = \sum y_i$ ve $n_0 = \sum (1 - y_i)$ olmak üzere sabit terimin tahmin edilen değeri n_1/n 'dir. G istatistiği şöyle hesaplanır:

$$G = 2 \{ \sum_{i=1}^n [y_i \ln(\hat{\pi}_i) + (1 - y_i) \ln(\hat{\pi}_i)] - [n_1 \ln(n_1) + n_0 \ln(n_0) - n \ln(n)] \} \quad (5.12)$$

$\beta_1 = 0$ hipotezi ile, G istatistiği 1 serbestlik dereceli χ^2 dağılımı gösterir. Tüm değişkenleri içeren model ile kestirilen modele ilişkin olabilirlik oran değerlerinin farkına dayanan ölçütlerin Ki-kare dağılacağı düşüncesinden hareketle kurulan modelin geçerliliği sınanmaktadır¹⁴².

5.2.2. Wald Testi

Wald istatistiği β parametresi ile standart hatasının oranıdır ve Z dağılımı göstermektedir. Doğrusal regresyondaki t testinin alternatifidir. Wald χ^2 istatistiği t değerlerinin karesine eşittir.

¹⁴² Bircan, a.g.e., s.16.

$$w = \frac{\hat{\beta}_i}{SE(\hat{\beta}_i)} \quad (5.13)$$

Wald istatistiği standart normal bir değişkendir. Karesi 1 serbestlik derecesi ile χ^2 dağılır. Bu durumda:

$$w = \left[\frac{\hat{\beta}_i}{SE(\hat{\beta}_i)} \right]^2 \quad (5.14)$$

İstatistiği tanımlanır. Büyük β değerleri için tahmin edilen standart hatalar da büyük çıkmaktadır. Bu durum, H_0 hipotezi yanlış iken kabul edilmesi olasılığını artırmaktadır. Modelin katsayılarının yorumlanması için lojistik regresyon modelinde aşağıdaki eşitlikten yararlanılır.

$$\ln \left(\frac{P_i}{1-P_i} \right) = d = X' \beta \quad (5.15)$$

$\left(\frac{P_i}{1-P_i} \right)$ 'nin logaritması modelin katsayılarının doğrusal bir şekilde yorumlanabilmesini sağlamaktadır. Bu şekilde açıklayıcı değişkendeki 1 birim değişimin olasılık üzerindeki etkisi görülmektedir¹⁴³.

Wald istatistiği, lojistik regresyon katsayısı mutlak değer olarak büyüdükçe tahmin edilen standart hatalar anormal bir biçimde büyümektedir. Bu durum, Wald istatistiğinin küçük çıkmasına ve gerçekte sıfır hipotezi reddedilmesi gerekirken kabul edilmesine yol açmaktadır. Bu yüzden katsayılar mutlak olarak büyüdükçe hipotez testlerinin sınanmasında Wald istatistiğinin kullanılması önerilmemektedir. Bunun yerine ilgili değişken modele eklenerek veya çıkartılarak -2LL değerindeki değişimler değerlendirilerek test yapılabilir¹⁴⁴.

¹⁴³ Ulucan, a.g.e., s.49.

¹⁴⁴ Ali Sait Albayrak, **Uygulamalı Çok Değişkenli İstatistik Teknikleri**, Asil Yayın Dağıtım, 2006, s.452.

5.3. Modelin Uyum İyiliğinin Ölçülmesi

Kurulan modelin uyum iyiliği testi Hosmer-Lemeshow'un hem onlu risk grupları hem de sabit kesim noktası yöntemine göre hesaplanmaktadır¹⁴⁵. Hosmer ve Lemeshow gözlenen verilerin beklenen verilere uyum iyiliğini test etmek amacıyla 7 adet istatistik geliştirmiştir. Ancak bunlardan sadece ikisi kullanıldığından, C ve H ile ifade edilen bu iki istatistiğe yer verilecektir. C istatistiği verileri tahmin edilen olasılıklara göre ayırırken, H istatistiği verileri belirlenen bir kesim noktasına (cut-off point) göre ayırır. Gruplara ayrılan verilere, daha sonra ki-kare uyum iyiliği testi uygulanmaktadır. Dolayısıyla her grup için beklenen ile gözlenen değerlerin bir karşılaştırması yapılmaktadır. Hesaplanan p değeri 0,05'den büyük ise gözlenen ve beklenen değerler arasında fark olmadığını gösteren sıfır hipotezi reddedilir. Bu durum modelin mevcut veriye olan uyumunun iyi olduğunu gösterir. Bu testte aşağıdaki istatistik kullanılmaktadır¹⁴⁶:

$$G^2_{HL} = \sum_{j=1}^{10} \frac{(O_j - E_j)^2}{E_j(1 - E_j/n_j)} \quad (5.16)$$

Belirtilen istatistikle eşit sayıda tahmin edilen olasılıklara göre 10 grup oluşturulmakta ve bu dağılımın Ki-kare dağılımına uyduğu düşünülmektedir.

Modelin uyum iyiliğinin belirlenmesinde R^2 istatistiklerinden de yararlanılmaktadır. LR'deki R^2 , doğrusal regresyon analizindeki R^2 'den farklıdır. Bu değer doğrusal regresyonda, bağımlı değişkenin açıklanan varyansının yüzdesini göstermekte, ancak LR'de ise bağımlı değişkenin varyansı bu değişkenin olasılık dağılımına bağlıdır¹⁴⁷. Bu nedenle bu iki R^2 değerini karşılaştırmak uygun değildir.

Modelin uyum iyiliğini belirlemede kullanılan istatistiklerden biri McFadden'in R^2 uyum iyiliği kriteridir. $-2\ln L_0$ logitmodelde yer alan sabiti ve $-2\ln L_1$ modelin tüm

¹⁴⁵ Bircan, a.g.e., s.197.

¹⁴⁶ Burmaoğlu, a.g.e., s.62.

¹⁴⁷ Şeref Kalaycı, SPSS Uygulamalı Çok Değişkenli İstatistik Teknikleri, Asil Yayın, 2006, s.293.

açıklayıcı değişkenlerini içermek kaydıyla McFadden R^2 formülü şu şekilde hesaplanır:

$$\text{McFadden} R^2 = 1 - \frac{-2\ln L_1}{-2\ln L_0} = 1 - \frac{\ln L_1}{\ln L_0} \quad (5.17)$$

Bir başka R^2 istatistiği ise Cox ve Snell R^2 istatistiğidir. Cox ve Snell R^2 ;

$$\text{Cox - Snell } R^2 = 1 - \left[\frac{L_0}{L} \right]^{\frac{2}{I}} \quad (5.18)$$

formülü ile gösterilir ve I gözlem sayısını göstermektedir. Formülden de anlaşılacağı üzere R^2 'nin en yüksek değerinin bile birden küçük çıkması diğer bir deyişle maksimum değerinin hiçbir zaman 1'e ulaşamaması sonucun yorumlanmasında sorun olmaktadır¹⁴⁸.

Diğer bir test istatistiği ise Nagelkerke R^2 istatistiğidir. Nagelkerke R^2 ;

$$\text{Nagelkerke} R^2 = \frac{\text{Cox - Snell } R^2}{1 - \left[\frac{L_0}{L} \right]^{\frac{2}{I}}} \quad (5.19)$$

Bu ölçümün maksimum değeri 1 olabilir ve doğrusal regresyondaki belirtme katsayısı gibi yorumlanabilir. Nagelkerke R^2 , Cox-Snell R^2 'den her zaman büyük ancak doğrusal regresyondaki R^2 değerinden genellikle küçük olmaktadır.

¹⁴⁸ Burmaoğlu, a.g.e., s.63.

6. UYGULAMA

6.1. Veri Seti

Çalışmada kullanılan veri seti, özel bir firmanın kredi risk bölümünden temin edilmiştir. Kredinin konusu binek ve hafif ticari araç kredisi kullanmak isteyen firmalardır. Veri seti 2472 başvuru ve 17 bağımsız değişkenden oluşmaktadır. Hedef ise, firmaların taleplerinin olumlu ya da olumsuz sonuçlanmasıdır. Bağımsız değişkenlerin, 2'ser tanesi nominal ve oran ölçekli, 13 tanesi ordinal ölçeklidir. Verilerin büyük bir kısmı ordinal ölçeklidir. Veri setindeki bağımsız değişkenler şöyledir:

- a. **Araç Tipi:** Krediye konu olan aracın binek mi yoksa hafif ticari araç için mi olduğunu belirten nominal değişkendir.
- b. **Para Birimi:** İstenen kredinin para birimini belirten değişkendir. TL ve Euro gözlemlerinden oluşur, nominal ölçeklidir.
- c. **Peşinat Oranı:** Firmanın, istediği kredi için verdiği peşinatın, istediği kredi miktarına oranıdır. Oran ölçeklidir.
- d. **Taksit Sayısı:** Firmanın, istenilen krediyi ne kadar vade ile kullanmak istediğini gösteren oran ölçekli değişkendir.
- e. **İş Deneyimi:** Firmaların, yeni olmalarından, beş yıldan fazla iş deneyimine sahip olmalarına göre 0 ile 10 arasında puan veren ordinal ölçekli değişkendir.
- f. **Sektör deneyimi:** Firmanın sektördeki deneyimine göre verilen puan değeridir. Firmanın sektördeki deneyimine göre 0 ile 5 arasında değer veren ordinal ölçekli değişkendir.
- g. **Mekan Sahipliği:** Firmanın, bulunduğu iş yerinin sahipliği veya kirada olması gibi farklı durumlara göre sınıflandırıp 0-12 arasında puanlandıran ordinal ölçekli değişkendir.
- h. **Moralite:** Kredi talebinde bulunan firmanın veya ortaklarının, karşılıksız çek-senet kayıtları, kredi kartları ödeme performanslarını ve firma hakkındaki

olumsuz referanslara göre çeşitli sınıflara ayırarak, puanlandıran ordinal ölçekli değişkendir. 0 ile 20 arasında değer alır.

- i. **Ödeme Geçmişi:** Firmanın, daha önce kullandığı kredilerin % 85 ve fazlasını zamanında ödemesine 10 puan verip, diğer durumlara 0 puan veren ordinal ölçekli değişkendir.
- j. **Telefon Değeri:** Kredi kullanacak firmanın verdiği telefon sayısına göre puan atayan ordinal değişkendir. Kredi kullanan firmanın ve kişilerin güvenilirliğini gösterir. 0-3 arasında puan alır ve ordinal ölçekli değişkendir.
- k. **Borçluluk Oranı:** Firmanın borçlarının gelirin oranına göre 0-20 arasında değerler alan ordinal ölçekli değişkendir.
- l. **Cari Oran:** Şirketin kısa vadeli borçlarını ödeyebilme gücünü ve net işletme sermayesinin yeterli olup olmadığını gösterir. Oranın yüksek olması şirketin likidite durumunun kuvvetli olduğunu gösterir. Burada, firmanın cari oranına göre 0-5 arasında değerler alır, ordinal ölçeklidir.
- m. **Pay Oranı:** Firmalar kredi talebinde bulunduğunda, finans şirketi, şirketin ortaklarından birinin krediye kefil olmasını ister. Bu kefil olan kişi veya kişilerin firmaya olan ortaklık oranlarına göre 0-5 arasında değişen puanlar verilir. Ordinal ölçekli değişkendir.
- n. **Para Birimi Uyum:** Firmanın kazancının para birimi ile istediği kredi para birimi arasındaki uyuma göre gözlemlere 0 ile -5 arasında değer veren ordinal ölçekli değişkendir.
- o. **Belgelenmiş Mal Varlığı:** Kredi talebinde bulunan firmaların, belgelenmiş mal varlıklarının, istedikleri kredi miktarına göre, sınıflandırarak 0 ile 15 arasında değer alan ordinal ölçekli değişkendir.
- p. **Krediyi Değerlendiren Görüşü:** Krediyi değerlendiren kişinin kredi verilerini inceledikten sonra, görüşünün olumlu-olumsuz oluşuna göre 10 ile -10 arasında puan alan, ordinal ölçekli değişkendir.
- q. **Bayii Görüşü:** Firmaların kredi talebini ilk alıp, kredi merkezine yönlendiren kişinin, firma hakkındaki olumlu-olumsuz görüşüne göre 0 ile 3 arasında puan alan, ordinal ölçekli değişkendir.

Aşağıdaki tabloda, bağımsız ve bağımlı değişkenlerin betimsel istatistikleri verilmiştir.

	N	Boş Değer	Ortalama	Medyan	Mod	Standart Sapma	Aralık	Minimum	Maksimum
Arac_Tipi	2472	0	1,43	1,00	1,00	0,50	1,00	1,00	2,00
Currencyno	2472	0	1,32	1,00	1,00	0,47	1,00	1,00	2,00
PesinatOrani	2472	0	37,60	36,03	25,00	15,01	86,39	0,00	86,39
Taksit	2472	0	24,51	24,00	36,00	12,35	55,00	5,00	60,00
ToplamPuan	2472	0	51,15	54,00	27,00	18,07	101,00	2,00	103,00
İsDeneyimi	2472	0	6,30	10,00	10,00	4,17	10,00	0,00	10,00
SektorDeneyimi	2472	0	2,98	3,00	5,00	2,19	5,00	0,00	5,00
MekanSahipligi	2472	0	0,56	0,00	0,00	2,32	12,00	0,00	12,00
Moralite	2472	0	18,94	20,00	20,00	2,73	20,00	0,00	20,00
OdemeGecmisi	2472	0	0,39	0,00	0,00	1,93	10,00	0,00	10,00
Telefon	2472	0	2,02	2,00	2,00	0,35	3,00	0,00	3,00
BorçlulukOranı	2472	0	10,68	12,00	20,00	8,56	20,00	0,00	20,00
CariOran	2472	0	1,59	0,00	0,00	2,14	5,00	0,00	5,00
PayOrani	2472	0	0,65	0,00	0,00	1,45	5,00	0,00	5,00
ParaBirimiUyumu	2472	0	-1,21	0,00	0,00	2,12	5,00	-5,00	0,00
BelgelenmişMalVarligi	2472	0	3,01	0,00	0,00	4,08	15,00	0,00	15,00
KrediDegerlendirenGorusu	2472	0	4,47	5,00	5,00	2,43	20,00	-10,00	10,00
BayiiGorusu	2472	0	0,78	0,00	0,00	1,07	3,00	0,00	3,00
Karar	2472	0	0,69	1,00	1,00	0,46	1,00	0,00	1,00

Tablo 6.1:Değişkenlerin Betimsel İstatistikleri

Analizlere başlamadan önce, bağımsız değişkenler arasındaki ilişkili değerlerin yani çoklu doğrusal bağlantılı değişkenlerin analizden çıkartılması gerekmektedir. Çoklu doğrusallık, modellerin, tahmin gücünün azalmasına neden olmaktadır. Çoklu doğrusallık sorunun tespiti için bağımsız değişkenler arasındaki korelasyon katsayılarına bakılmıştır.

Analizde kullanılacak olan bağımsız değişkenlerden, ordinal ve nominal ölçekli olanları spearman sıra korelasyon testi, oran ölçekliler ile ordinal ölçekli değişkenlerin arasındaki korelasyonu hesaplamak için ise Kendal Tau b korelasyon testi yapılmıştır¹⁴⁹.Aralarındaki ilişki %70 ve daha fazla olan değişkenler, çoklu doğrusal bağlantılı olarak modelden çıkartılmıştır.

¹⁴⁹ Ek-1

Yapılan analizler sonucunda, Para Birimi Uyumu değişkeni ile, kredinin para birimi değişkeni arasında ve iş ve sektör deneyimleri arasında korelasyon olduğu sonucuna ulaşılmıştır. Bu değişkenlerden, para birimi ve sektör deneyimi değişkenleri atılmıştır. Analizler 15 bağımsız değişkenle yapılmıştır.

6.2. Yapay Sinir Ağları Analiz Sonuçları

Yapay sinir ağı analizi uygulaması, Neurosolutions 5.0 programı kullanılarak çok katmanlı Geri Yayılım Ağları (GYA) ile yapılmıştır. Veri seti, 2472 kredi başvurusu ve 15 bağımsız değişkenden oluşmaktadır. Uygulamadaki amaç, gelen kredi taleplerinin değerlendirilerek doğru bir şekilde analiz edilmesi ve kredinin evet/hayır şeklinde sınıflandırılmasıdır. Daha önce de belirtildiği gibi YSA'da kesin ve tek bir çözüm yoktur. Uygulamadaki amaç, en az hata ile en iyi sonuca ulaşmaktır. Bu nedenle değişik modeller denenerek en uygun ağ yapısı bulunmaya çalışılmıştır.

Uygulamadaki veriler özel bir finans kuruluşunun araba kredileri kısmından elde edilmiştir. Elimizdeki veride 2472 kredi başvurusu ve 15 bağımsız değişken vardır. Kredi talepleri ise evet (1) ve hayır (0) olmak üzere kategorik değişkenlerdir. Kredi taleplerinin %31'i olumsuz, % 69'u ise olumludur.

Kredi taleplerinin doğru bir şekilde tahmin edilebilmesi için en uygun mimariyi belirlemek gerekir. Bu nedenle ağın eğitim ve test kümeleri, katman sayısı, katmanlardaki nöron sayısı, aktivasyon fonksiyonu, öğrenme kuralı önemli parametrelerdir.

Uygulamada, yapay sinir ağı problemlerinden en çok kullanılan yöntem olan GYA kullanılmıştır. Ağın giriş katmanındaki nöron sayısı giriş parametresi kadar yani 15 tanedir. Çıkış katmanı ise tek sonuç bulunacağı için 1 tanedir. Ara katman sayısı ve hücre sayıları için kesin bir kural yoktur. Bu nedenle bu konuda daha önce yapılmış çalışmalardan yararlanılarak değişik mimariler denenerek en uygun çözüm

bulunmaya çalışılmıştır. Ara katmanda, yaygın görüş bir ara katmanın yeterli olacağı yönündedir. Ayrıca 2 katmanlı denemelerde yapılmıştır.

Her bir eğitim algoritması değişik ağ mimarilerinde en az 5'er kez çalıştırılıp ortalaması alınarak en uygun ağ yapısı bulunmaya çalışılmıştır. Eğitimde veri setinin %70'i, çapraz doğrulama (cross validation) % 20, test kümesinde % 10'u kullanılmıştır. Model 5000 döngü (iterasyon, Epoch) kullanılarak eğitilmiştir. Ayrıca, Neurosolutions programı ekranda görsellik sağladığı için, istenilen anda eğitimi kesip, ağın gereksiz yere öğrenmesine ve zaman kaybı önlenmiştir.

Veri seti, yanlış sonuçlara yol açmamak için normalize edilmiştir. Uygulamada kullanılan Neurosolutions programının özelliği olan normalize özelliğinden yararlanılarak veriler (-1,1) arsasına indirgenmiştir.

Çalışmada, en düşük hatalı, en iyi tahmin eden en uygun mimariyi bulabilmek için şu yol izlenmiştir:

1. Daha önce yapılan çalışmalar incelenerek en uygun parametreler bulunup sabit tutularak gizli katmandaki nöron sayısı bulunmaya çalışılmıştır. Önce öğrenme oranı 0.1, momentum 0.9, aktivasyon fonksiyonları sırasıyla Hiperbolik tanjant (tanjantaxon) ve sigmoid olarak belirlenmiştir.
2. En uygun nöron sayısı bulunduktan sonra en uygun öğrenme oranı ve momentum sayısını bulmak için çeşitli denemeler yapılmıştır.
3. En uygun katman sayısı, öğrenme oranı ve momentum katsayısı bulunduktan sonra, en uygun aktivasyon fonksiyonunu bulmak için denemeler yapılmıştır.
4. Son olarak da 2 ara katmanlı ağ yapısı denenerek kıyaslanmıştır.

6.2.1. Ara Katmandaki En Uygun Nöron Sayısının Bulunması

Girdi katmanında 15 nöron vardır. Ara katmandaki nöron sayısı ise, daha önce yapılan çalışmalarda giriş katmanındaki nöron sayısının 2 katı civarında en iyi sonuç alındığı belirtildiğinden, 9 nörondan başlayıp, 45 nörona kadar denemeler yapılmıştır. 5000 döngü (epoch) ile öğrenme oranı 0.1, momentum 0.9, aktivasyon

fonksiyonları sırasıyla Hiperbolik tanjant (tanjantaxon) ve sigmoid olarak belirlenmiştir. Bu denemelerin sonuçları aşağıdaki tabloda gösterilmiştir.

Ara Katman Sayısı:1 Öğrenme Oranı: 0,1 Momentum: 0,9 Döngü: 5000			
Aktivasyon Fonksiyonu: Hiperbolik Tanjant (Giriş-Ara Katman arasında), Sigmoid (Ara Katman-Çıktı arasında)			
Ara Katmandaki Nöron Sayısı	MSE	Tahmin	r
9	0,0329	0,9455	0,9050
10	0,0328	0,9501	0,9069
11	0,0320	0,9523	0,9079
12	0,0318	0,9521	0,9068
13	0,0333	0,9523	0,9038
14	0,0318	0,9523	0,9095
15	0,0320	0,9521	0,9088
16	0,0321	0,9490	0,9081
17	0,0319	0,9487	0,9093
18	0,0317	0,9501	0,9090
19	0,0320	0,9501	0,9107
20	0,0317	0,9522	0,9093
21	0,0325	0,9522	0,9068
22	0,0313	0,9595	0,9100
23	0,0329	0,9433	0,9088
24	0,0314	0,9510	0,9114
25	0,0324	0,9522	0,9069
26	0,0313	0,9498	0,9097
27	0,0317	0,9522	0,9092
28	0,0312	0,9514	0,9096
29	0,0310	0,9522	0,9118
30	0,0327	0,9514	0,9057
31	0,0325	0,9433	0,9067
32	0,0333	0,9457	0,9040
33	0,0322	0,9514	0,9071
34	0,0335	0,9466	0,9046
40	0,0338	0,9482	0,9082
45	0,0325	0,9521	0,9068

Tablo 6.2: Ara Katmandaki En Uygun Nöron Sayısı için Deneme Sonuçları

Görüldüğü gibi, analizin sonuçları birbirlerine çok yakın çıkmıştır. Fakat, en iyi sonuç. % 3,1 hata ve % 95,22 tahmin ile ara katmandaki nöron sayısı 29 olarak belirlenmiştir. En iyi ağ yapısı: 15-29-1 seçilmiştir. Bu ağa ait bulgular, Ek-2’de bulunmaktadır.

6.2.2. En Uygun Öğrenme Oranı ve Momentum Katsayısının Belirlenmesi

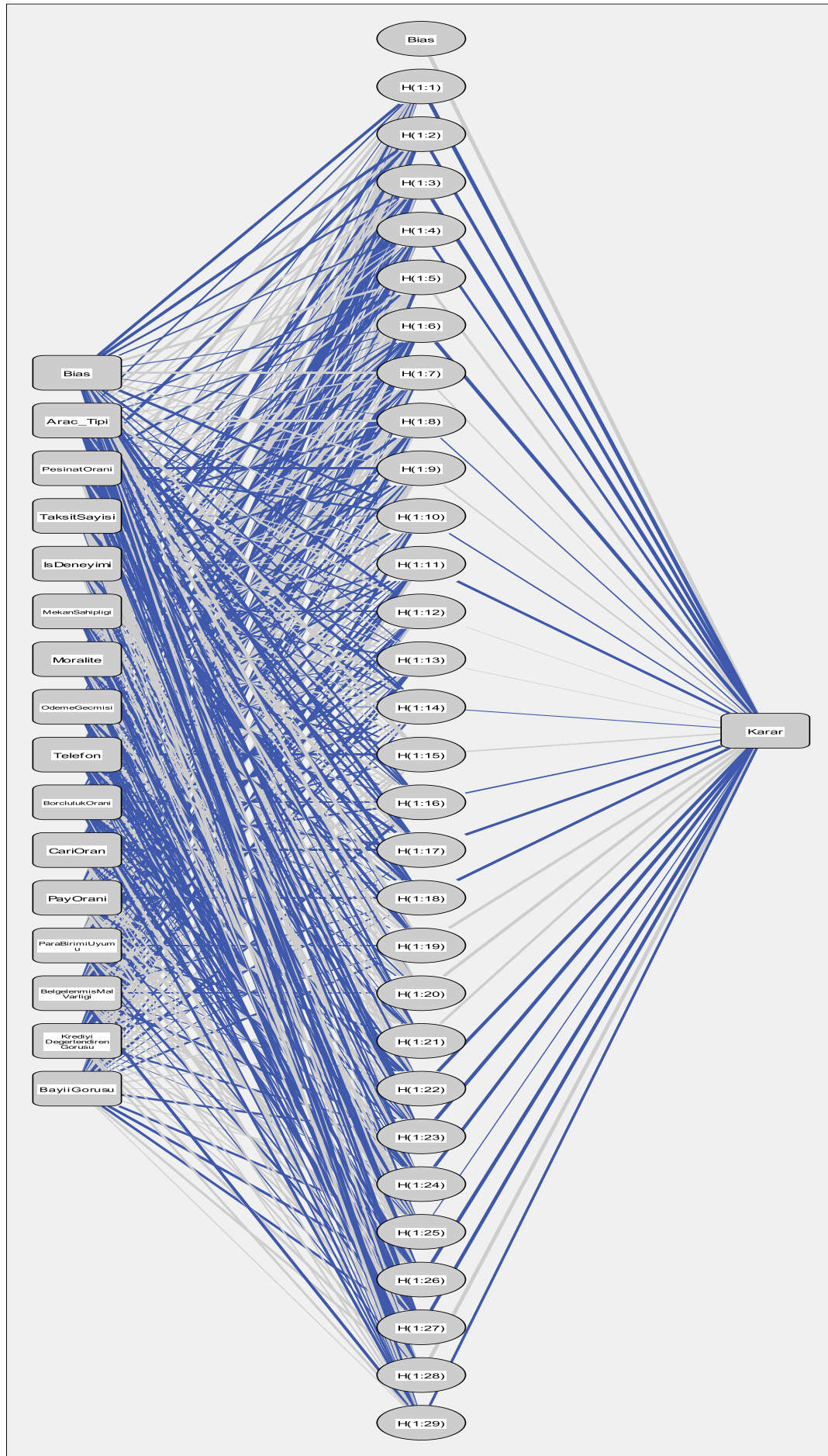
En uygun momentum katsayısı ve öğrenme oranı için literatürde çok çeşitli değerler bulunmuştur. Bu çalışmada, 2009 yılında Science Direct’de yayınlanan Chuang C.L. ve Lin R.H¹⁵⁰.’nin çalışmalarında kullanılan değerler kullanılarak denemeler yapılmıştır.

¹⁵⁰Chuang, Chun-Ling ve Lin, Rong-Ho, Constructing a reassigning credit scoring model” Expert Systems with Applications 36,1685–1694,2009.

Ağ Yapısı 15-29-1 Döngü: 5000				
Aktivasyon Fonksiyonu: Hiperbolik Tanjant (Giriş-Ara Katman arasında), Sigmoid (Ara Katman-Çıktı arasında)				
Öğrenme Oranı	Değerler	Momentum Katsayısı		
		0,8	0,9	0,99
0,01	MSE	0,035	0,033	0,031
	Tahmin	0,955	0,950	0,957
	r	0,899	0,904	0,912
0,03	MSE	0,033	0,032	0,031
	Tahmin	0,952	0,952	0,958
	r	0,905	0,908	0,913
0,05	MSE	0,032	0,031	0,027
	Tahmin	0,951	0,950	0,962
	r	0,908	0,910	0,926
0,07	MSE	0,032	0,032	0,029
	Tahmin	0,952	0,951	0,961
	r	0,911	0,909	0,921
0,09	MSE	0,032	0,031	0,029
	Tahmin	0,951	0,950	0,961
	r	0,907	0,908	0,921
0,1	MSE	0,032	0,031	0,030
	Tahmin	0,950	0,952	0,963
	r	0,908	0,912	0,920
0,2	MSE	0,031	0,031	0,032
	Tahmin	0,953	0,951	0,960
	r	0,910	0,911	0,910
0,3	MSE	0,032	0,032	0,030
	Tahmin	0,953	0,953	0,962
	r	0,909	0,907	0,922
0,4	MSE	0,032	0,031	0,033
	Tahmin	0,950	0,956	0,960
	r	0,910	0,909	0,905

Tablo 6.3: En Uygun Öğrenme Oranı ve Momentum Katsayısı Deneme Sonuçları

Yapılan denemeler sonucunda, en uygun öğrenme oranı 0,05, en iyi momentum sayısı ise 0,99 olarak bulunmuştur. Hata oranı, % 2,7' ye düşmüş ve ağ % 96,2 doğru tahmin etmiştir. Bu ağa ait bulgular Ek-3'de verilmiştir.



Şekil 6.1: En Uygun Yapay Sinir Ağı Modeli (15-29-1)

6.2.3. En Uygun Aktivasyon Fonksiyonunun Bulunması

YSA analizlerinde en çok kullanılan aktivasyon fonksiyonları olan hiperbolik tanjant (tanh), sigmoid ve bias fonksiyonları denenerek analiz yapılmıştır.

Ağ Yapısı 15-29-1 Döngü: 5000				
Öğrenme Oranı:0,05 Momentum:0,99				
Giriş Katman-Ara Katman	Değerler	Ara Katman-Çıktı Katmanı		
		sigmoid	tanh	bias
sigmoid	MSE	0,063	0,146	0,168
	Tahmin	0,898	0,934	0,940
	r	0,818	0,881	0,876
tanh	MSE	0,027	0,127	0,126
	Tahmin	0,962	0,965	0,958
	r	0,926	0,907	0,904
bias	MSE	0,032	0,103	0,256
	Tahmin	0,960	0,962	0,936
	r	0,915	0,922	0,815

Tablo 6.4: En Uygun Aktivasyon Fonksiyonu Deneme Sonuçları

En iyi sonuç yine hiperbolik tanjant-sigmoid aktivasyon fonksiyonları ile sağlanmıştır. Bu nedenle tahmin sonuçlarında herhangi bir değişiklik olmamıştır.

6.2.4. İki Ara Katmanlı Ağ Yapısı Denemesi:

Ağ Yapısı 15-29-1 Döngü: 5000			
Aktivasyon Fonksiyonu: Hiperbolik Tanjant (Giriş-Ara Katman arasında), Sigmoid (Ara Katman-Çıktı arasında)			
Öğrenme Oranı: 0,05 Momentum 0,99			
Ağ Yapısı	MSE	Tahmin	r
15-29-29-1	0,044	0,939	0,877
15-14-15-1	0,038	0,947	0,897
15-15-14-1	0,040	0,939	0,887
15-13-12-1	0,041	0,943	0,876
15-12-13-1	0,038	0,939	0,896
15-10-5-1	0,045	0,912	0,877

Tablo6.5: İki ara katman Deneme Sonuçları

Tablodan anlaşıldığı üzere ara katman sayısını arttırmak sinir ağının tahmin gücüne herhangi bir katkıda bulunmamaktadır. Bu nedenle, karmaşık ağ yapılarının ağın öğrenmesini zorlaştırdığı sonucuna da ulaşabiliriz. Ayrıca, ara katman sayısını arttırmak, ağın öğrenme süresini de arttırmıştır.

Yapılan Analizler Sonucunda En Uygun Ağ Geri Yayılım Model Yapısı: şöyledir:

Ağın Türü	Çok Katmanlı Perseptron
Öğrenme Algoritması	Geri Yayılım
Giriş Katmanındaki Nöron Sayısı	15
Eğitim Kümesi	70%
Onaylama Kümesi	20%
Test Kümesi	10%
Aktivasyon Fonksiyonu	Hiperbolik Tanjant-Sigmoid
Ara Katman	1
Ara Katmandaki Nöron Sayısı	29
Öğrenme Oranı	0,05
Momentum	0,99
Devir Sayısı	5000

Tablo6.6 : En Uygun Ağ Yapısı

Bu modelden elde edilen tahmin sonucu ise (yapılan denemelerin modu yani, en çok tekrarlanan tahmin sonucu alınmıştır):

Tahmin/Gerçek	Hayır	Evet
Hayır	71 (% 93,24)	6 (% 6,76)
Evet	4 (% 4,62)	166 (% 95,38)
Toplam	75	172
Toplam Doğru Sınıflandırma Oranı: % 95,95		

Tablo6.7: YSA Sınıflandırma Sonuçları

Modelin doğru sınıflandırma yüzdesi % 95,95’dir.

6.3. Lojistik Regresyon Analizi Sonuçları

Kredi talepleri verileri üzerinde, SPSS 18.0 for Windows programı kullanılarak lojistik regresyon yöntemi uygulanmıştır. Verilerin % 90’ı kullanılarak model oluşturulmuş, % 10’u test kümesi olarak ayrılmıştır. Bu test kümesi, YSA sonuçları ile ROC eğrisinde karşılaştırılmıştır.

Veri seti, ham halinde, sıfır ve sıfırın altında değerler içerdiğinden, lojistik regresyon analizi için uygun hale getirilmiştir. Ordinal ölçekli değişkenlerin puan değerleri, küçükten büyüğe sıralanarak 1,2,3.. sıra numaraları verilerek yeniden düzenlenmiştir.

Oluşturulan veri 2225 gözlemden oluşmakta ve kredi taleplerinin %31,4’ü hayır, % 68,6’sı evet olarak sonuçlanmıştır. Karar değişkeni 1 ve 0 değerleri almaktadır. 0 hayır, 1 ise evet anlamına gelmektedir.

Karar					
		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	hayır	698	31,4	31,4	31,4
	evet	1527	68,6	68,6	100,0
	Total	2225	100,0	100,0	

Tablo6.8: LR Hedef Değişken Betimsel İstatistikleri

Daha önce de bahsedildiği gibi, başlangıçta, 17 bağımsız değişkenle çalışılmaya başlanmış, birbirleriyle ilişkili olan 2 değişken modelden çıkarılarak analizler yapılmıştır. Çünkü lojistik regresyon (LR) analizi bağımsız değişkenler arasında çoklu doğrusal bağlantı olmadığını varsayar. Enter yöntemi ile yapılan analiz sonucunda, peşinat oranı, taksit sayısı, krediyi değerlendiren görüşü ve bayii görüşü olmak üzere 4 bağımsız değişkenle kurulan model, en uygun sonuç olarak bulunmuştur.

Enter yöntemi ile lojistik regresyon modeli kurulurken, önce bütün değişkenler modele dahil edilmiş, katsayıların anlamlılık seviyelerine göre anlamsız olan değişkenler modelden atılarak en uygun model bulunmaya çalışılmıştır.

Classification Table ^{a, b}					
Observed			Predicted		
			Karar		Percentage Correct
			hayır	evet	
Step 0	Karar	hayır	0	698	,0
		evet	0	1527	100,0
Overall Percentage					68,6

a. Constant is included in the model.

b. The cut value is ,500

Tablo6.9: Başlangıç Adımı Sınıflandırma Tablosu

Variables in the Equation

	B	S.E.	Wald	df	Sig.	Exp(B)
Step 0 Constant	,783	,046	293,570	1	,000	2,188

Tablo6.10: Başlangıç Adımı Anlamlılık Testi

Başlangıç çözümü incelendiğinde, kredi talebi sonucunun olumlu olduğu firmaların hepsinin doğru tahmin edildiği, buna karşın reddedilen firmaların tamamının da yanlış atandığı görülmektedir. Bunun ana sebebi, başlangıç çözümünde oluşan denkleme sadece sabit katsayının alınmış olunmasıdır.

6.3.1. Modelin Anlamlılık Testi:

Omnibus Tests of Model Coefficients

	Chi-square	df	Sig.
Step 1 Step	1212,265	5	,000
Block	1212,265	5	,000
Model	1212,265	5	,000

Tablo6.11: Model katsayılarının Omnibus Testleri

Model katsayıları, SPSS paket programında Omnibus Ki-Kare testi ile test edilir. Bu test, bir blok değişkeninin modele dahil edilmesiyle meydana gelen değişmeyi gösterir.¹⁵¹ Katsayıların genel olarak test edildiği bu testin hipotezi şu şekildedir:

$$H_0 : B_1 = B_2 = \dots = B_K = 0$$

$$H_1 : B_1 \neq B_2 \neq \dots \neq B_K \neq 0$$

Bu hipotez, tabloda değerine göre, % 5 anlamlılık seviyesinde reddedilmiştir. Yani, lojistik regresyon katsayıları aynı anda sıfıra eşit değildir.

¹⁵¹ Kalaycı,a.g.e.,s.292

Model Summary

Step	-2 Log likelihood	Cox & Snell R Square	Nagelkerke R Square
1	1555,793 ^a	,420	,590

a. Estimation terminated at iteration number 8 because parameter estimates changed by less than ,001.

Tablo6.12: Model Özet Tablosu

Modele ait Nagelkerke $R^2 = 0,59$ olarak bulunmuştur. Bu değer, regresyon analizinde kullanılan modelin açıklayıcısı olan R^2 değeriyle tamamı ile aynı işlevde değildir (Menard, 2002). Fakat, onun ikamesi olarak kullanılır. Yani kurulan lojistik modelin kullanılan değişkenlerce açıklanma oranının %59 olduğunu söyleyebiliriz. Ayrıca Cox ve Snell R^2 de %42 olarak bulunmuştur ki bu açıklama oranlarının yeterince yüksek olduğu ve kurulan modelin anlamlı olduğu söylenebilir. Modelin verileri iyi temsil edip etmediğini test etmek için, -2LogL istatistiği hesaplanmaktadır. Model verilerini tam temsil edilmesi durumunda olabilirlik bir olur ve -2LogL istatistiği sıfır değerini alır. -2LogL istatistiği, k modeldeki değişken sayısını göstermek üzere, n-k serbestlik derecesiyle Ki-Kare dağılımına uyar.

6.3.2. Modelin Uyum İyiliği Testi

Hosmer and Lemeshow Test

Step	Chi-square	df	Sig.
1	11,059	8	,198

Tablo6.13: Hosmer Lemeshow Uyum İyiliği Testi

Bu istatistik, “Hosmer ve Lemeshow G” istatistiği olarak da bilinir. Model Ki- Kare istatistiği, lojistik regresyon modelini genel olarak test etmektedir. Bağımsız değişkenlerden hiçbirinin bağımlı üstünlük oranıyla anlamlı doğrusal bir ilişki göstermediğini ileri süren sıfır hipotezini test etmektedir.

H_0 : Parametreler iyi bir ayırmıcıdır.

H_1 : Parametreler iyi bir ayırmıcı değildir.

Test istatistiğine göre, % 5 anlamlılık seviyesinde, H_0 kabul edilmiş ve Lojistik regresyon modelinin, kredi taleplerinin sonuçlandırılmasında iyi bir model olduğu bulunmuştur.

Uygulama sonucunda denklemdaki değişkenlere ait katsayılar, standart hatalar, Wald istatistikleri, Wald istatistiğine ait serbestlik dereceleri, katsayılarla ilişkin önemlilik düzeyleri verilmiştir.

Classification Table^a

Observed			Predicted		
			Decision		Percentage Correct
			hayır	evet	
Step 1	Decision	hayır	500	198	71,6
		evet	152	1375	90,0
	Overall Percentage				84,3

a. The cut value is ,500

Tablo6.14: LR Sınıflandırma Tablosu

Modelin doğru sınıflandırma yüzdesi % 84,3'dür.

Menard'a göre modelin teorik olarak anlamlılığı en iyi Omnibus Ki-Kare testi ve McFadden R^2 değeri ile ölçülür. McFadden R^2 istatistiği, modele dahil edilen bağımsız değişkenlerin, modelin anlamlılığını ne kadar arttırdığını ölçer. Omnibus Ki-Kare testi SPSS çıktılarında olduğundan yorumlanabildi fakat, McFadden R^2 istatistiği SPSS'de hesaplanmaz. Fakat kolayca elle hesaplanabilir.

$$\text{McFadden } R^2 = \frac{G_M}{G_M + D_M}$$

G_M , bağımsız değişkenlerin ilave edildiği model ile sadece sabit terimin olduğu modelin arasındaki hata farkını verir. Bu değer, SPSS çıktısında Omnibus Testte bulunan Ki-Kare değeridir. D_M ise, analize bağımsız değişken(ler)in ilave edildiğinde

modelin hatasını gösteren -2LogL istatistiğidir. Bu istatistik, bağımlı değişkendeki açıklanmayan varyansın anlamlılığını gösterir¹⁵².

$$\text{McFadden } R^2 = \frac{1212,265}{1212,265 + 1555,793} = 0,438$$

McFadden R^2 istatistiğine göre, modele bağımsız değişkenlerin eklendiği model, sadece sabit terimin olduğu modele göre % 43,8 daha iyidir denebilir.

Modelin anlamlılığı test edildikten sonra, tahmin gücü hakkında da SPSS paket programında hesaplanmayan bazı önemli istatistikler vardır¹⁵³.

- **λ_p (Lamda-p):** Bir değişken bilindiğinde, diğer değişkenin tahminindeki hatanın göreceli azalmasının ölçütüdür. Hatadaki orantısal azalma (proportional reduction in error) fikrine dayalıdır. Bu değer pozitif olması, tahminin iyi olduğunu, negatif olması ise sınıflandırmanın iyi olmadığını göstermektedir. Bu değer, 1 ile (1-N) arasında bir değer alabilir.

$$\lambda_p = \frac{(\text{Az Sayıdaki Gözlenen Değerler}) - (\text{Yanlış Sınıflandırılan Değerler})}{(\text{Az Sayıda Gözlenen Değerler})}$$

$$\lambda_p = \frac{698 - (152 + 198)}{652} = 0,4986$$

Sınıflandırmanın orantısal hata azalması % 49,86 bulunmuş olup, bu oldukça iyi bir değerdir.

- **τ_p (Tau p):** Lamda p gibi değişimin ölçüsüdür. Bağımsız değişken(ler)in, modelin sınıflandırma hatasını ne kadar düşürdüğünü gösterir. -1 ile +1 arasında değer alır¹⁵⁴.

¹⁵²Kalaycı,a.g.e,s.292.

¹⁵³Menard,a.g.e.

$$\tau_p = \frac{\frac{2x(\text{Birinci Kategorideki Gözlem Sayısı})x(\text{İkinci Kategorideki Gözlem Sayısı})}{\text{Toplam Gözlem Sayısı}} - \text{Yanlış Sınıflandırılan Değerler}}{\frac{2x(\text{Birinci Kategorideki Gözlem Sayısı})x(\text{İkinci Kategorideki Gözlem Sayısı})}{\text{Toplam Gözlem Sayısı}}}$$

$$\tau_p = \frac{\frac{2 \times 698 \times 1527}{2225} - 350}{\frac{2 \times 698 \times 1527}{2225}} = 0,6347$$

Bu model, gözlemlerin sınıflandırma hatasını %63,47 azaltmıştır.

6.3.3. Modelin Katsayıları:

Buna göre Lojistik Regresyon Denklemi:

$$\hat{Y} = \frac{1}{1 + e^{-z}} \text{ olmak üzere,}$$

$z = 5,176 + 0,057\text{Peşinat Oranı} - 0,064\text{Taksit sayısı} - 4,582\text{Krediyi Değerlendiren Görüşü(1)} - 5,214\text{BayiiGörüşü (1)} - 2,262\text{Bayii Görüşü (2)}$

Variables in the Equation						
	B	S.E.	Wald	df	Sig.	Exp(B)
Step 1 ^a PesinatOrani	,057	,005	115,728	1	,000	1,058
TaksitSayisi	-,064	,006	116,302	1	,000	,938
KrediyiDegerlendirenGorusu(1)	-4,582	,620	54,682	1	,000	,010
BayiiGorusu			222,138	2	,000	
BayiiGorusu(1)	-5,214	1,016	26,350	1	,000	,005
BayiiGorusu(2)	-2,262	1,030	4,828	1	,028	,104
Constant	5,176	1,052	24,216	1	,000	176,888

a. Variable(s) entered on step 1: PesinatOrani, TaksitSayisi, KrediyiDegerlendirenGorusu, BayiiGorusu.

Tablo 6.15: LR Model Katsayıları ve Anlamlılık Testleri

Bu tablodaki Wald Ki-Kare istatistiği, sabit terim ve bağımsız değişkenlerin anlamlılığını test eder.

¹⁵⁴Menard,a.g.e,s.33.

H_0 :Değişkenler anlamsızdır.

H_1 :Değişkenler anlamlıdır.

Değişken tablosunda elde edilen sonuçlarda sabit terimin standart hatası, değişkenin anlamlılığını test eden Wald istatistiği, Wald istatistiğinin anlamlılık düzeyi ve ilgili değişken bir birim arttırıldığı zaman üstünlük oranındaki değişimini gösteren odds oranı $\text{Exp}(B)$ istatistiği verilmektedir. Wald değeri ne kadar yüksekse, modelde değişken o kadar önemlidir. Modelde, Bayiinin görüşü en önemli değişkendir.

Tahmin edilen lojistik regresyon analizi modelinin yorumları odds oranları kullanılarak yapılır:

Peşinat oranı ve taksit sayısı % 5 anlamlılık seviyesinde modelde anlamlı değişkenlerdir. Peşinat oranının, bir birim artması kredinin olumlu sonuçlanması olasılığını 1,058 kat arttırmakta, taksit sayısının bir birim artması ise olasılığı 0,938 kat azaltmaktadır.

Krediyi değerlendiren kişinin görüşü % 5 anlamlılık seviyesinde modelde anlamlı bir değişkendir. Krediyi değerlendiren kişinin görüşünün iyi olması referans değeri olarak alınmıştır. Krediyi değerlendiren kişinin, hakkında görüşünün kötü olduğu firmanın talebinin kabul edilme olasılığı, görüşünün iyi olduğu bir firmanın kredi talebinin olumlu değerlendirilmesi olasılığına göre 0,01 kat daha düşüktür

Bayi görüşü, % 5 anlamlılık seviyesinde modelde anlamlı bir değişkendir. Bayi görüşünün “çok iyi” olması referans değeri olarak alınmıştır. Bayiinin firma hakkında herhangi bir fikir belirtmediği firmaların, kredi taleplerinin olumlu sonuçlanma olasılığı, çok iyi olduğunu belirttiği firmaların kredi talebinin olumlu sonuçlanma olasılığına göre 0,005 kat daha azdır. Bayiinin, firma hakkında “iyi” olduğunu belirttiği firmaların kredi taleplerinin olumlu sonuçlanma olasılığı, “çok iyi” olduğunu belirttiği firmaların kredi taleplerinin olumlu sonuçlanma olasılığına göre 0,104 kat daha azdır.

6.4. ROC Eğrileri (Receiver Operating Characteristic Curves) Analizi

6.4.1. ROC Eğrileri

Sınıflandırma problemlerinde, doğru sınıflandırma oranı en çok kullanılan ölçüttür. Fakat çoğunlukla, bu oran tek başına modelin anlamlılığını test etmede yeterli olmamaktadır. Çünkü doğru sınıflandırma oranı, 1. tip hata ile 2. tip hatayı vermemektedir. Bu nedenle seçilen farklı eşik değerleri için bulunan farklı duyarlılık-özellik karakterlerine bağlı olarak ara seçenekler belirlenerek, ROC eğrileri (*Receiver operating characteristic curves*) oluşturulmuştur. Tahmin modellerinin performanslarının değerlendirilmesi ve kıyaslanması için bu sınırlamaların üstesinden gelen en değerli yöntemlerden biri bu eğrilerdir. Doğruluk, duyarlılık ve özellik gibi performans ölçüleri sadece önceden belirlenmiş bir kopuş değeri için geçerliken, ROC eğrisi, önsel olasılıklardan ve kopuş değerlerinden bağımsız bir performans ölçüsüdür¹⁵⁵.

ROC eğrisi yöntemi;

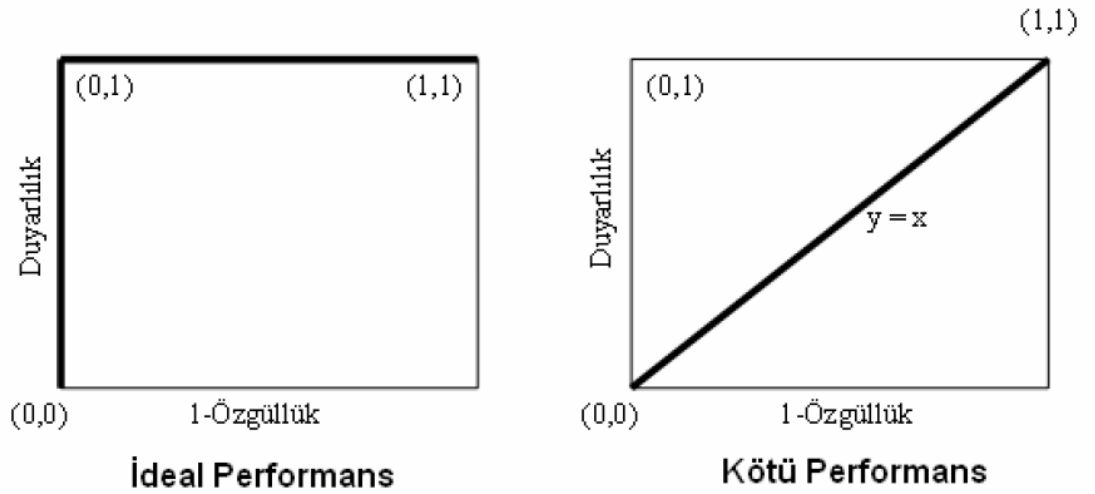
- Modelin sınıflandırma gücünün belirlenmesine,
- Çeşitli modellerin etkinliklerinin kıyaslanmasına,
- En uygun kopuş değerinin belirlenmesine,
- Sonuçların kalitesinin izlenmesine olanak sağlamaktadır¹⁵⁶.

ROC eğrisinin oluşturulacağı koordinat sisteminin ordinatında tanı testinin gerçek pozitif değeri (duyarlılık), apsisinde ise yanlış pozitif değeri (1-özellik) yer alır. Tanı testi ne kadar iyi ise eğri o kadar yukarıya (yüksek duyarlılık bölgesi) ve sola

¹⁵⁵ Talip Torun, Finansal Başarısızlık Tahmininde Geleneksel İstatistikî Yöntemlerle Yapay Sinir Ağlarının Karşılaştırılması ve Sanayi İşletmeleri Üzerinde Uygulama, Erciyes Üniversitesi Sosyal Bilimler Enstitüsü Yüksek Lisans Tezi, 2007, s.115.

¹⁵⁶ Dirican, Ahmet; Tanı Testi Performanslarının Değerlendirilmesi ve Kıyaslanması, Cerrahpaşa Tıp Dergisi, 32-1,25-30,2001.

(düşük yanlış pozitif oranı bölgesi) doğru kayar. Yanlış değerlere sahip olmayan ideal bir testte ROC eğrisi (0,0)-(0,1)-(1,1) noktalarını birleştirmektedir. Buna karşın ROC çizimi $y=x$ fonksiyonuna yaklaştıkça başarısız bir test ortaya çıkar. Çünkü bu testte yanlış değerlerin oranı en yükselmektedir. Bu fonksiyonun altındaki ROC eğrisine sahip test başarısızdır¹⁵⁷.



Şekil 6.2: İdeal ve Kötü Performans Göstergesi Olan ROC Eğrileri

ROC eğrisi altında kalan alan, aynı örnek üzerinde geliştirilmiş modellerin performansını karşılaştırmada sağlam bir performans ölçüsüdür. Bir modelin ROC eğrisi altında kalan alanı, modelin doğru tahmin olasılığını göstermektedir. Böylece, bir tanı testi için ROC altında kalan alan etkinlik düzeyine bağlı olarak 0.50 ile 1.00 arasında değerler alabilecektir. Bu alan ne kadar büyükse, yapılan tahmin o kadar ayırım yeteneğine sahip olacaktır. Bu alan 0.975 ve daha üzerinde ise mükemmel sayılmaktadır¹⁵⁸.

¹⁵⁷Dirican,a.g.e.

¹⁵⁸Torun,a.g.e,s.116.

Özgüllük ve Duyarlılık Oranlarının Bulunması:

		Gerçek Durum		
		Başarısız	Başarılı	Toplam
Tahmin Sonucu	Başarısız	A (GP)	B (YP)	A+B
	Başarılı	C (YN)	D (GN)	C+D
	Toplam	A+C	B+D	A+B+C+D

Tablo 6.16: Genel Sınıflandırma Matrisi

Gerçekte başarısız olan işletmeler açısından; “A” gözü gerçek duruma uygun olarak modelin de başarısız olarak tahmin ettiği işletmeler, gerçek pozitifler (GP) olarak adlandırılırlar. Buna karşılık C gözündeki olgular, gerçekte başarısız olup modelin başarılı olarak tahmin ettiği yani yanlış negatif (YN) işletmelerdir. Bu doğrultuda Modelin duyarlılık (Sensitivity) ve özgüllük (Specificity) değerleri:

Duyarlılık: Testin, gerçekte başarısız olan değerleri yine başarısız olarak ayırma yeteneğidir. $Duyarlılık = A/(A+C) = GP/(GP+YN)$

Bu oran II. Tip doğruluk olarak da adlandırılır¹⁵⁹.

Özgüllük: Testin, gerçekte başarılı olan değerleri yine başarılı olarak ayırma yeteneğidir.

$$Özgüllük = D/(D+B) = GN/(GN+YP)$$

Bu oran I. Tip doğruluk olarak da adlandırılır¹⁶⁰.

Yine sınıflandırma problemlerinde, I.tip ve II. Tip hata önemli birer göstergedir. I. Tip hata, gerçekte başarılı olan gözlemlerin başarısız olarak sınıflandırılması (B), II. Tip hata ise gerçekte başarısız olan gözlemlerin, başarılı olarak sınıflandırılmasıdır (C). Analiz sonucundaki, bütün veri için bulunan sınıflandırma oranı, bu oranları göz ardı etmektedir.

¹⁵⁹Yu, L., Wang, Wang S., Lai K.K., “An intelligent-agent-based fuzzy group decision making model for financial multicriteria decision support: The case of credit scoring”, European Journal of Operational Research 195, 942–959, 2007,s.948.

¹⁶⁰ A.e.

6.4.2. Roc Eğrisi Analiz Sonuçları

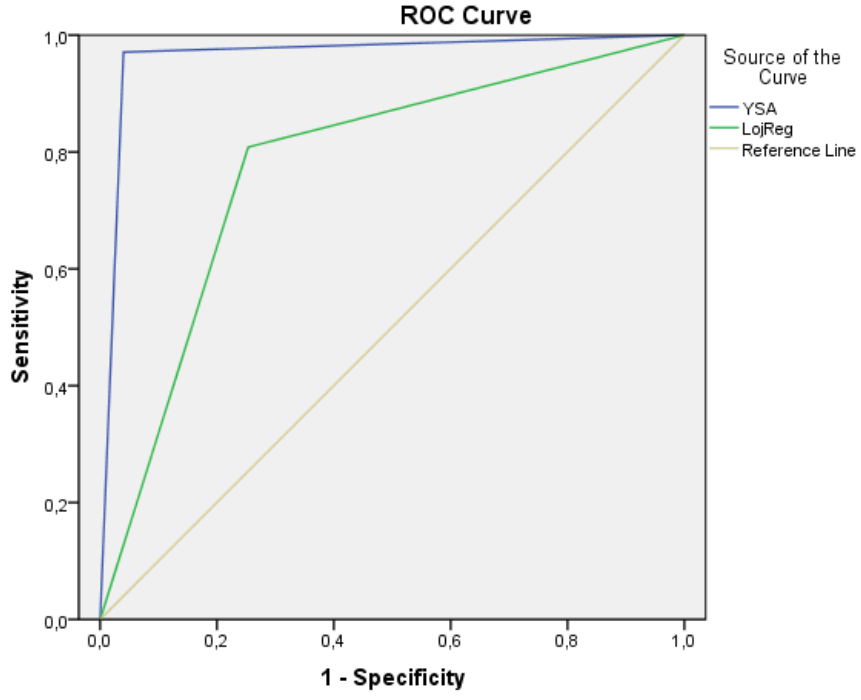
Yapay sinir ağı analizi sonuçları ile lojistik regresyon analizi sonuçlarını objektif bir şekilde karşılaştırmak için, lojistik regresyonda test kümesi için % 10 veri kümesi ayrılmış olup, bu veri setine, lojistik regresyon modeli uygulanarak tahmin sonucu bulunmuştur. Bu tahmin sonuçlarına göre sınıflandırma oranları şöyle bulunmuştur:

Tahmin/Gerçek	Hayır	Evet	Toplam
Hayır	56 (%74,7)	33 (%19,2)	89
Evet	19 (%25,3)	139 (%80,8)	158
Toplam	75	172	247
Toplam Doğru Sınıflandırma 195 (%78,9)			

Tablo 6.17: LR Test Kümesi Sınıflandırma Tablosu

Görüldüğü gibi, anlamlı modelin tahmin gücü % 84,3 iken, modelin, hiç görmediği test kümesinin tahmin gücü % 78,9'a düşmüştür. Yapay sinir ağında bu oran ise % 95,95 olarak bulunmuştur.

Kredi taleplerinin yapay sinir ağı ve lojistik regresyon tahmin sonuçları ROC eğrisinde gerçek sonuçlarla karşılaştırılarak aşağıda gösterilmiştir.



Diagonal segments are produced by ties.

Şekil 6.3: ROC Eğrisi Grafiği

Area Under the Curve					
Test Result Variable(s)	Area	Std. Error ^a	Asymptotic Sig. ^b	Asymptotic 95% Confidence Interval	
				Lower Bound	Upper Bound
YSA	,965	,015	,000	,936	,995
LojReg	,777	,034	,000	,711	,844

Tablo 6.18: ROC Eğrisi Anlamlılık Testi

Yapılan analiz sonucunda, yapay sinir ağının ROC eğrileri altında kalan alan % 96,5, lojistik regresyonunda ise % 77,7 olarak bulunmuştur. Yapay sinir ağının kredi taleplerini değerlendirmede mükemmele yakın tahminde bulunduğu söylenebilir. Anlamlılık düzeyine bakıldığında, bulunan sonuçların % 5 anlamlılık seviyesinde istatistiksel olarak anlamlı olduğu söylenebilir.

Coordinates of the Curve			
Test Result Variable(s)	Positive if Greater Than or Equal To ^a	Sensitivity	1 - Specificity
YSA	-1,0000	1,000	1,000
	,5000	,971	,040
	2,0000	,000	,000
LojReg	-1,0000	1,000	1,000
	,5000	,808	,253
	2,0000	,000	,000

Tablo 6.19: ROC eğrisi Duyarlılık ve Özgüllük Değerleri

Bu tabloda modellerin Duyarlılık (II. Tip doğruluğu) ve Özgüllükleri (I. Tip hata) görülmektedir. Yani, Yapay sinir ağı, gerçekten evet olarak sonuçlanması gereken değişkenlerin % 97,1'ini doğru sınıflandırmıştır. Gerçekte olumsuz olan kredilerin olumlu olarak değerlendirilmesi hatası ise % 4'dür. Lojistik regresyonda ise, gerçekten evet olarak sonuçlanması gereken değişkenlerin % 80,8'ini doğru sınıflandırmış, fakat gerçekte olumsuz olan kredilerin %25,3'ünü olumlu olarak değerlendirilmiştir. Duyarlılığı oldukça yüksek olan lojistik regresyon analizinin, I. Tip hatası ise yüksektir.

7. SONUÇLAR VE DEĞERLENDİRME

İnsanoğlunun kendine benzeyen varlıklar oluşturmaya, zekasını taklit etmeye çalışmasından ortaya çıkan yapay zeka teknolojilerinden biri de YSA'dır. Geleneksel çözüm yöntemlerinin, gerçek hayat problemlerinin karmaşıklığını çözmede yetersiz kalması üzerine, insan beyninin çalışma şeklini taklit ederek tahmin yapabilen YSA öne çıkmıştır.

YSA yöntemi finans alanında ve başka birçok alanda geleceğe dair tahmin yapabilmek için kullanılan yöntemlerden biridir. Bu çalışmada sınıflandırma problemini öngörebilmek için YSA kullanılmıştır. Bu nedenle, tezin ilk bölümlerinde yapay zeka ve YSA detaylı bir şekilde incelenmiş ve sık kullanılan bazı YSA mimarileri tanıtılmıştır. Uygulama kısmında ise YSA problemlerinin çözümlerinde en çok kullanılan, Geri Yayılım Algoritması kullanılarak kredi taleplerinin değerlendirilmesi yapılmıştır. YSA'nın tahmin gücünü karşılaştırmak için ise aynı verilere LR uygulanarak performansları, ROC eğrisinde karşılaştırılmıştır.

Analizler, bir özel bir finans kuruluşunun, 2472 firmaya ait binek ve hafif ticari araçlar kredi taleplerinden oluşmaktadır. Hedef değişken ise, bu taleplerin başvurularının olumlu/olumsuz sonuçlarıdır. Veride 17 bağımsız değişken vardır. İlişkili bağımsız değişkenlerin modelde anlamlılığı etkileyeceğinden, korelasyon analizi yapılmıştır. Korelasyon değeri % 70'in üzerinde olan iki bağımsız değişken, analizden çıkartılmıştır. Geriye kalan 15 değişkenin, 1 tanesi nominal, 2 tanesi oran, 12 tanesi ise ordinal ölçeklidir.

YSA analizinde, önce verileri % 70 öğrenme, % 20 onaylama ve % 10 test kümesi olarak ayrılmıştır. Gözlemler (-1,+1) arasında normalize edilmiş ve literatürde en çok kullanılan parametreler kullanılarak eğitilip, test edilmişlerdir. Sonuçta, bu uygulama için en iyi ağ yapısının 15-29-1 olduğu, momentum katsayısının 0,99, öğrenme oranının 0,05 olduğu, aktivasyon fonksiyonlarının ise hiperbolik tanjant ile

sigmoid fonksiyonları oldukları bulunmuştur. Bu ağın performansı ise % 2,7 hata ile % 96,2 tahmin ile oldukça yüksek olduğu görülmüştür.

Lojistik regresyon analizinde ise, veri yapısı lojistik regresyon analizine elverişli olmadığı için, puan değerleri kategorik veriye dönüştürülmüştür. Ayrıca LR Analizi ile YSA Analiz sonuçlarının ROC eğrisinde objektif bir şekilde karşılaştırılabilmesi için verilerin % 10'u test kümesi olarak ayrılmıştır. Dönüştürülen veride yapılan analizlerde, LR'nun 4 bağımsız değişkenle kredi taleplerinin % 84'ünü doğru sınıflandırdığı gözlenmiştir.

LR'de elde edilen model, Regresyon analizinde hiç kullanılmamış olan, toplam veri setinin % 10'unu oluşturan veride uygulanmış ve % 78,9 doğru sınıflandırma sonucu elde edilmiştir. Ardından, her iki analizin % 10'luk veri setinde yaptığı sınıflandırmalar, ROC Eğrisinde analiz edilmiştir. YSA'da karşılaştırma yapabilmek için, yapılan denemelerdeki en çok tekrarlanan sonuç, yani denemelerin modu kullanılarak LR ile karşılaştırılmıştır (% 95,95). YSA'da ROC eğrileri altında kalan alan %95,6, LR'de ise % 77,7 olarak bulunmuştur. Yapılan analizler sonucunda, bu uygulama için YSA'nın LR'a göre daha iyi bir tahminci olduğu söylenebilir.

YSA çok iyi tahminci olmalarına rağmen, kara kutu özelliği nedeni ile, tahminin nasıl yapıldığı bilinmemektedir. Yani, lojistik regresyondaki gibi bir modeli yoktur ve değişkenlerin önemi hakkında bir sonuç ortaya koymamaktadır. Bilgi, ağın içinde saklıdır.

YSA, kara kutu özelliğine rağmen, kullanıldığı hemen her alanda çok etkin bir analiz yöntemidir. Ayrıca, YSA'nın öngörü performanslarının artırılması ve kara kutu özelliğini çözebilmek için halen birçok araştırma ve çalışma sürdürülmektedir. Örneğin, ağın ezberlemesinin ve yerel minimum noktalarına takılmasını önleme için bulanık mantık ve genetik algoritmalar uygulamaları geliştirilmektedir.

KAYNAKÇA

KİTAPLAR

Allison, P. D., **Logistic Regression Using the SAS System:Theory and Application**, SAS Institute Inc, USA,2001.

Albayrak, Ali Sait.,**Uygulamalı Çok Değişkenli İstatistik Teknikleri**, Asil Yayın Dağıtım, 2006.

Anderson, D., McNeill G., **Artificial Neural Networks Technology**, Kaman Science Corporation, New York,1992.

Andersen, E.B., **The Statistical Analysis of Categorical Data**, Springer-Verlag, Berlin- Heidelberg,1994.

Agresti, Alan **Categorical Data Analysis**, A John Willey & Sons, Inc, Publication, Second Edition, 2002.

ELMAS, Çetin., **Yapay Sinir Ağları: Kuram, Mimari, Eğitim, Uygulama**, Seçkin Yayıncılık, Ankara, 2003.

Görz, Günther ve Bernhard, Nebel, **Yapay Zeka**, İnkılap Yayıncılık, İstanbul,2005

Gujarati, D.N., **Temel Ekonometri** (Çev., Ü., Senesen & G, Senesen), Literatür Yayıncılık, İstanbul, 2006.

Gurney, K., **Computers and Symbols versus Nets and Neurons**, UCL Draft Papers, No:1, U.K, 1996.

Kalaycı, Şeref, **SPSS Uygulamalı Çok Değişkenli İstatistik Teknikleri**, Asil Yayın, 2006.

Kröse, B. and van der Smagt P., **An Introduction to Neural Networks**, The University of Amsterdam,1996.

Luger, George F., **Artificial Intelligence 6e : Structures and Strategies for Complex Problem Solving**, Pearson Education, 2005.

MEHROTRA, Kishan, MOHAN, Kishan ve RANKA, Sanjay, **Elements of Artificial Neural Networks**, A Bradford Book, The MIT Pres, Massachusetts Institute of Technology, 1997.

Menard,S., **Applied Logistic Regression Analysis**, Sage Publications, California,2002.

Mitchell, Tom , **Machine Learning**,McGraw Hill, USA, 1997.

Minsky, M.L. and Papert, S.S., **Perceptrons**, MA:MIT Press,Cambridge,1969.

Nabiyev, Vasif V.:., **Yapay Zeka – Problemler, Yöntemler, Algoritmalar**, Seçkin Yayınevi,2003.

Özdamar, Kazım, **Paket Programlar ile İstatistiksel Veri Analizi-1**, Kaan Kitabevi, Eskişehir, 2004.

Öztemel, Ercan, **Yapay Sinir Ağları**, Papatya Yayıncılık, 2006.

Padhy, N.P., **Artificial Intelligence and Intelligent Systems**, Oxford University Press, 2005.

Russell, S. J. and Norvig, P.,**Artificial Intelligence: A Modern Approach** , 1995.

Whitby, Blay, **Yapay Zeka**, İletişim Yayıncılık, 2005.

Winstanley, Graham, **Artificial Intelligence in Engineering**, Wiley, New York, 1991.

SÜRELİ YAYINLAR

Akgöbek Ö., Çakır F., “Veri madenciliğinde bir uzman sistem tasarımı”, **Akademik Bilişim’09**, Harran Üniversitesi, 11-13 Şubat 2009.

Akkuş, Z., Çelik, Y., Satıcı, Ö., Dasdağ, M.M. ve Sanisoglu, Y., “Hastane Personelinin Kan Bağışı Hakkındaki Bilgi, Tutum ve Davranışlarının Çok Değişkenli Lojistik Regresyon Tekniğiyle İncelenmesi”, **İnönü Üniversitesi Tıp Fakültesi Dergisi**, 12(1):25-29,2005.

Abdou, Hussein, Pointon, John, El-Masry, Ahmed,”Neural Nets versus conventional techniques in credit scoring in Egyptian banking”,**Expert Systems with Applications** 35,1275-1292,2008.

Bircan, Hüdaverdi, “Lojistik Regresyon Analizi:Tıp Verileri Üzerine Bir Uygulama”,**Kocaeli Üniversitesi Sosyal Bilimler Enstitüsü Dergisi**,2004/2:s.185-208,2004.

Chuang, Chun-Ling ve Lin, Rong-Ho, “Constructing a reassigning credit scoring model”, **Expert Systems with Applications**, 36,1685–1694,2009.

Dima, A.M. ve Vasilache S., “ANN Model for Corporate Credit Risk Assessment” **2009 International Conference on Information and Financial Engineering**,2009.

Dirican, Ahmet; “Tanı Testi Performanslarının Değerlendirilmesi ve Kıyaslanması”, **Cerrahpaşa Tıp Dergisi**, Cilt 32, Sayı 1, ss.25-30, 2001.

Er, H., Çetin, M. K., Çetin, E. İ.,”Finansta Evrimsel Algoritmik Yaklaşımlar : Genetik Algoritma Uygulamaları” **Akdeniz İ.İ.B.F. Dergisi**. s: 73-94, 2005.

Huang Z., Chen H., Hsu, C.J., Chen W.H. ve Wu, S., “Credit rating analysis with support vector machines and neural networks: a market comparative study”, **Decision Support Systems**, 37,543– 558, 2003.

İşçi,Ö, Körükoğlu,S., "Genetik Algoritma Yaklaşımı ve Yöneylem Araştırmasında Bir Uygulama", - **CBÜ İİBF Yönetim ve Ekonomi Dergisi**(ISSN-1302-0064),Cilt 10 sayı :2,2003.

KAASTRA Iebling, BOYD Milton: “Designing a Neural Network for Forecasting Financial and Economic Time Series”, **Neurocomputing**, 1994.

Kayacan, M.C., Çelik, Ş.A., "Bilgisayar Yardımıyla CNC Tornalama İşlemleri İçin Kesme Parametrelerinin Belirlenmesi", **Makine Mühendisliği Dergisi**, Cild. 1, Sayı 9,1996.

Khasman, Adnan, “Neural networks for credit risk evaluation: Investigation of different neuralmodels and learning schemes,**Expert Systems with Applications**,2010.

Lee, Tian-Shyug ve Chen, I-Fei,“A two-stage hybrid credit scoring model using artificial neural networks and multivariate adaptive regression splines”, **Expert Systems with Applications**, 28,743–752,2005.

Lee, Tian-Shyug ve Chiu, Chih-Chou, Lu, Chi-Jie ve Chen, I-Fei,“ Credit scoring using the hybrid neural discriminant technique”, **Expert Systems with Applications**, 23,245–254,2002.

Lin, Shu Ling, ”A new two-stage hybrid approach of credit risk in banking industry”, **Expert Systems with Applications**, 36, 8333–8341, 2009.

Malhotra, Rashmi, Malhotra D.K., “Evaluating consumer loans using neural networks”,**Omega**, 31,83-96,2003.

Özkan, Murat Tolga, Gülesin, Mahmut, "Uzman Sistem Yaklaşımı İle Cıvata ve Dişli Çark Seçimi", **Turkish Journal of Engineering and Environmental Sciences**, 25, 2001.

Prechelt, Lutz,”Automatic early stopping using cross validation: quantifying the criteria”,**Pergamon Neural Networks**,11,761-767,1998.

Quin, Rui, Liu, Li Lie ve Xie, Jun, “An Application of Improved BP Neural Network in Personal Credit Scoring”, **2010 Second International Conference on Computer Modeling and Simulation**,2010.

Sustersic, Maja, Mramor, Dusan, Zupan, Jure,”Consumer credit scoring with limited data”, **Expert Systems with Applications**,36, 4736-4744,2009.

Tsai C.F. ve Wu, J.W., “Using neural network ensembles for bankruptcy prediction and credit scoring”,**Expert Systems with Applications**, 34, 2639–2649,2008.

YAO Jing Tao, TAN Chew Lim: “Guidelines for Financial Forecasting with Neural Networks”, Nisan 2001.

Yıldız, Birol,: “Finansal Başarısızlığın Öngörülmesinde YSA Kullanımı ve Halka Açık Sirketlerde Ampirik Bir Uygulama”, **İMKB Dergisi**, Cilt 5 Sayı:17, 2001.

Yu, L., Wang, Wang S., Lai K.K., “Credit risk assessment with a multistage neural network ensemble learning approach”, **Expert Systems with Applications**, 34, 1434–1444, 2008.

Yu, L., Wang, Wang S., Lai K.K., “An intelligent-agent-based fuzzy group decision making model for financial multicriteria decision support: The case of credit scoring”, **European Journal of Operational Research**, 195, 942–959, 2007.

DiĞER KAYNAKLAR

Baş, Nuray, “**Yapay Sinir Ağları Yaklaşımı ve Bir Uygulama**”, Mimar Sinan Ü. FBE İstatistik ABD Yüksek Lisans Tezi, 2006.

Bayır, Fırat, “**Yapay Sinir Ağları ve Tahmin Modellemesi Üzerine Bir Uygulama**”, İstanbul Ü. Sosyal Bilimler Enstitüsü İşletme Fak. Sayısal Yöntemler ABD Yüksek Lisans Tezi, 2006.

Burmaoğlu, Serhat, “**Birleşmiş Milletler Kalkınma Programı Beşeri Kalkınma Endeksi Verilerini Kullanarak Diskriminant Analizi, Lojistik Regresyon Analizi ve Yapay Sinir Ağlarının Sınıflandırma Başarılarının Değerlendirilmesi**”, Atatürk Üniversitesi Sosyal Bilimler Enstitüsü İşletme ABD Doktora Tezi, 2009.

Gülseçen, Sevinç., “**Yapay Sinir Ağları, İşletme Alanında Uygulanması ve Bir Örnek Çalışma**”, İstanbul Üniversitesi Sosyal Bilimler Enstitüsü İşletme Fakültesi Sayısal Yöntemler ABD Doktora Tezi, 1993.

Kocabaş, Şakir, “**Yapay Zekaya Giriş**”, Yayınlanmamış Ders Notu, 2006.

Lin, Yingchih, “**An Application of Artificial Neural Networks to Forecast Winning Prices: A Comparison of Back-Propagation Networks and Adaptive Network-Based Fuzzy Inference Systems**”, Lisans Tezi, Submitted in partial fulfillment of the requirements for the degree of Master of Science in Industrial and Systems Engineering in the Graduate School of Binghamton University State University of New York 2010, 2003.

Saraç, Tuğba, “**Yapay Sinir Ağları, Basılmamış Seminer Projesi**”, Gazi Üniversitesi Endüstri Mühendisliği Bölümü Anabilim Dalı Seminer Projesi, Ankara,2004.

Torun, Talip,”**Finansal Başarısızlık Tahmininde Geleneksel İstatistiki Yöntemlerle Yapay Sinir Ağlarının Karşılaştırılması ve Sanayi İşletmeleri Üzerinde Uygulama**”, Erciyes Üniversitesi Sosyal Bilimler Enstitüsü İşletme ABD Yüksek Lisans Tezi, 2007.

Ulupınar, Sultan Deniz, “ **2001 Kriz Dönemi, Öncesi ve Sonrasında Türk Ticari Bankalarının Karlılıklarının Lojistik Regresyon Analizi ile İncelenmesi**”, Marmara Üniversitesi İstatistik Bilim Dalı Yüksek Lisans Tezi, İstanbul, 2007.

Ürük, Ersan,”**İstatistiksel Uygulamalarda Lojistik Regresyon Analizi**”, Marmara Üniversitesi Fen Bilimleri Enstitüsü Matematik Anabilim Dalı Yüksek Lisans Tezi, 2007.

Yıldız, Ömür, “**Döviz Kuru Tahmininde Yapay Sinir Ağlarının Kullanımı**”, Eskişehir Osmangazi Üniversitesi Sosyal Bilimler İşlet ABD Yüksek Lisans Tezi, 2006.

Yurtoğlu, Hasan “**Yapay Sinir Ağları Metodolojisi ile Öngörü Modellemesi: Bazı Makroekonomik Değişkenler için Türkiye Örneği**”, Uzmanlık Tezi, DPT Uzmanlık Tezleri, Ekonomik Modeller ve Stratejik Araştırmalar Genel Müdürlüğü, 2005.

İNTERNET KAYNAKLARI

www.aaai.org, 2010

www.bilimteknik.com, Bilim Teknik Eylül 2003 Beyin Özel Sayısı.

Bencan, Nadir, “Yapay Beyin Değil, Yapay Zeka(2008)”,www.yapay-zeka.org, 2010

Erhan Altuntaş, Tuncay ÇELİK, Yapay Zeka (2003), (Çevrimiçi) <http://bagem.tripod.com/>,2010

Fröhlich, Jochen, “Neural Net Overview” (1997), <http://fbim.fh-regensburg.de/~saj39122/jfroehl/diplom/e-1.html>,2010.

Halıcı ,Uğur, Kayıhan,Yaman, “Yapay Zeka”, www.yapay-zeka.org,2010.

Şakir Kocabaş, “Yapay Zeka Araştırma ve Uygulama Alanları”, <http://www.sakirkocabas.com/>,2010, s.3.

EKLER

EK-1Değişkenler Arasındaki Spearman ve Kendall Tau-b Korelasyon Analizleri

Correlations																
	Arac_Tipi	Currencyno	İsDeneyimi	Sektor Deneyimi	Mekan Sahipligi	Moralite	Odeme Gecmisi	Telefon	BorçlulukOran	CariOran	ParaBirimi Uyumu	PayOrani	Belgelemlis MalVarligi	Kredi DegerlendirenGorusu	BayiGorusu	
Spearman's rho	Arac_Tipi	Correlation Coefficient	1,000	-,568**	-,123*	-,102**	-,035	-,072**	-,028	,051*	-,110**	-,050*	,466**	,049*	-,030	-,076**
		Sig. (2-tailed)	,000	,000	,000	,079	,000	,171	,011	,000	,013	,000	,014	,142	,000	,058
		N	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472
	Currencyno	Correlation Coefficient	-,568**	1,000	,089**	,105**	,060**	,056**	,039	-,039	,086**	-,008	-,831**	-,008	,064**	,060**
		Sig. (2-tailed)	,000	,000	,000	,003	,006	,055	,051	,000	,688	,000	,676	,001	,003	,023
		N	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472
	İsDeneyimi	Correlation Coefficient	-,123**	,089**	1,000	,698**	,152**	-,117**	,142**	-,084**	,445**	,333**	-,185**	,238**	,288**	,200**
		Sig. (2-tailed)	,000	,000	,000	,000	,000	,000	,000	,000	,000	,000	,000	,000	,000	,000
		N	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472
	SektorDeneyimi	Correlation Coefficient	-,102**	,105**	,698**	1,000	,158**	-,113**	,131**	-,053**	,445**	,316**	-,197**	,221**	,265**	,208**
		Sig. (2-tailed)	,000	,000	,000	,000	,000	,000	,008	,000	,000	,000	,000	,000	,000	,000
		N	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472
	MekanSahipligi	Correlation Coefficient	-,035	,060**	,152**	,158**	1,000	-,024	,103**	-,067**	,057**	,084**	-,051**	,092**	,184**	,074**
		Sig. (2-tailed)	,079	,003	,000	,000	,000	,224	,000	,001	,005	,000	,011	,000	,000	,044
		N	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472
	Moralite	Correlation Coefficient	-,072**	,056**	-,117**	-,113**	-,024	1,000	-,008	,002	-,055**	-,066**	-,023	-,019	-,057**	,012
		Sig. (2-tailed)	,000	,006	,000	,000	,224	,000	,706	,907	,006	,001	,250	,339	,005	,934
		N	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472
	OdemeGecmisi	Correlation Coefficient	-,028	,039	,142**	,131**	,103**	-,008	1,000	-,087**	,113**	,118**	-,052**	,132**	,119**	,034
		Sig. (2-tailed)	,171	,055	,000	,000	,000	,706	,000	,000	,000	,000	,010	,000	,000	,094
		N	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472
	Telefon	Correlation Coefficient	,051*	-,039	-,084**	-,053**	-,067**	,002	-,087**	1,000	-,035	-,015	,007	-,035	,013	-,006
		Sig. (2-tailed)	,011	,051	,000	,008	,001	,907	,000	,000	,084	,450	,724	,083	,511	,755
		N	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472
	BorçlulukOranı	Correlation Coefficient	-,110**	,086**	,445**	,445**	,057**	-,055**	,113**	-,035	1,000	,347**	-,205**	,205**	,396**	,395**
		Sig. (2-tailed)	,000	,000	,000	,000	,005	,006	,000	,084	,000	,000	,000	,000	,000	,000
		N	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472
	CariOran	Correlation Coefficient	-,050*	-,008	,333**	,316**	,084**	-,066**	,118**	-,015	,347**	1,000	-,036	,570**	,170**	,148**
		Sig. (2-tailed)	,013	,688	,000	,000	,000	,001	,000	,450	,000	,000	,075	,000	,000	,000
		N	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472
	ParaBirimiUyumu	Correlation Coefficient	,466**	-,831**	-,185**	-,197**	-,051**	-,023	-,052**	,007	-,205**	-,036	1,000	-,006	-,137**	-,106**
		Sig. (2-tailed)	,000	,000	,000	,000	,011	,250	,010	,724	,000	,075	,000	,781	,000	,000
		N	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472
	PayOrani	Correlation Coefficient	,049*	-,008	,238**	,221**	,092**	-,019	,132**	-,035	,205**	,570**	-,006	1,000	,108**	,071**
		Sig. (2-tailed)	,014	,676	,000	,000	,000	,339	,000	,083	,000	,000	,781	,000	,000	,012
		N	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472
	BelgelemlisMalVarligi	Correlation Coefficient	-,030	,064**	,288**	,265**	,184**	-,057**	,119**	,013	,396**	,170**	-,137**	,108**	1,000	,205**
		Sig. (2-tailed)	,142	,001	,000	,000	,000	,005	,000	,511	,000	,000	,000	,000	,000	,000
		N	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472
	KrediDegerlendirenGorusu	Correlation Coefficient	-,076**	,060**	,200**	,208**	,074**	,012	,134**	,010	,282**	,117**	-,106**	,071**	,205**	1,000
		Sig. (2-tailed)	,000	,003	,000	,000	,000	,548	,000	,625	,000	,000	,000	,000	,000	,000
		N	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472
	BayiGorusu	Correlation Coefficient	-,038	,046**	,226**	,186**	,041*	-,034	,034	-,006	,395**	,148**	-,106**	,051*	,509**	1,000
		Sig. (2-tailed)	,058	,023	,000	,000	,044	,090	,094	,755	,000	,000	,000	,012	,000	,000
		N	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472

** . Correlation is significant at the 0.01 level (2-tailed).

* . Correlation is significant at the 0.05 level (2-tailed).

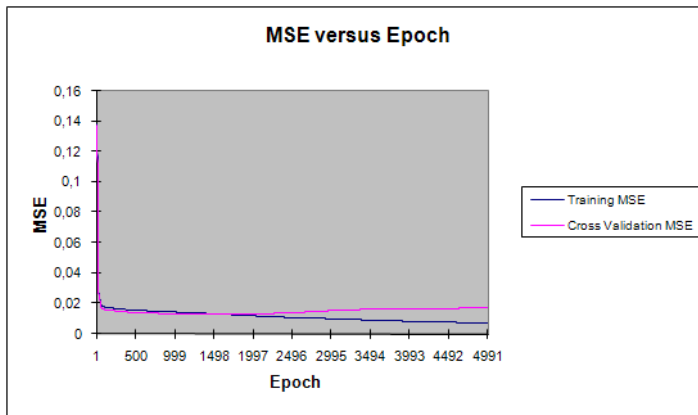
Correlations																
	PesinatOrani	Taksit	İsDeneyimi	Sektor Deneyimi	Mekan Sahipligi	Moralite	Odeme Gecmisi	Telefon	BorçlulukOran	CariOran	PayOrani	ParaBirimi Uyumu	Belgelemlis MalVarligi	Kredi DegerlendirenGorusu	BayiGorusu	
Kendall's tau_b	PesinatOrani	Correlation Coefficient	1,000	-,421**	,123*	,154**	-,029	,036	,040	,004	,215**	,019	,007	-,131**	,076**	,120**
		Sig. (2-tailed)	,000	,000	,000	,000	,072	,025	,016	,625	,000	,232	,862	,000	,000	,000
		N	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472
	Taksit	Correlation Coefficient	-,421**	1,000	-,171**	-,207**	-,016	-,015	-,050**	-,019	-,228*	-,080**	-,060**	,204*	-,101**	-,138**
		Sig. (2-tailed)	,000	,000	,000	,000	,348	,393	,005	,268	,000	,000	,000	,000	,000	,000
		N	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472	2472

EK-2 -Ara Katmandaki En İyi Nöron Sayısının Belirlenmesi

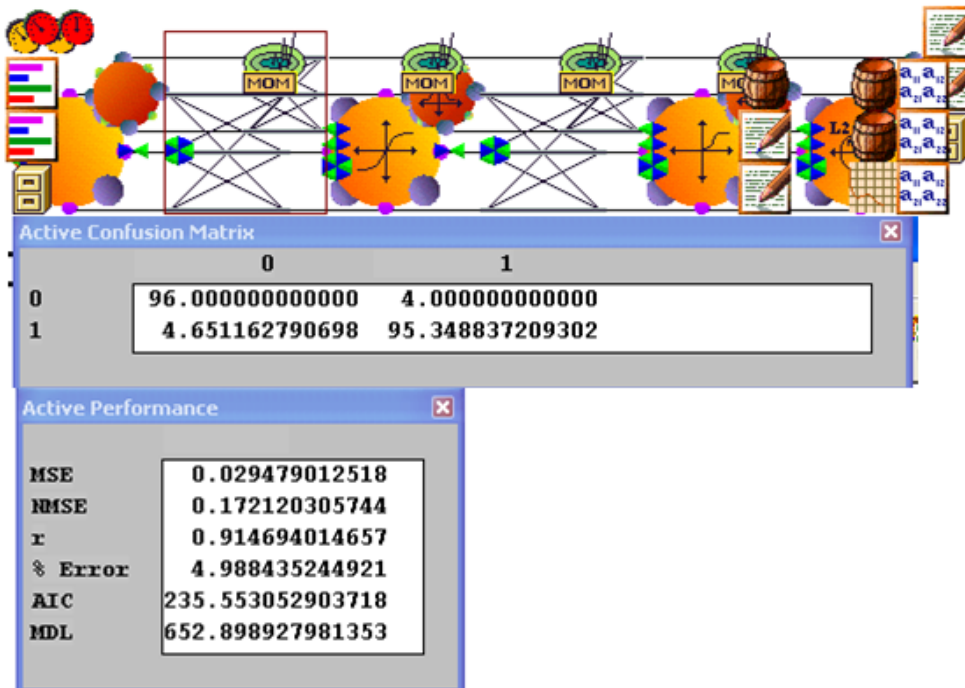
En İyi Sonuç olan 15-29-1 Ağ Yapısı Sonuçları (Ara Katman Sayısı:1, Öğrenme Oranı: 0,1, Momentum Katsayısı: 0,9, Aktivasyon Fonksiyonu: Hiperbolik Tanjant ve Sigmoid, Döngü: 5000)

	MSE	Tahmin	r
Deneme1	0,0319	0,9514	0,9102
Deneme2	0,0307	0,9514	0,9125
Deneme3	0,0295	0,9555	0,9147
Deneme4	0,0326	0,9474	0,9081
Deneme5	0,0306	0,9555	0,9126
Ortalama	0,0310	0,9522	0,9118

En iyi Sonuç olan, 3. Denemenin, üretmiş olduğu sonuçlar şöyledir:



Output / Desired	Decision(0)	Decision(1)
Decision(0)	72	8
Decision(1)	3	164

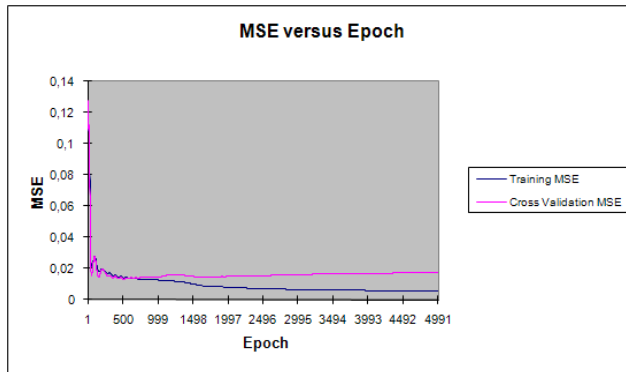


EK-3: En Uygun Öğrenme Oranı ve Momentum Katsayısının Belirlenmesi

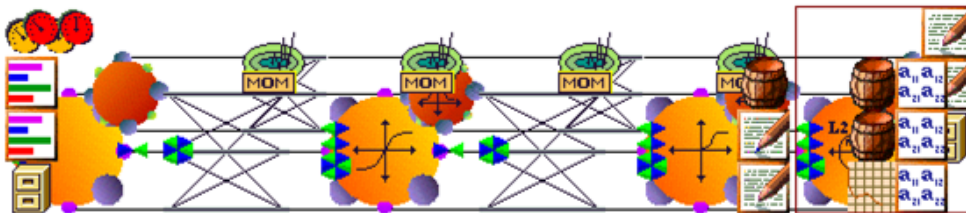
En iyi sonuç olan 15-29-1 ağ yapısı, 0,05 öğrenme oranı ve 0,99 momentum katsayısı denemeleri (Aktivasyon Fonksiyonu: Hiperbolik Tanjant ve Sigmoid, Döngü: 5000)

	MSE	Tahmin	r
Deneme1	0,0249	0,9636	0,9330
Deneme2	0,0272	0,9595	0,9259
Deneme3	0,0276	0,9595	0,9176
Deneme4	0,0268	0,9595	0,9255
Deneme5	0,0268	0,9676	0,9284
Ortalama	0,0267	0,9619	0,9261

En iyi sonuç olan, 1. Denemenin üretmiş olduğu sonuçlar şöyledir:



Output / Desired	Decision(0)	Decision(1)
Decision(0)	72	6
Decision(1)	3	166



Active Confusion Matrix	
	0 1
0	96.000000000000 4.000000000000
1	3.488372093023 96.511627906977

Active Performance	
MSE	0.024905403137
RMSE	0.145416187195
r	0.933532692480
% Error	4.182444218951
AIC	193.910385119962
MDL	611.256260197598