

**T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

İMALAT SEKTÖRÜNDE YAPAY SINIR AĞLARI UYGULAMASI

EMİNE SEVİNÇTEKİN

**YÜKSEK LİSANS TEZİ
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI
SİSTEM MÜHENDİSLİĞİ PROGRAMI**

**DANIŞMAN
DOÇ. DR. SEMİH ÖNÜT**

İSTANBUL, 2014

T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

İMALAT SEKTÖRÜNDE YAPAY SİNİR AĞLARI UYGULAMASI

Emine SEVİNÇTEKİN tarafından hazırlanan tez çalışması 26.03.2014 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı'nda

YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Tez Danışmanı

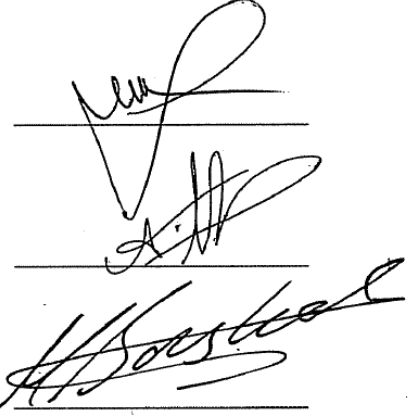
Doç. Dr. Semih ÖNÜT
Yıldız Teknik Üniversitesi

Jüri Üyeleri

Doç. Dr. Semih ÖNÜT
Yıldız Teknik Üniversitesi

Doç. Dr. Alev Taşkın GÜMÜŞ
Yıldız Teknik Üniversitesi

Yrd. Doç. Dr. Murat BASKAK
İstanbul Teknik Üniversitesi



ÖNSÖZ

Üretim planlama için gerçeğe yakın sonuçlar veren bir talep tahmini yapısı kurmak önemlidir. Trend, piyasanın durumu, mevsimsellik vb. birçok faktörden etkilenen talebin bunlarla arasında doğrusal veya doğrusal olmayan farklı bağlantılar vardır.

Yapay sinir ağları metodu, belirli değişkenlere bağlı bir problemin sonucunu, değişkenlerin problem üzerindeki etkisini ilk başta bilmeden sonra ise verilen örnekler yardımıyla öğrenen bir metottur. Öğrenme tamamlandığında ağ parametrelerin sonuç üzerindeki etkisini bildiğinden kendiliğinden doğru sonuç verir.

Bu tez çalışmasında talep tahmini problemi, çeşitli yapay sinir ağ yapıları kurularak minimum hatayı veren sinir ağıyla çözülmeye çalışılmıştır.

Tezin oluşturulmasında bana yol gösteren Prof. Dr. Hüseyin BAŞLIGİL'e, yardımlarını esirgemeyen tez hocam Doç. Dr. Semih ÖNÜT'e, Doç. Dr. Alev TAŞKIN GÜMÜŞ'e ve bugünlere gelmemde en fazla emeği olan anne ve babama teşekkür ederim.

Mart, 2014

Emine SEVİNÇTEKİN

İÇİNDEKİLER

	Sayfa
ÖNSÖZ.....	ii
İÇİNDEKİLER	iv
KISALTMA LİSTESİ.....	vi
ŞEKİL LİSTESİ.....	vii
ÇİZELGE LİSTESİ.....	viii
ÖZET	ix
ABSTRACT.....	x
BÖLÜM 1	
GİRİŞ.....	1
1.1 Literatür Özeti.....	2
1.2 Tezin Amacı.....	5
1.3 Hipotez	5
BÖLÜM 2	
YAPAY SİNİR AĞLARI.....	6
2.1 Yapay Sinir Ağları	6
2.1.1 Yapay Sinir Ağlarının Avantajları	9
2.2 Yapay Sinir Ağlarının Yapısı	11
2.2.1 İleri Beslemeli Yapay Sinir Ağları (Feed Forward Network)	15
2.2.2 Geri Beslemeli Yapay Sinir Ağları (Recurrent Networks, Feedback Networks).....	16
2.3 Yapay Sinir Ağlarının Eğitimi ve Öğrenmesi	17
2.3.1 Öğrenme Eğrisi	19
2.3.2 YSA'nın Öğrenme Yapısına Göre Sınıflandırılması	19
2.3.2.1 Danışmanlı Öğrenme (Supervised Training)	20
2.3.2.2 Danışmansız Öğrenme (Unsupervised Training)	21
2.3.2.3 Takviyeli Öğrenme (Reinforcement Training).....	22
2.3.3 Öğrenme Kuralları	23

2.3.3.1	Hebb Kuralı	24
2.3.3.2	Hopfield Kuralı	24
2.3.3.3	Kohonen Kuralı	24
2.3.3.4	Delta Kuralı	25
2.3.4	Uygulamaya Göre Öğrenme Algoritmaları	25
2.4	Yapay Sinir Ağlarının Tasarımı	26
2.4.1	YSA Ağ Yapısının Seçilmesi	26
2.4.2	Öğrenme Algoritmasının Seçilmesi	27
2.4.3	Ara Katman Sayısının Belirlenmesi	27
2.4.4	Nöron Sayısının Belirlenmesi	28
2.4.5	Normalizasyon	28
2.4.6	Performans Fonksiyonu Seçimi	29
2.5	Basit Algılayıcı Modeli (Perceptron)	29
2.6	Çok Katmanlı Algılayıcılar ÇKA (Multi Layer Perceptron MLP)	30
2.7	Radyal Tabanlı Sinir Ağı (Radyal Based Network RBN)	32
2.8	Hopfield Ağı	32
2.9	Öğrenme Vektör Parçalama Ağı (LVQ Learning Vector Quantisation)	33
2.10	Elman ve Jordan Ağları	35
2.11	MATLAB Programı (Matrix Laboratory)	36
2.11.1	Örnek Program 1	38
2.11.2	Örnek Program 2	39
BÖLÜM 3		
UYGULAMA		41
3.1	Uygulama Yapılan Firma Hakkında Bilgi	41
3.2	Farklı Yapıda MLP Yapay Sinir Ağlarının Kurulması	43
BÖLÜM 4		
SONUÇ VE ÖNERİLER		56
KAYNAKLAR		59
EK-A		62
YSA EĞİTİM SETİ		62
EK-B		71
YSA TEST SETİ		71
ÖZGEÇMİŞ		73

KISALTMA LİSTESİ

ANFIS	Adaptive Network Based Fuzzy Inference System
ANN	Artificial Neural Network
ARIMA	Auto-Regressive Integrated Moving Average
ÇKA	Çok Katmanlı Algılayıcı
LVQ	Learning Vector Quantisation
MAPE	Mean Absolute Percentage Error
MAE	Mean Absolute Error
MSE	Mean Squared Error
MLP	Multi Layer Perceptron
PSO	Particle Swarm Optimization
RBN	Radyal Based Network
XOR	eXclusive OR
YSA	Yapay Sinir Ağları

ŞEKİL LİSTESİ

	Sayfa
Şekil 2. 1	Biyolojik sinir hücresi7
Şekil 2. 2	Yapay sinir hücresi11
Şekil 2. 3	Çeşitli aktivasyon fonksiyonlarının gösterimi.....13
Şekil 2. 4	İleri beslemeli ve geri beslemeli çok katmanlı ağlar [24]15
Şekil 2. 5	YSA sınıflandırma20
Şekil 2. 6	Danışmanlı öğrenme21
Şekil 2. 7	Danışmansız öğrenme22
Şekil 2. 8	Takviyeli öğrenme23
Şekil 2. 9	Basit algılayıcı modeli30
Şekil 2. 10	Çok katmanlı algılayıcı31
Şekil 2. 11	Hopfield ağı33
Şekil 2. 12	LVQ öğrenme vektör parçalama ağı.....34
Şekil 2. 13	25 girişli 3 çıkışlı örnek MLP ağı.....38
Şekil 2. 14	XOR problemi MLP ağı.....39
Şekil 2. 15	Matlab NNTOOL penceresi40
Şekil 3. 1	Tasarlanan MLP ağ yapısı.....43
Şekil 3. 2	'logsig', 'purelin','trainlm','learngdm','mse' (3,9,1) ağı eğitim.....47
Şekil 3. 3	'logsig' 'purelin','trainlm','learngdm','mse' (3,9,1) ağı test48
Şekil 3. 4	'logsig' 'purelin','trainlm','learngdm','mse' (3,9,1) ağı öğrenme eğrisi.....51
Şekil 3. 5	'logsig', 'logsig' 'purelin','trainlm','learngdm','mse' (3,9,6,1) ağı eğitim.....53
Şekil 3. 6	'logsig' 'logsig' 'purelin','trainlm','learngdm','mse' (3,9,6,1) ağı test53
Şekil 3. 7	'logsig' 'logsig' 'purelin','trainlm','learngdm','mse'(3,9,6,1) ağı öğren. eğrisi.54

ÇİZELGE LİSTESİ

	Sayfa
Çizelge 1. 1 Ağ yapıları ve hata karşılaştırmaları [5].....	3
Çizelge 2. 1 Biyolojik sinir hücresi ile yapay sinir hücresinin karşılaştırılması.....	8
Çizelge 2. 2 XOR problemi.....	9
Çizelge 2. 3 Kullanılabilecek toplama fonksiyonları.....	12
Çizelge 2. 4 Ağ türleri ve başarılı oldukları alanlar[19].....	27
Çizelge 3. 1 Üretilen makine grupları.....	42
Çizelge 3. 2 Programa girişi yapılan örnek veri.....	44
Çizelge 3. 3 Gizli nöron sayısı için duyarlılık analizi.....	45
Çizelge 3. 4 Fonksiyon ve eğitim algoritması için duyarlılık analizi.....	46
Çizelge 3. 5 Ağın test sonuçları.....	49
Çizelge 3. 6 İkinci gizli katmandaki gizli nöron sayısı için duyarlılık analizi.....	52
Çizelge 3. 8 Bir girişli ve bir çıkışlı ağın gizli nöron sayısı için duyarlılık analizi...	55

İMALAT SEKTÖRÜNDE YAPAY SİNİR AĞLARI UYGULAMASI

Emine SEVİNÇTEKİN

Endüstri Mühendisliği Anabilim Dalı
Yüksek Lisans Tezi

Tez Danışmanı: Doç. Dr. Semih ÖNÜT

Gelecek her zaman belirsizdir ve bu belirsizlik gerek insanları gerek işletmeleri geleceği tahmin etmeye zorlar. Talep tahmini ile işletmeler bir sonraki dönem satışlarını tahmin etmeye ve buna göre tüm planlarını oluşturmaya çalışır. Gelecek dönem talepleri ne kadar gerçeğe yakın olursa işletme o kadar başarılı olur.

Yapay sinir ağları metodu insan beyni gibi çalışmaktadır. Beyin örnek bir olayda öğrendiği bilgilerin ışığında yeni kararlar vermekte, yeni tahminlerde bulunmaktadır. Yapay sinir ağları metodunda ağ öncelikle aynı beyin gibi eğitime tabi tutulur ve ağın öğrenmesi sağlanır. Öğrenen ağ artık kendi tahminlerini yapabilir. İş hayatında daha tecrübeli olan kişilerin daha doğru karar alıp daha doğru tahmin yapması gibi çok iyi kurulan bir yapay sinir ağı da gerçeğe yakın tahmin yapabilir.

Tez çalışmasında birinci bölümde yapay sinir ağları kullanılarak yapılan tahminlere yönelik literatür araştırması yapılmıştır; makalelerde YSA yapısının nasıl kurulduğu incelenmiştir. İkinci bölümde ise yapay sinir ağları detaylı bir şekilde anlatılmıştır.

Uygulama bölümünde ise makine imalatı yapan, satışları mevsime ve piyasaya göre çok değişkenlik gösteren bir firmanın talep tahmini YSA metodu ile gerçekleştirilmiştir. Uygulama MATLAB bilgisayar programında yapılmıştır. Oluşturulan yapay sinir ağında makine modeli, ay, ilgili ayda seçilen makinenin satış miktarı girdi değişkenleri ; bir sonraki sene ilgili makine modeli satışı ise çıktı değişkeni olarak tanımlanmıştır. Farklı ağ yapıları minimum hataya göre incelenmiştir.

Anahtar Kelimeler: Talep tahmini, yapay sinir ağları (YSA, ANN), çok katmanlı algılayıcı (ÇKA, MLP)

ABSTRACT

IMPLEMENTING OF ARTIFICIAL NEURAL NETWORK TO MANUFACTURING SECTOR

Emine SEVİNÇTEKİN

Department of Industrial Engineering

MSc. Thesis

Adviser: Assoc. Dr. Semih ÖNÜT

Future is always unknown. This makes people and company predict future. Companies try to forecast new period sales and prepare all its plans in related to new sales demand by demand forecasting. More correctly forecasted demand, more successful company.

Artificial neural network works as human brain. The brain takes new decisions, makes new predictions under learned information at sample events. At artificial neural network method firstly, network educated as brain and got learned. Yet this learned network can make predictions itself. As more experienced person can predict more correctly in life; well educated network can predict more truly.

In this thesis, at first section literature research made. On articles structure of artificial neural network investigated. At second section artificial neural network explained elaborately.

In practice section demand forecasting done for a company manufacturing machine, its sales changing with trend behavior and seasonality by artificial neural network. MATLAB programme used for this. At designed artificial neural network machine model, months, sales on related month accepted as inputs and coming year same month sales predictions accepted as output. Different network constructions investigated for minimum error.

Keywords: Demand forecasting, artificial neural networks (ANN), multi layer perceptron (MLP)

YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

GİRİŞ

İnsanlar dünyaya geldiklerinden itibaren bir yaşayarak öğrenme süreci içerisine girerler. Bu süreç içinde beyin yeni bilgiler öğrenir ve sürekli gelişir. Hayat devam ederken her geçen gün beyin farklı düşünceler öğrenir ve bunu hafızasına kaydeder. Beyin karar verirken öğrenmiş olduğu tüm bilgiler ışığında karar verir. Bir insanın, düşünme ve gözlemlleme yeteneklerini gerektiren problemlere yönelik çözümler üretebilmesinin temel sebebi insan beyninin ve dolayısıyla insanın sahip olduğu yaşayarak öğrenme yeteneğidir.

Beyin çalışma sistemi örnek alınarak geliştirilen yapay sinir ağları (YSA) bir bilgi işlem teknolojisidir. Normalde bir insanın düşünme ve gözlemlmeye yönelik doğal yeteneklerini gerektiren problemlere çözüm üretmektedir. Girdi ile çıktı arasındaki lineer olmayan ilişkiyi tespit etmede yapay sinir ağları kullanılmaktadır. Bunu da eski verileri analiz ederek gerçekleştirmektedir. Bu nedenle ağı eğitirken kullanılan verilerin doğruluğu ve miktarı gerçeğe yakın tahminlerde bulunabilen bir ağ tasarımı için çok önem taşır. Çünkü yeterli örneğe sahip olmayan bir yapay sinir ağı girdi ve çıktı arasında doğru bir ilişki oluşturamaz. YSA desen tanıma, veri sınıflama gibi özel problemlerin çözümünde kullanılabilmektedir.

Yapay sinir ağları; ekonomide, iktisatta, endüstri mühendisliğinde, otomasyonda, elektronik mühendisliğinde, elektronik devre tasarımında, bilgisayar mühendisliğinde, tıp alanında, çeşitli zeka problemlerinin çözümünde, optik algılama gibi birçok alanda kullanılmaktadır.

1.1 Literatür Özeti

Talep tahminlerine yönelik birçok araştırma bulunmaktadır. Son zamanlarda talep tahminlerinde yapay sinir ağları modelleri kullanılarak farklı yöntemler geliştirilmiştir. Yapılan literatür araştırmasında 2010 ve sonrası yayınlanan makaleler incelenmiştir. Bu makalelerde kurulan ağın girdileri, çıktıları ve ağın yapısı incelenmiştir.

Khashei ve Bijari [1]'de ARIMA otoregresiv hareketli ortalamayı kullanan yapay sinir ağı tasarlamışlardır. Çalışmalarında çeşitli denemeler yapmışlardır en iyi sonuç veren yapay sinir ağının 8-3-1 yapısında olduğunu söylemişlerdir. Çıktı katmanında çoğunlukla transfer fonksiyonu olarak lineer fonsiyon kullanılması gerektiği aksi takdirde YSA'nın yanlış veya bozuk tahmin verdiği belirtilmiştir.

Wang vd. [2]'de otomobil satışları için ANFIS (Bulanık Mantık Sistemine Dayalı Uyarlanabilir Ağ) kullanmışlardır. Bunun sonuçları ARIMA ve ANN ile karşılaştırılmıştır. ANFIS modelinin diğer yöntemlere göre daha iyi olduğu öne sürülmüştür.

Khah vd. [3]'te elektrik fiyat tahmini için dalga dönüşümü, ARIMA, radyal temele dayalı ANN içeren hibrit bir model tasarlamışlardır. Dalga dönüşümü ve ARIMA teknikleriyle elektrik fiyat tahmin probleminde lineer kısımlar tespit edilmiş, ANN ise diğer yöntemlerden elde edilen tahminlerdeki hataların düzeltilmesi amacıyla kullanılmıştır. Ağ yapısını optimize etmek için parça küme optimizasyonu (PSO) yapılmıştır.

Martins ve Werner [4]'te ARIMA ve ANN teknikleriyle tahminler yapmışlardır. Kombinasyon, aritmetik ortalama ve minimum varyans alınarak oluşturulmuştur. Daha sonra hatalar arasında korelasyonun hiç olmadığı ve pozitif korelasyon olduğu durumlara göre tekrar tahminler yapılmıştır. Hangi kombinasyonun daha iyi olduğunu belirlemek için doğruluk ölçütü MAPE, MSE, MAE seçilmiştir.

Yu vd. [5]'te Çin'in 2020 enerji taleplerini MPSO-RBF modeli ile tahmin etmişlerdir. 5-4-1 yapısında bir ağ tasarlanmıştır. Diğer ANN'lere göre daha az gizli nöron kullanılmış ve böylece daha az tahmin hatası oluşmuştur. Yu vd. tarafından tasarlanan model, Çizelge 1.1'de diğer yazarlarınkiyle karşılaştırılmaktadır [5].

Çizelge 1. 1 Ağ yapıları ve hata karşılaştırmaları [5]

Table 1

The comparison of structure and MAPE^a in the present study with other authors.

Authors	ANN structure	Training algorithms	Target/country	MAPE (%)
Limanond et al. [8]	Four layers MLP (5-5-5-1)	Back-propagation	Transport energy-Thailand	0.0263 ^b
Ekonomou [20]	Four layers MLP (4-20-17-1)	Levenberg-Marquardt	Total energy-Greece	3.52
Azadeh et al. [24]	Four layers MLP (5-3-3-1)	Resilient propagation and momentum and weight decay	Industrial sectors-Iran	0.99
Sözen and Arcaklioglu [32]	Three layers BP (2-4-1)	Levenberg-Marquardt	Net electricity energy-Turkey	0.93
Pao [34]	Three layers BP (2-5-1)	Back-propagation and momentum	Electricity energy-Taiwan	2.56
Coşkun [37]	Three layers BP (4-4-4)	Back-propagation and momentum	Four-sector net electricity energy-Turkey	8.17
Murat and Ceylan [40]	Three layers BP (3-14-1)	Back-propagation	Transport energy-Turkey	13.01
The present study	Three layers RBF (5-4-1)	MPSO-BP algorithm	Total energy-China	0.78/ 0.025 ^b

^a Average relative to the test period of each model.

^b Normalized root mean square error (NRMSE) of the test data.

Marvuglia ve Messineo [6]'da İtalya'da evlerde bir saat sonraki elektrik tüketimini radyal temele dayalı yapay sinir ağı kullanarak tahmin etmişlerdir. Ayrıca bu çalışmada klima kullanımının elektrik tüketimi üzerindeki etkisi incelenmiştir. Elektrik tüketimi ile ilgili hava durumu verileri girdi kabul edilmiştir. Buna ilaveten klima olup olmaması da bir girdi değişkeni olarak modele eklenmiştir.

Teixeira ve Fernandes [7]'de ANN ile turizm sektöründe talep tahmini yapmışlardır. Oluşturulan ağda yıl, ay ve önceki 12 ay gelen turist sayısı girdi değişkeni olarak tanımlanmıştır. Aynı sayıda girdi, çıktı ve gizli nörona sahip ileri beslemeli, geri beslemeli ve kademeli ağ yapısı oluşturulup hatalar karşılaştırılmıştır. Hata % 4 ve % 6 arasında çıkmıştır. Ancak ileri beslemeli ağın diğerlerine nazaran daha iyi sonuç verdiği gözlemlenmiştir.

Akınar ve Akkoyunlu [8]'de Düzce'deki Melen Çayı'nın akış oranını ANN ile tahmin etmişlerdir. 5 değişkenli 12 örnekle bir model oluşturulmuş. Daha sonra da burdan edinilen bilgiler ışığında SWOT analizi yapılmıştır.

Efendigil ve Önüt [9]'da ANFIS modeli ve ANN'nin yer aldığı kombine bir model yapmışlardır. Müşteri talep tahmininde ANFIS modeli; fabrika ve depo taleplerinde ise ANN kullanılmış. ANN'de MLP ve RBF modelini oluşturmuşlar. MLP olumlu sonuç vermiş. RBF (Radyal Temele Dayalı Fonksiyon) olumsuz sonuç vermiştir. 6-13-6 yapısında bir ağ oluşturulmuştur. MLP'nin doğruluğu istatistiksel verilerle yapılan tahminlerle kıyaslanarak kontrol edilmiştir.

Lau vd. [10]'da talep tahmininde 1'den 20'ye değişen nöronlara sahip çeşitli yapay sinir ağları oluşturup bunların arasından optimum ağı seçmek için minimum tanımlama uzaklığı şeklinde bir kriter oluşturmuşlardır.

Kulkarni vd. [11]'de enerji sistemlerinde kısa dönem yük tahmini için SNN ağı tasarlamış ve diğer ağlarla karşılaştırmışlardır.

Rodger [12]'de doğalgaz talep tahmini için 6 girdisi bir çıktısı olan ağ oluşturmuştur. Gerçeğe yakın sonuçların çıkması için gizli katmandaki nöron sayısının önemli olduğunu belirtmiş ve kendisi modelinde 20 gizli nöron kullanmıştır. Tasarlanan sistem bulanık mantık, regresyon ve ağa en yakın komşu yöntemiyle karşılaştırılmıştır.

Jaipuria ve Mahapatra [13]'te ANN modeline Gizli Dalga Dönüşümü (Discrete Wavelet Transform) ekleyerek DWT-ANN şeklinde yeni bir model oluşturmuşlardır. Bu modelle tedarik zincirinde kamçı etkisi azaltılarak talep tahmini yapılmıştır. Model, 3 farklı üretim firmasından elde edilen verilerle ayrıca test edilmiştir.

Kialashaki ve Reisel [14]'te evlerde tüketilen enerji talep tahmini için regresyon modelleri ve yapay sinir ağlarını içeren karma 6 değişkenli bir model oluşturmuşlardır.

Belayneh vd. [15]'te uzun dönem kuraklık tahminini ARIMA modeli kullanarak yapmışlardır. Bu modeli ANN ve yardımcı vektör regresyonu ile karşılaştırmışlardır.

Kheirkhah vd. [16] 'da ANN-PCA-DEA-ANOVA modeli kullanmışlardır. Parça Analizi (Principal Component Analysis PCA), Veri Zarflama Analizi (Data Envelopment Analysis DEA). ANN modellerinde girdi değişkenlerini seçmek için PCA yöntemi kullanılmış. Deneme ve yanılmanın zorluğu ve yanlış olabileceği göz önüne alındığında PCA yaklaşımının öneminin anlaşılabileceği belirtilmiştir.

Taşpınar vd. [17]'de Türkiye'de bölgesel bazda doğalgaz tüketimi tahmini yapmışlardır. ANN ve Zaman Serilerini kombine kullanan bir model tasarlanmıştır. SARIMAX (Seasonal form of ARIMA), ANN-MLP, ANN-RBF ve OLS modeline göre daha iyi sonuç vermiştir. ANN-MLP modeli 5-8-1 yapısında ANN-RBF modeli ise 5-9-1 yapısında oluşturulmuştur. 8 gizli nöronlu ağların daha iyi performans gösterdiği belirtilmiştir.

Claveria ve Torra [18]'de ANN ve zaman serilerini kombine kullanan bir model kullanmışlardır. ARIMA model SETAR ve ANN modellerine göre daha iyi sonuç vermiştir.

1.2 Tezin Amacı

Makine imalatı yapan bir firmanın gelecek dönem satılacak makine adedi yapay sinir ağıları metoduyla bulunmaya çalışılmıştır. Böylece yöneticilerin sadece geçen seneki satışlara göre yeni seneyi tahmin etmelerini önleyerek geçmiş tüm satış verileri ışığında daha doğru tahminlerin yapılması amacı güdülmüştür.

1.3 Hipotez

Satışları mevsime, giyim trendine, piyasaların ekonomik durumuna göre çok değişkenlik gösteren bir makine imalat firmasının nihai ürün talebini tahmin etmede yapay sinir ağlarını kullanması, elde edilen tahminlerin doğruluğunu arttıracaktır.

YAPAY SİNİR AĞLARI

2.1 Yapay Sinir Ağları

Yapay sinir ağları; insan beyninin özelliklerinden olan öğrenme yolu ile yeni bilgiler türetebilme, yeni bilgiler oluşturabilme ve keşfedebilme gibi yetenekleri herhangi bir yardım almadan otomatik olarak gerçekleştirmek amacı ile geliştirilen bilgisayar sistemleridir. Bu yetenekleri geleneksel programlama yöntemleri ile gerçekleştirmek oldukça zor veya mümkün değildir. Bu nedenle yapay sinir ağlarının, programlanması çok zor veya mümkün olmayan olaylar için geliştirilmiş adaptif bilgi işleme ile ilgilenen bir bilgisayar bilim dalı olduğu söylenebilir [19].

Yapay sinir ağı; insan beyninin sinir hücrelerinden oluşmuş katmanlı ve paralel olan yapısının tüm fonksiyonlarıyla beraber sayısal dünyada gerçeklenmeye çalışılan modellenmesidir. Sayısal dünya ile belirtilmek istenen donanım ve yazılımdır.

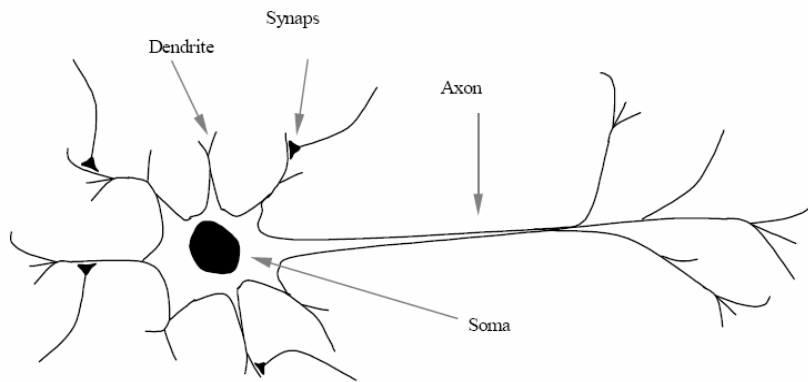
Yapay sinir ağları; olayların örneklerine bakmakta, onlardan ilgili olay hakkında genellemeler yapmakta, bilgiler toplamakta ve daha sonra hiç görmediği örnekler ile karşılaşınca öğrendiği bilgileri kullanarak o örnekler hakkında karar verebilmektedir.

Biyolojik bir sinir hücresi Şekil 2.1’de görüldüğü üzere sinaps, axon, soma ve dendritlerden oluşmaktadır.

- **Dendrit:** Görevi diğer sinir hücrelerinden iletilen sinyalleri, sinir hücresinin çekirdeğine iletmektedir. Bu yapı basit gibi görünse de günümüzde dendritlerin görevlerinin daha karmaşık olduğu yolunda söylemler hâkim olan görüştür. Hücrenin çekirdeği ile her bir dendrit arasında farklı bir iletişim söz konusudur.

- **Soma:** Hücreyi denetlemeyle ve yönetmekle görevlidir. Bu görevi gerçekleştirmek için gelen sinyalleri işleme alır ve hücrenin kendi elektrik sinyalini oluşturur. Dendritler yoluyla iletilen tüm sinyalleri alıp toplayan merkezdir. Biyolojide hücre çekirdeği (nükleus) olarak da bilinmektedir. Çekirdek gelen toplam sinyali diğer sinir hücrelerine göndermek üzere, bilgiyi aksona iletir.
- **Akson:** Hücre çekirdeğinden aldığı toplam bilgiyi bir sonraki sinir hücresine dağıtmakla görevlidir. Ancak akson bu toplam sinyalin ön işleminden geçirilmeden diğer sinir hücresine aktarılmasına engel olur. Çünkü akson ucunda sinaps denilen birimlere bilgiyi aktarır. Burda ön işleminden geçer.
- **Sinaps:** Aksondan gelen toplam bilgiyi ön işleminden geçirdikten sonra diğer sinir hücrelerinin dendritlerine iletmekle görevlidir. Sinapsın ön işlem ile gerçekleştirdiği görev çok önem taşımaktadır. Bu ön işlem gelen toplam sinyalin, belli bir eşik değerine göre değiştirilmesinden ibarettir. Böylece toplam sinyal olduğu gibi değil, belli bir aralığa indirgenerek diğer sinir hücrelerine iletilmiş olur. Bu açıdan, her gelen toplam sinyal ile dendrite iletilen sinyal arasında bir korelasyon (ilişki) oluşturulur. Buradan yola çıkılarak “öğrenme” işleminin sinapslarda gerçekleştiği fikri ortaya atılmış ve bu hipotez, günümüz yapay sinir ağı dünyası için teori haline dönüşmüştür. Yapay sinir ağı modelleri üzerinde “öğrenme” bu teoriye dayanılarak, sinapslar ve dendritler arasında yer alan ağırlık katsayılarının güncellenmesi olarak algılanmaktadır.

Milyarlarca sinir hücresi bir araya gelerek insan sinir sistemini oluşturmaktadır. İnsan beyinde yaklaşık 10 milyar sinir hücresi ve 60 trilyon sinaps bağlantısı bulunmaktadır.



Şekil 2. 1 Biyolojik sinir hücresi

İnsan beyinde öğrenme 3 şekilde olur:

- Yeni aksonlar üreterek
- Aksonların uyarılmasıyla
- Mevcut aksonların güçlerini değiştirerek

Her aksonun, üzerinden geçen işaretleri değerlendirebilecek yetenekte olduğu savunulmaktadır. Aksonun bu özelliği, bir işaretin belli bir sinir için ne kadar önemli olduğunu göstermektedir.

Biyolojik sistemlerde öğrenme, nöronlar arasındaki sinaps bağlantılarının ayarlanması ile olur. Bu durum YSA için de geçerlidir. Öğrenme, giriş ve çıkış verilerinin işlenmesiyle, yani eğitme algoritmasının bu verileri kullanarak bağlantı ağırlıklarını bir yakınsama sağlanana kadar, tekrar tekrar ayarlamasıyla olur.

Çizelge 2. 1 Biyolojik sinir hücresi ile yapay sinir hücresinin karşılaştırılması

Biyolojik Sinir Hücresi	Yapay Sinir Hücresi
Sinir sistemi	Hesaplama sistemi
Sinir	Düğüm, İşlem elemanı, Sinir
Sinaps	Bağlantı ağırlıkları
Dentrit	Toplama fonksiyonu
Hücre gövdesi	Etkinlik (aktivasyon) fonksiyonu
Akson	Sinir çıkışı

Yapay sinir ağlarının tarihsel gelişimine bakıldığında 1970'e kadar çalışmaların yavaşladığı görülmüştür. Bunun nedeni Çizelge 2.2'de gösterilen XOR probleminin YSA ile çözülmemesidir. 1970'te çok katmanlı ağ keşfedilince XOR problemi YSA ile çözüldü ve çalışmalar tekrar hız kazandı. XOR problemini özetlemek gerekirse 2 girdisi bir çıktısı olan bir problemdir. Girdilerin işaretleri birbirinden farklı olduğu zaman çıkış olarak bir, diğer durumlarda sıfır vermektedir. 1970'lerden önce sadece tek katmanlı YSA bulunmuştu. Ancak XOR problemi çok katmanlı algılayıcı gerektirdiğinden dolayı tek katmanlı ağ ile çözülmemiştir.

Çizelge 2. 2 XOR problemi

	1.Girdi (X1)	2.Girdi (X2)	Çıkış (Y)
1.Örnek	0	0	0
2.Örnek	0	1	1
3.Örnek	1	0	1
4.Örnek	1	1	0

2.1.1 Yapay Sinir Ağlarının Avantajları

- **Paralel Çalışma:** YSA'da tüm işlem elemanları eş zamanlı çalıştıkları için çıktı üretmesi çok hızlı olur. Paralel bir sistemde yavaş bir birimin etkisi çok azdır.
- **Doğrusal Olmama:** YSA'nın temel işlem birimi olan hücre doğrusal olmadığından hücrelerin birleşmesinden meydana gelen YSA da doğrusal değildir. Bu özellik, tüm ağı yayılmış durumdadır ve bu özelliği ile YSA, doğrusal olmayan karmaşık problemlere çözüm getirmektedir.
- **Genelleme:** YSA, problemi öğrendikten sonra eğitim sırasında karşılaşmadığı örnekler için de ilgili tepkiyi üretme yeteneğine sahiptir. Örneğin, karakter tanıma amacıyla eğitilmiş bir YSA, bozuk karakter girişlerinde de doğru sonucu verir. Eğitilmiş bir ağı, girişin tamamı değil sadece bir kısmı verilse bile ağı, hafızadan bu girişe en yakın olanını seçerek tam bir giriş verisi alıyormuş gibi kabul eder ve buna uygun bir çıkış değeri üretir. YSA'ya, eksik, bozuk veya daha önce hiç karşılaşmadığı şekilde veri girilse bile, ağı kabul edilebilir en uygun çıkışı üretecektir. Bu özellik ağı genelleştirme özelliğidir [20].
- **Öğrenme:** YSA'nın arzu edilen davranışı gösterebilmesi için amaca uygun olarak ayarlanması gerekir. Bu, hücreler arasında doğru bağlantıların yapılması ve bağlantıların uygun ağırlıklara sahip olması demektir. YSA'nın karmaşık yapısı nedeniyle bağlantılar ve ağırlıklar önceden ayarlı olarak verilemez. Bu nedenle YSA, istenen davranışı gösterecek şekilde ilgilendiği problemten aldığı eğitim örneklerini kullanarak problemi öğrenmelidir.

- **Hata Toleransı:** YSA, çok sayıda hücrenin çeşitli şekillerde bağlanması ile oluştuğundan paralel dağılmış bir yapıya sahiptir ve ağına sahip olduğu bilgi, ağıdaki bütün bağlantılar üzerine dağılmış durumdadır. Bu nedenle, eğitilmiş bir YSA' nın bazı bağlantılarının hatta bazı hücrelerinin etkisiz hale gelmesi, ağına doğru bilgi üretmesini önemli ölçüde etkilemez. Bu nedenle, geleneksel yöntemlere göre hatayı tolere etme yetenekleri son derece yüksektir.
- **Donanım ve Hız:** YSA, paralel yapısı nedeniyle büyük ölçekli bütünleşmiş devre (VLSI) teknolojisi ile tasarlanabilir. Bu özellik, YSA' nın hızlı bilgi işleme yeteneğini artırır ve gerçek zamanlı uygulamalarda tercih edilmesini sağlar.
- **Analiz ve Tasarım Kolaylığı:** YSA' nın temel işlem elemanı olan hücrenin yapısı ve modeli, bütün YSA yapılarında yaklaşık aynıdır. Dolayısıyla, YSA' nın farklı uygulama alanlarındaki yapıları da standart yapıdaki bu hücrelerden oluşacaktır. Bu nedenle, farklı uygulama alanlarında kullanılan YSA' lar benzer öğrenme algoritmalarını ve teorilerini paylaşabilirler. Bu özellik, problemlerin YSA ile çözümünde önemli bir kolaylık getirecektir [21].
- **Kendi İlişkisini Oluşturma:** Yapay sinir ağları verilere göre kendi ilişkilerini kendisi oluşturabilir. Bünyesinde sabit bir denklem içermez.
- **Sınırsız Sayıda Değişken ve Parametre Kullanımı:** Yapay sinir ağları sınırsız sayıda değişken ve parametre ile çalışabilir. Bu sayede çok başarılı bir tahmin ve genel çözümler üretebilmektedir.
- **Algılamaya Yönelik Olaylarda Kullanılabilirlik:** Yapay sinir ağları daha çok algılamaya dönük bilgileri işlemede kullanılır. Bilgiye dayalı işlemlerde ise genellikle uzman sistemler kullanılır. Bazı durumlarda bu iki sistem birleştirilerek daha başarılı sonuçlar üreten bir sistem elde edilebilir.
- **Dereceli Bozulma:** Hatalara karşı toleranslı oldukları için sistemin bozulması da dereceli olur. Yani klasik programlarda sistemde bir hata var ise sistem tamamen çalışamaz duruma geçer, sistem yorum yapamayacağı için kısmi de olsa bilgi üretemez. Fakat YSA' lar eldeki verilerle, sağlam olan hücrelerle bilgi üretmeye çalışırlar [22].

YSA birçok avantaja sahip olmakla birlikte iki önemli kısıt da içerir. Bunlardan ilki modeldeki parametrelerin açıklanmasının çok güç olmasıdır. Tasarlanan bir YSA, ortaya çıkan doğrusal olmayan örüntüleri tasvir etmekte hangi tür parametrik modelin daha uygun olabileceği konusunda bilgi sunmaz.

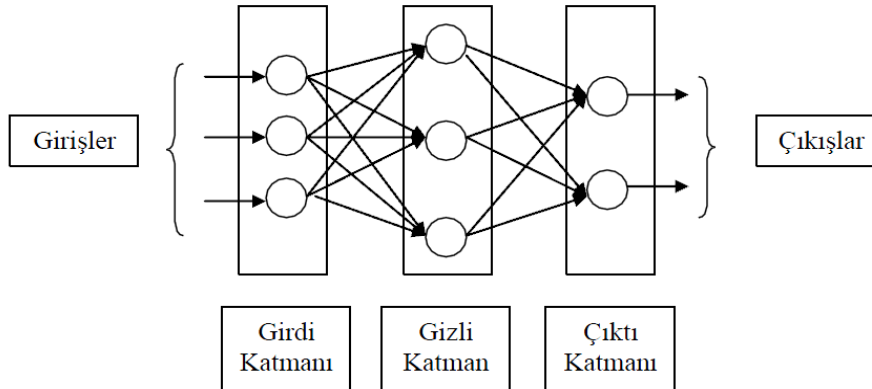
Parametre değerlerine anlam yüklemek genellikle güçtür. Bu yüzden de YSA'lar sık sık “kara kutu modelleri” olarak bilinir.

YSA'ların bir diğer kısıtı ise aşırı ezberleme tehlikesidir. Modelin esnekliğini artırmak suretiyle hemen hemen mükemmel bir örneklem içi uyum elde etmek mümkündür; ancak böyle bir durumda model örneklem dışı uyum için daha az yararlı olacaktır.

Yapay sinir ağları ile ilgili süreçler, YSA yaklaşımının geleneksel bilgi işleme tekniklerinden farklı olduğunu gösterir. Geleneksel tekniklerde problemin çözüm yöntemi bilinir ve problem algoritmik olarak tanımlanabilir. Sinir ağları ise problemi modellemeye çalışmak yerine verilerden ve hatalardan sürekli öğrenir. Öğrendikçe ağırlıkları uyarlayarak performansını (bir hata kriterine göre hesaplanan çıktı ile gerçekleşen çıktı arasındaki toplam fark) optimal düzeye çıkarmaya çalışır [23].

2.2 Yapay Sinir Ağlarının Yapısı

Yapay sinir hücreleri biyolojik hücreden esinlenerek ortaya çıkmıştır. Şekil 2.2’de bir yapay sinir hücresi görülmektedir. Yapay sinir ağı 3 katmandan oluşmaktadır. Girdi katmanı, bilginin işlenmediği sadece gelen bilgilerin ağı iletildiği katmandır. Gizli (ara) katman, girdi katmanından gelen bilgilerin işlendiği ve çıkış katmanına gönderildiği katmandır. Bir ağda birden fazla gizli katman olabilir. Çıktı katmanı ise gizli katmandan gelen bilgiyi işleyerek dış dünyaya gönderen katmandır.



Şekil 2. 2 Yapay sinir hücresi

Bir yapay sinir hücresinin 5 temel elemanı bulunmaktadır:

- **Girdiler:** Girdiler ($u_1, u_2, \dots, u_n$) çevreden aldığı bilgileri sinire getirir. Bu bilgiler dış dünyadan olabileceği gibi başka hücrelerden ya da hücrenin kendisinden gelebilir.
- **Ağırlıklar:** Ağırlıklar ($w_1, w_2, \dots, w_n$) hücreye gelen bilginin hücre üzerindeki etkisini gösterir. Ağırlık değerinin büyük ya da küçük olması, onların önemini belirleyen bir unsur değildir. Ağırlığın artı veya eksi olması o girdinin ağa etkisinin pozitif veya negatif olduğunu gösterir. Belirli bir özellik önemli ise ağda onun gösterimi ile görevlendirilen hücre sayısı daha fazla olmalı.
- **Toplama Fonksiyonu:** Hücreye gelen net girdiyi hesaplayarak, çıkan sonucu etkinlik (aktivasyon) fonksiyonuna gönderir. Net girdinin hesaplanması için değişik fonksiyonlar kullanılabilir. En çok, hücreye gelen girdi değerleri kendi ağırlıkları ile çarpılarak bulunan ağırlıklı toplam kullanılır. Belirli bir problem için en uygun toplama fonksiyonunun belirlenmesi amacıyla kullanılacak herhangi bir yöntem bulunmamaktadır. Bazen ağırlıklı toplam gibi basit işlem yerine Çizelge 2.3'te görüldüğü üzere minimum, maksimum, çarpım ya da kümülatif toplam algoritmaları da kullanılabilir. Ağda bulunan her hücrenin farklı toplama fonksiyonuna sahip olması mümkün olduğu gibi, ağdaki hücrelerden bir kısmı grup olarak aynı toplama fonksiyonuna sahip olabilir.

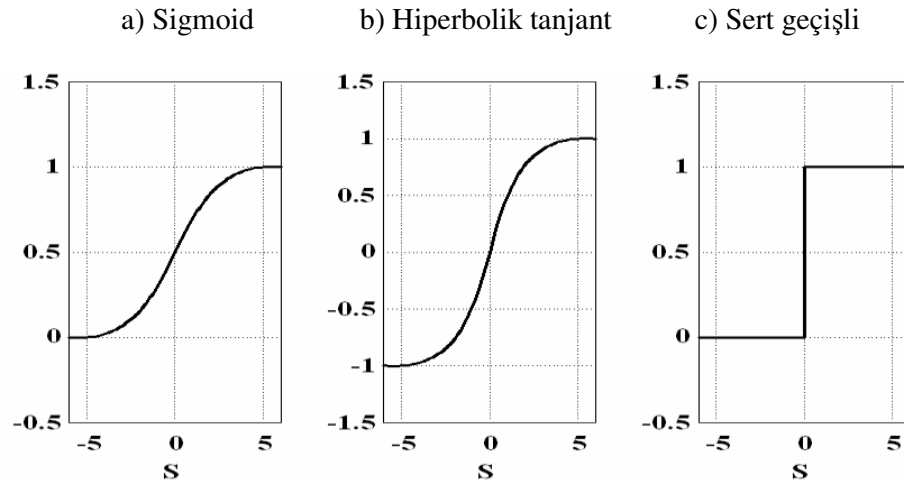
Çizelge 2. 3 Kullanılabilecek toplama fonksiyonları

Fonksiyonun Adı	Fonksiyon
Toplam	$Net\ Girdi = \sum_i G_i W_i$
Çarpım	$Net\ Girdi = \prod_i G_i W_i$
Maksimum	$Net\ Girdi = maks(G_i W_i)$
Minimum	$Net\ Girdi = \sum_i min(G_i W_i)$
Çoğunluk (Signum)	$Net\ Girdi = \sum_i sgn(G_i W_i)$
Kümülatif Toplam	$Net\ Girdi = Net_{eski} + \sum_i G_i W_i$

- **Aktivasyon (Etkinlik) Fonksiyonu:** Toplama fonksiyonundan alınan sonuç aktivasyon fonksiyonundan geçirilerek çıkışa verilir. Aktivasyon fonksiyonuna transfer fonksiyonu da denmektedir ve genellikle doğrusal olmayan bir fonksiyondur. Yapay sinir ağlarında ağın yapısına göre değişik aktivasyon fonksiyonları kullanılabilir. Bir problemin çözülüp kullanılacak aktivasyon fonksiyonunun belirlenebilmesi için herhangi bir yöntem bulunmamaktadır. Aktivasyon fonksiyonu YSA'da nöronun çıkış genişliğini istenilen değerler arasında sınırlar; genellikle net girdiyi $[0,1]$ veya $[-1,1]$ tanımlar.

Aktivasyon fonksiyonunun seçimi büyük ölçüde yapay sinir ağının verilerine ve ağın neyi öğrenmesinin istendiğine bağlıdır. Aktivasyon fonksiyonları içinde en çok kullanılan sigmoid ve hiperbolik tanjant fonksiyonlarıdır. Örneğin, eğer ağın bir modelin davranışını öğrenmesi isteniyorsa sigmoid fonksiyon, ortalamadan sapmasını öğrenmesi isteniyorsa hiperbolik tanjant fonksiyon kullanılmalıdır.

YSA'da kullanılacak aktivasyon fonksiyonlarının türevi alınabilir olması ve süreklilik arz etmesi gereklidir. Lineer veya doğrusal olmayan transfer fonksiyonlarının kullanılması YSA'ların karmaşık ve çok farklı problemlere uygulanmasını sağlamıştır. Şekil 2.3'te çeşitli aktivasyon fonksiyonlarına yer verilmiştir.



Şekil 2.3 Çeşitli aktivasyon fonksiyonlarının gösterimi

2.1, 2.2 ve 2.3 denklemleri sırasıyla sigmoid, hiperbolik tanjant ve sert geçişli aktivasyon fonksiyonlarına aittir.

$$\Psi_1(S) = \frac{1}{1+e^{-S}} \quad \text{Sigmoid tipi aktivasyon fonksiyonu} \quad (2.1)$$

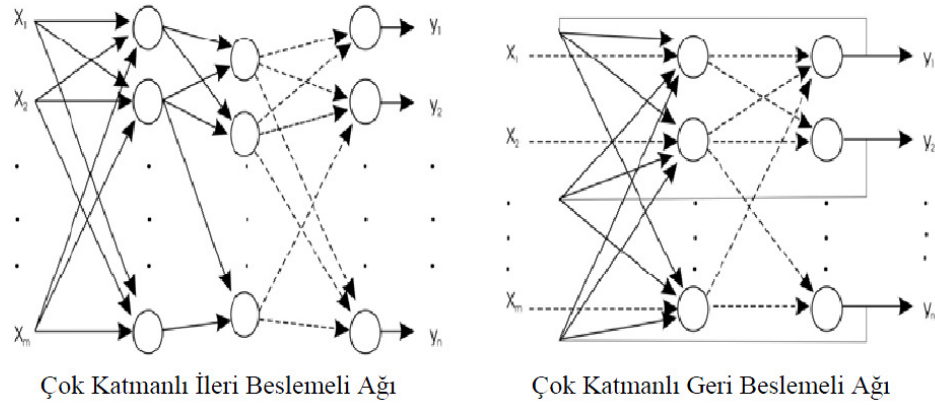
$$\Psi_2(S) = \frac{e^S - e^{-S}}{e^S + e^{-S}} = \tanh(S) \quad \text{Hiperbolik tanjant tipi aktivasyon fonksiyonu} \quad (2.2)$$

$$\Psi_3(S) = \begin{cases} 0 & S \leq 0 \\ 1 & S > 0 \end{cases} \quad \text{Sert geçişli tipte aktivasyon fonksiyonu} \quad (2.3)$$

- **Çıktı:** Aktivasyon fonksiyonu tarafından belirlenen değerdir. Bu değer birden fazla hücreye girdi olarak gönderilir. Ancak her bir hücrenin tek bir çıkış değeri vardır.

YSA hesaplamalarının merkezinde dağıtılmış, adaptif ve doğrusal olmayan işlem kavramları vardır. Geleneksel işlemcilerde, tek bir merkezi işlem birimi her hareketi sırasıyla gerçekleştirir. YSA'da ise her biri büyük bir problemin bir parçası ile ilgilenen, çok sayıda basit işlem birimleri bulunmaktadır. Bir nöron, girdiyi bir ağırlık kümesi ile ağırlıklandırır, doğrusal olmayan bir şekilde dönüşümünü sağlar ve bir çıktı değeri oluşturur. İlk bakışta, nöronların çalışma şekli basittir. Ancak YSA'ların gücü, toplam işlem yükünü paylaşan nöronların birbirleri arasındaki yoğun bağlantı yapısından gelmektedir. Bu da YSA yapısını karmaşık hale getirmektedir.

Yapay sinir ağları katman sayısı ve katmanlar arası bağlantı yapısına göre sınıflandırılabilir. Yapay sinir ağları, katmanlar arasındaki bağlantı yapısına göre Şekil 2.4'te görüldüğü üzere ileri beslemeli ve geri beslemeli olmak üzere ikiye ayrılmaktadır.



Şekil 2. 4 İleri beslemeli ve geri beslemeli çok katmanlı ağlar [24]

2.2.1 İleri Beslemeli Yapay Sinir Ağları (Feed Forward Network)

İleri beslemeli yapay sinir ağlarında, hücreler katmanlar şeklinde düzenlenir ve bir katmandaki hücrelerin çıkışları bir sonraki katmana ağırlıklar üzerinden giriş olarak verilir. Giriş katmanı, dış ortamlardan aldığı bilgileri hiçbir değişikliğe uğratmadan gizli katmandaki hücrelere iletir. Bilgi, orta ve çıkış katmanında işlenerek ağ çıkışı belirlenir. Bu yapı ile ileri beslemeli ağlar doğrusal olmayan statik bir işlevi gerçekleştirir. İleri besleme sinir ağlarında, işlem elemanları arasındaki bağlantılar bir döngü oluşturmazlar ve bu ağlar girdi veriye genellikle hızlı bir şekilde karşılık üretirler.

İleri beslemeli yapay sinir ağlarında n boyutlu girdi vektörü üzerinden ağa giriş katmanı aracılığı ile uygulanan girişler, ait oldukları ağırlıkları ile bulundukları katmanlarda hesaplanarak ve sonuç komşu katmandaki tüm nöronlara iletilerek tam olarak nöronlar arasında bir bağlantı oluşturulur. Gecikmeler yoktur ve işlem girişlerden çıkışlara doğru ilerler. Çıkış değerleri, öğreticiden alınan istenen çıkış değeriyle karşılaştırılıp bir hata sinyali elde edilir ve ağ ağırlıkları buna göre güncellenir.

İleri beslemeli ağların özellikleri:

- Katmanlar ardışık bir şekilde bir araya getirilir ve bir katmandaki hücrelerin çıkışları bir sonraki katmana ağırlıklar üzerinden giriş olarak uygulanır.
- Özellikle gizli katmanlardaki nöronların doğrusal olmayan (nonlinear) davranışlarından dolayı, ağın toplam davranışı da doğrusal olmaz.
- Ağı oluşturan giriş ve çıkış katmanlarındaki nöron sayısı problemin yapısına göre belirlenebilirken, gizli katmanların yapısını belirlemede kullanılabilecek analitik bir

yöntem henüz belirlenememiştir. Kişisel tecrübe veya deneme yanılma yöntemi ile en uygun çözüm belirlenir.

- Veri akışı ileri yönlüdür.

Ağ yapısı üzerinde veriler hep ileri yönde ilerlemektedir. Giriş katmanı yapısı itibarı ile doğrusal olmayan bir yapıya sahip olmadığı için kendisine girilen verileri değiştirmeden gizli katmanlara yönlendirir [25].

İleri beslemeli ağlara örnek vermek gerekirse perceptron, multi layer perceptron ağları söylenebilir.

2.2.2 Geri Beslemeli Yapay Sinir Ağları (Recurrent Networks, Feedback Networks)

Geri beslemeli ağ mimarileri, genellikle danışmansız öğrenme kurallarının uygulandığı ağlarda kullanılmaktadır. Bu tip ağlarda en az bir hücrenin çıkışı kendisine ya da diğer hücrelere giriş olarak verilir ve genellikle geri besleme bir geciktirme elemanı üzerinden yapılır. Geri besleme, bir katmandaki hücreler arasında olduğu gibi katmanlar arasındaki hücreler arasında da olabilir. Bu yapısı ile geri beslemeli yapay sinir ağları, doğrusal olmayan dinamik bir davranış göstermektedir. Geri beslemeli ağlara örnek olarak Hopfield, Elman ve Jordan ağları verilebilir [26].

Geri beslemeli ağlarda bağlantılar döngü içerirler ve hatta her seferinde yeni veri kullanabilmektedirler. Bu ağlar, döngü sebebiyle girdinin karşılığını yavaş bir şekilde oluştururlar. Bu yüzden, bu tür ağları eğitme süreci daha uzun olmaktadır.

Geri besleme algoritmasında her bir ağırlık için bir ilk değer belirlenmesi zorunludur. Bu algoritma süreç durdurma kriteri karşılanana kadar devam eder. Bu konudaki üç ortak nokta ise şöyledir: İterasyonların sayısını belirlemek, çıktı ortalama karesel hatası için eşik değer belirlemek ve melez doğrulamayı kullanmaktır.

Bir geri beslemeli ağ modelinde giriş, gizli ve çıkış olmak üzere 3 katman bulunmakla birlikte, problemin özelliklerine göre gizli katman sayısını arttırabilmek mümkündür [27].

Geri beslemenin yapılış şekline göre farklı yapıda ve davranışta geri beslemeli YSA yapıları elde edilebilir.

2.3 Yapay Sinir Ağlarının Eğitimi ve Öğrenmesi

Sinir ağları insanlar gibi örneklerle eğitilirler. Yapay sinir ağlarının öğrenmesi aynı bir çocuğun öğrenmesi gibidir. Sıcak bir cisme dokunmaması gerektiğini çeşitli tecrübelerle öğrenen çocuklar zamanla daha az sıcak olan bir cisme dokunabilme cesaretini gösterebilir. Bu örnekten anlaşılacağı gibi çocuk sıcaklık bilgisini öğrenmiş bulunmaktadır [28].

Öğrenme, ağı başlangıçta rastgele atanan ağırlık değerlerinin belirlenmesi işlemidir. Ağ, gördüğü her örnek için bu ağırlık değerlerini değiştirir. Bu işlem ağı doğru ağırlık değerlerine ulaşması, bir başka deyişle, örneklerin temsil ettiği olayla ilgili genelleme yapabilecek duruma gelmesi ile son bulur. Ağırlık değerlerinin değiştirilmesi işlemi, kullanılan öğrenme stratejisine göre değişen ve öğrenme kuralı olarak adlandırılan belli bazı kurallara göre yürütülür [29] .

Eğitim iki yolla uygulanabilir: Ya bir örnek sunarız ve ağırlıkları uyarlarız ya da girdi dosyasındaki tüm örnekleri sunarız, ağırlık güncellemelerini toplarız ve daha sonra ortalama ağırlık güncellemesiyle ağırlıkları güncelleriz. Bu süreç yığın öğrenme olarak adlandırılır.

Girdi ve çıktı arasındaki ilişkiyi en iyi tasvir eden bağlantı ağırlıklarının optimal değerine karar vermek için değişik yöntemler vardır. Ağırlıkların optimal değerini belirlemenin bir yolu ağırlıkları önsel bir bilgi kullanarak açık bir şekilde belirlemektir. Başka bir yolu da belirli bir öğrenme kuralına göre ağı ağırlıkları değiştirmesine izin vermesi ve sinir ağının eğitilmesidir. Eğitim, ilk rassal ağırlıkla başlayarak üretilen çıktı ve arzu edilen çıktı arasındaki farkı minimize etmek suretiyle yapılır.

Çeşitli öğrenme algoritmalarıyla hata azaltılıp gerçek çıkışa yaklaşılmaya çalışılır. Bu çalışma süresince yenilenen, YSA ağırlıklarıdır. Ağırlıklar her bir çevrimde yenilenerek amaca ulaşmaya çalışılır. Amaca ulaşmanın veya yaklaşmanın ölçüsü de yine dışarıdan verilen bir değerdir. Eğer YSA verilen giriş-çıkış çiftleriyle amaca ulaşmış ise ağırlık değerleri saklanır. Ağırlıkların sürekli yenilenip istenilen sonuca ulaşılan kadar geçen zamana öğrenme adı verilir ve YSA için en önemli kavramlardan birisidir.

YSA öğrendikten sonra daha önce verilmeyen girişler verilip, sinir ağı çıkışıyla gerçek çıkışın yaklaşımı incelenir. Eğer yeni verilen örneklerle doğru yaklaşıyorsa sinir ağı işi öğrenmiş demektir. Sinir ağına verilen örnek sayısı optimum değerden fazla ise sinir ağı

iş i öğrenmemiş ezberlemiştir. Genellikle eldeki örneklerin yüzde sekseni ağ a verilip ağ eğitilir, daha sonra geri kalan yüzde yirmilik kısım verilip ağ ın davranış ı incelenerek ağ test edilir [19].

Sinir ağ ını eğittiğ imiz zamanki ortak bir prosedür; verileri eğitim seti, doğrulama seti ve test seti olarak bölmektir. Eğitim seti ağı rlıkları tahmin etmek için kullanılır. Ağ öğrenirken performansı doğrulama setinden izlenir ve daha sonra test seti bilinmeyen veri üzerinde ağ ın performansını deę erlendirmek için kullanılır.

Doğ rulama seti, eğ itimi ne zaman durdurmamız gerektiğ i konusunda karar vermek için kullanılır. Test verisi eğitim ve doğrulama süreci içerisinde tamamen dokunulmadan bırakılır. Test seti eğitim ve erken durdurma süreci esnasında kullanılmaz ancak yeni veriler üzerinde eğ itilen modelin performansını görmek için kullanılır. Doğ rulama setinin performansı azalmaya başlad ıę ı zaman eğ itim durdurulmalıdır.

Genel olarak ifade etmek gerekirse eğ itim sı rasındaki amaç , bulunması gereken doğ ru sonuçlara en yakın çıktı yı üretebilmektir. Bu sebeple ağ verilen girdilere göre kendi mimarisine de uygun olarak işlem yaptıktan sonra bir çıktı üretir. Ç ıktı ile hedef deę erler arasındaki fark hatadır. Ağ bu hatayı kabul edilebilir sınırlar arasına indirebilmek için işlemi tekrarlar. Eğ itim setinin ağ içinde bir kez işlem den geçirilmesine iterasyon (epoch) denir. iterasyon sayısının çok olması ağ ın öğrenme sürecinde önemli bir etkindir. Fakat iterasyon sayısının yüksek seç ilmesi de performansı düş üren bir etkindir. Ağ ın mimarisi, aktivasyon fonksiyonu, öğrenme yönt emi ve iterasyon sayısı seç ilirken optimizasyonun iyi bir şekilde yapılması gerekir.

Bütün yapay sinir ağ ları; sinirler, baę lantılar ve aktarım iş levlerine baę lı olduę u için, farklı mimariler, yapılar ya da sinir ağ ları arasında bir benzerlik bulunmaktadır. Ç eş itlilik ise çoę unlukla farklı öğrenme kurallarından ve bu kuralların bir ağ ın yapısını deę işt irmesinden kaynaklanmaktadır.

Eğitim zamanını azaltmak ve daha iyi performans için aşağıda belirtilenler göz önünde tutulmalıdır.

- Özellikle gizli katmanlardaki nöronların doğrusal olmayan eğitim verisini normalleştirmek
- Lojistik fonksiyon yerine doğrusal olmayan hiperbolik tanjant fonksiyonu kullanmak
- Girdiye yönelik daha yüksek adım genişliği ayarlamak
- Eğriselliğin doğrusal bölgesinde çıktı ağırlıklarını sıfırlamak
- Kullanılan öğrenme yöntemini değiştirmek
- Daima ağırlıklardan daha fazla eğitim örneklerine sahip olmak

2.3.1 Öğrenme Eğrisi

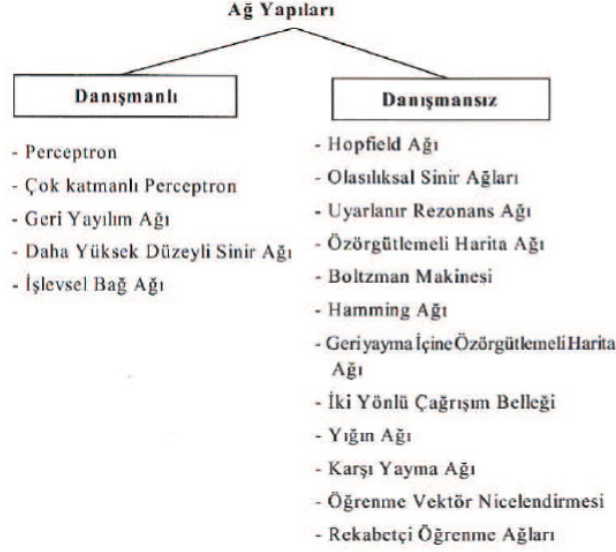
Öğrenme sürecindeki ilerlemeyi ölçmek; herhangi bir eğitim sürecinde oldukça önemlidir. Öğrenme sürecini gösteren öğrenme eğrisi hata kareleri ortalamasının eğitim iterasyonu boyunca nasıl hareket ettiğini göstermektedir. Öğrenme parametrelerinin nasıl kontrol edilmesi gerektiği öğrenme eğrisinden görülebilir. Öğrenme eğrisi düzleştiği zaman öğrenme oranı öğrenmeyi hızlandırmak için arttırılmalıdır. Öte yandan öğrenme eğrisi aşağı yukarı dalgalandığı zaman eşik büyüklüğü azaltılmalıdır.

Ekstrem noktada hata öğrenmenin istikrarsız olduğunu göstererek yükselir. Bu noktada ağ yeniden başlatılmalıdır. Birçok iterasyondan sonra kabul edilemeyen bir hata düzeyinde öğrenme eğrisi istikrarlı hale geldiği zaman ağın topolojisi ya da eğitim prosedürü yeniden gözden geçirilerek değiştirilmelidir.

2.3.2 YSA'nın Öğrenme Yapısına Göre Sınıflandırılması

Yapay sinir ağları kullandığı öğrenme stratejilerine göre 3'e ayrılmaktadır:

- Danışmanlı Öğrenme (Supervised Training)
- Danışmansız Öğrenme (Unsupervised Training)
- Takviyeli Öğrenme (Reinforcement Training)



Şekil 2. 5 YSA sınıflandırma

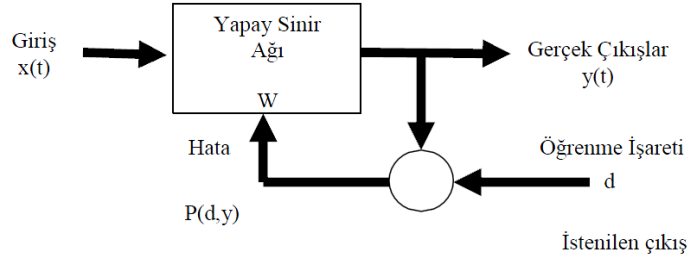
2.3.2.1 Danışmanlı Öğrenme (Supervised Training)

Ağ eğitilirken, çıktıların istenen değerleri ağa tanıtılabiliyorsa, bu tip öğrenmeye danışmanlı öğrenme adı verilir. Girdi ve çıktı kümeleri ağa verilir. Ağ, girdiyi süreçten geçirerek kendi çıktısını üretir ve gerçek çıktı ile karşılaştırır. Öğrenme sayesinde, mevcut hatayı en aza indirmek için bağlantılardaki ağırlıklar yeniden düzenlenir. Bu işlem, hata seviyesi kabul edilebilir seviyeye ulaşınca kadar devam eder [30].

Danışmanlı öğrenmede, her bir örnekleme zamanında giriş uygulandığında sistemin arzu edilen çıkışı (y), eğitici tarafından sağlanır. Arzu edilen çıkış (y) ile sinir ağı çıkışı arasındaki fark hata ölçüsüdür ve ağ parametrelerini güncellemekte kullanılır. Ağırlıkların güncellenmesi süresince eğitici ödüllendirme-cezalandırma şemasını ağa uygulayarak hatayı azaltır. Bu öğrenme modelinde giriş ve çıkış örnekleri kümesi eğitim kümesi olarak adlandırılır. Danışmanlı öğrenme, ağın çıktı için istenilen veri değerleri verebilmesi için girdi-çıkış ilişkisini elde edebilmesini sağlayacak bir mekanizma içermektedir. Burada eğitici, ağ sonuçlarının geçerliliğini kontrol eden bir gözlemci olabilir.

Yapay sinir ağlarının büyük bir çoğunluğunda danışmanlı öğrenme kullanılmaktadır. Bu yöntemde, ağ tarafından üretilen çıkış değeri istenilen gerçek çıkış değeri ile karşılaştırılır. Başlangıçta ağırlıklar rastgele bir şekilde belirlenir, bir sonraki yineleme değerleri göz önüne alınarak ağırlıklar yeniden ayarlanır. Böylelikle istenilen ve

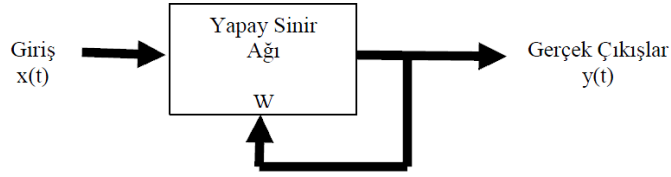
hesaplanan çıkış değerlerinin birbirine yakınlaşması ve tüm işlem elementlerindeki hatanın mümkün en küçük değerine ulaşması amaçlanır. Danışmanlı öğrenme metodunda, ağ kullanışlı hale gelmeden önce eğitilmek zorundadır. Eğitim, eğitim seti olarak bilinen giriş ve çıkış verilerinin ağı sunulmasından oluşmaktadır. Ağ, giriş verileri ve ağ üzerinde sürekli değişerek ayarlanan ağırlıkları kullanarak kullanıcının belirlediği performans seviyesine gelene kadar hedeflenen çıkış değerleri ve üretilen çıkış değerleri üzerinde işlemleri gerçekleştirir. Ağın eğitimi için kullanılacak olan eğitim setinin doğası ağ performansı ve öğrenmenin gerçekleşme süreci açısından oldukça önemlidir. Şekil 2.6’da danışmanlı eğitim yapısı görülmektedir. Danışmanlı öğrenme prensibini kullanan ağ yapılarına perceptron ve çok katmanlı perceptron örnek verilebilir.



Şekil 2. 6 Danışmanlı öğrenme

2.3.2.2 Danışmansız Öğrenme (Unsupervised Training)

Çıktıların istenen değerleri ağı tanıtılamıyorsa, bu tip öğrenme şekli danışmansız öğrenme olarak adlandırılır. Danışmansız öğrenme algoritmaları daha çok sistemin geçmişte karşılaştığı veri kümesinin içerdiği istatistiksel bilgilerin elde edilmesini amaçlar. Çok elemanlı veri kümeleri içerisinde deneme-yanılma yoluyla bilgi genelleştirilmesi yapılabilir. Danışmansız öğrenmede girdiler aynı zamanda çıktı görevi görmektedir. Hem çıktı hem de girdi olan veriler arasındaki kural ve ilişkilerin araştırılması ve en uygununun bulunması ağın eğitilmesi anlamına gelmektedir. Şekil 2.7’de danışmansız öğrenme yapısı görülmektedir.



Şekil 2. 7 Danışmansız öğrenme

Belirtildiği gibi, danışmansız öğrenmede danışmanlı öğrenmedeki gibi istenilen çıkış değerleri bilinmemektedir. Bu yüzden kesin bir hata bilgisini ağın davranışını değiştirmekte kullanmak mümkün değildir. Cevabın doğruluğu veya yanlışlığı hakkında bilgi sahibi olunmadığı için öğrenme, girişlerin verdiği cevaplar sayesinde olur. Yani, dışarıdan herhangi bir etki olmaksızın nöronlar arasındaki ilişki kendi kendine oluşur [25].

Danışmansız öğrenmenin temeli dış müdahale olmaksızın, girdilerin ağ tarafından analiz edilmesi ve bu analiz sonucunda bağlantıların oluşturulmasıdır.

Örneklendirmek gerekirse, 100 kişilik bir sınıfın boy, kilo, saç rengi ve ayakkabı numarası girdilerin değerlerini oluştursun. Yapay sinir ağı ilk gelen kişiyi bir sınıf olarak ilan edecek ve ardından sisteme gelen tüm girdileri bu buna benziyor ya da benzemiyor diye ayırt edecektir. Bu ayrımı yapabilmesi için 100 kişilik grubun değerleri sisteme tekrar tekrar girdiği takdirde, yapay sinir ağı en sonunda öğrenmesini tamamlayacak ve 2 sınıf öğrenmiş olacaktır. 1. girdiye benzeyenler ve benzemeyenler. Adını veremediği bu sınıflar yardımıyla “hanımlar” ve “beyler” gibi rahatlıkla sınıf ayrılmış daha doğrusu bu iki sınıf başarıyla öğrenilmiş olunacaktır. Böyle bir ağ modeline yabancı bir kişi değerleri verilirse, sinir ağı bu kişiye cinsiyetini söyleyecektir.

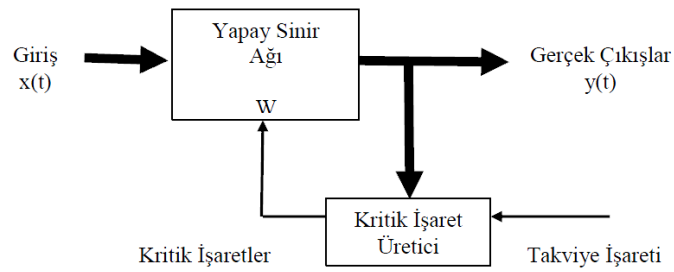
Danışmansız öğrenme stratejisini kullanan ağ yapılarına Hopfield Ağı, Boltzman Makinesi, Hamming Ağı, Rekabetçi Öğrenme Ağları ve ART (Adaptif Rezonans Teorisi) ağları örnek gösterilebilir.

2.3.2.3 Takviyeli Öğrenme (Reinforcement Training)

Takviyeli öğrenme yöntemi danışmanlı öğrenme yöntemine benzemekle birlikte, ağa hedef çıktılar yerine, ağın çıktılarının ne ölçüde doğru olduğunu belirten bir skor veya derece bildirilir. Danışmansız öğrenme algoritması, istenilen çıkışın bilinmesine gerek duymaz.

Hedef çıktıyı vermek için bir öğretmen yerine burada yapay sinir ağına bir çıkış verilmemekte, elde edilen çıkışın verilen girişe karşılık iyi ya da kötü olarak değerlendiren bir ölçüt kullanılmaktadır. Performans bilgisi genellikle iki sayıdır ve denetim hareketlerinin başarısını göstermektedir. Optimizasyon problemlerini çözmek için Hilton ve Sejnowski tarafından geliştirilen “Boltzman Kuralı”, takviyeli öğrenmeye örnek olarak verilebilir.

Takviyeli öğrenmede, ağın davranışının uygun olup olmadığını belirten bir öz yetenek bilgisine ihtiyaç duyulur. Bu bilgiye göre ağırlıklar ayarlanır. Gerçek zamanda öğrenme yöntemi olup deneme-yanılma esasına göre sinir ağı eğitilmektedir.



Şekil 2. 8 Takviyeli öğrenme

2.3.3 Öğrenme Kuralları

Öğrenme kuralları birimler arasındaki bağlantıların ağırlıklarının ayarlanmasıyla sonuçlanır. Temel fikir “j” ve “k” gibi iki birim eş zamanlı olarak aktifse onların iç bağlantısının ağırlıklandırılması gerektiğidir. Bir diğer genel kural “k” biriminin asıl aktivasyonunun kullanılmaması fakat ağırlıkları ayarlamak için asıl ve arzu edilen aktivasyon arasındaki farkın kullanılmasıdır [31]. Öğrenme işlemini kolaylaştırmak için birçok öğrenme kuralı (algoritması) geliştirilmiştir.

En bilinen ve en yaygın olarak kullanılan öğrenme kuralları şunlardır:

- Hebb Kuralı
- Hopfield Kuralı
- Kohonen Kuralı
- Delta Kuralı

2.3.3.1 Hebb Kuralı

Öğrenme kuralları arasında en çok bilinen bu kural, Ponaldd Hebb tarafından geliştirilmiş ilk öğrenme kuralıdır. Bu kural, Hebb' in 1949 yılında yayınlanan The Organization of Behavior adlı kitabında ilk kez tanımlanmıştır. Bu kuralın temel tanımı şöyle yapılmıştır: Eğer bir nöron diğer bir nörondan girdi aldığında bu iki nöron yüksek aktivitede ise yani matematiksel olarak aynı işarete sahip ise, nöronlar arasındaki ağırlık güçlendirilmelidir [32]. Diğer bir deyişle, eğer ağ içerisinde birbirine bağlı iki hücre (nöron) aynı anda “on” durumunda ise, bu iki ünite arasındaki ağırlık değerlerinin artması beklenir ve sinapsların gerilimlerinin (ağırlıklarının) ayarlanması ile öğrenmenin oluşacağı öngörülür.

2.3.3.2 Hopfield Kuralı

Hebb kuralına benzer bir kuraldır ancak bir istisnası vardır. Hopfield kuralına göre, ağırlıkların güçlendirilmesi işleminde güçlendirme veya zayıflatma için bir büyüklük tanımlamak gerekmektedir. İstenen çıktı ve girdinin her ikisi de aktif veya pasifse, bağlantı ağırlığı öğrenme oranı kadar artırılır. Ters durumda ise bağlantı ağırlığı öğrenme oranı kadar azaltılır. Hopfield kuralı, Hebb kuralından farklı olarak YSA elemanlarının bağlantılarının ne kadar güçlendirilmesi ya da zayıflatılması gerektiğini de belirler. Eğer beklenen çıktılar ve girdilerin her ikisi de aktif veya pasif ise, öğrenme katsayısı kadar ağırlık değerlerini de güçlendirir veya zayıflatır. Ağırlıkların güçlendirilmesi ya da zayıflatılması, öğrenme katsayısının yardımıyla gerçekleşir. Öğrenme katsayısı, genellikle 0 ile 1 arasında kullanıcı tarafından belirlenen sabit pozitif bir değerdir [20].

2.3.3.3 Kohonen Kuralı

Bu kural, Kohonen tarafından 1982 yılında geliştirilmiştir. Bu kuralda sinir hücreleri ağırlıkları değiştirmek için birbirleri ile yarışlar. En büyük çıktıyı üreten hücre, kazanan hücre olur ve bağlantı ağırlıklarını değiştirir. “Kazanan hepsini alır (Winner takes all.)” yaklaşımını uygular.

Kazanan hücre, yakınındaki hücelere göre daha kuvvetli hale gelmektedir. Böylece kazanan hücrenin komşularına bağlantı ağırlıklarını güncellemeleri için izin verilir. Bağlantı ağırlıklarını daha fazla güncelleyebilmek, en büyük çıktıyı üretebilme

denemelerindeki şansı artırdığı için, kazanan bir hücre, komşu hücrelerin bağlantı ağırlıklarının güncellenmesine izin vererek onların da şanslarını arttırmış olur.

2.3.3.4 Delta Kuralı

En çok kullanılan kurallardan biri olan Delta kuralı, Hebb kuralının daha geliştirilmişidir. Bu kural bir sinirin gerçek çıkışı ile istenilen çıkış değeri arasındaki farkı azaltmak için giriş bağlantı güçlerini sürekli olarak geliştirme fikrine dayanır. Bu kural ağ hatasının karesini minimize etmek için bağlantı boyutlarını değiştirir. Hata bir önceki katmana geri çoğaltılır. Her bir zaman dilimi için bir hata şeklinde bu geri çoğaltma işlemi ilk katmana ulaşıncaya kadar devam eder. Bu tip ağ ileri beslemeli ağ olarak isimlendirilir, geri yayılım ağı adını bu hata terimlerini toplama yönteminden türetir. Bu kural ayrıca Windrow-Hoff Öğrenme ve En Küçük Ortalamalar Karesi (Least Mean Square) kuralı olarak da adlandırılır.

Bu kurala göre hedef çıktı ile elde edilen çıktı arasındaki farkı azaltmak için YSA elemanlarının bağlantılarının ağırlık değerleri sürekli yeniden hesaplanır. Amaç hedef çıktı ile elde edilen çıktı arasındaki hata karelerinin ortalamasını en aza indirebilmektir. Hatalar en son katmandan geriye doğru ardışık iki katman arasındaki bağlantı ağırlıklarına dağıtılır [33].

2.3.4 Uygulamaya Göre Öğrenme Algoritmaları

Çevrimiçi (On-line) Öğrenme

Bu kurala göre öğrenen sistemler, gerçek zamanda çalışırken bir taraftan fonksiyonlarını yerine getirmekte, bir taraftan da öğrenmeye devam etmektedirler. Kohonen öğrenme kuralı çevrimiçi öğrenmeye örnek verilebilir.

Çevrim Dışı (Offline) Öğrenme

Bu kurala dayalı sistemler, kullanıma alınmadan önce örnekler üzerinde eğitilirler. Bu kuralı kullanan sistemler eğitildikten sonra gerçek hayatta kullanıma alındığında artık öğrenme olmamaktadır. Delta öğrenme kuralı bu tür öğrenmeye örnek verilebilir.

2.4 Yapay Sinir Ağlarının Tasarımı

YSA uygulamasının başarısı, uygulanacak olan yaklaşımlar ve deneyimlerle yakından ilgilidir. Uygulamanın başarısında uygun metodolojiyi belirlemek büyük önem taşır. Yapay sinir ağının geliştirilmesi sürecinde ağın yapısına ve işleyişine ilişkin şu kararların verilmesi gerekir:

- Ağ mimarisinin seçilmesi ve yapı özelliklerinin belirlenmesi (katman sayısı, katmandaki nöron sayısı gibi)
- Nörondaki fonksiyonların karakteristik özelliklerinin belirlenmesi
- Öğrenme algoritmasının seçilmesi ve parametrelerinin belirlenmesi
- Eğitim ve test verisinin oluşturulması

Bu kararların doğru verilememesi durumunda, YSA'ların sistem karmaşıklığı artacaktır. Sistem karmaşıklığı yapısal ve toplam hesaplama karmaşıklığının bir fonksiyonudur. Toplam hesaplama karmaşıklığı ise, genellikle yapısal karmaşıklığın bir fonksiyonu olarak ortaya çıkar ve bu hesaplamanın en aza indirilmesi amaçlanır. Bu hesaplama karmaşıklığının ölçülmesinde de genellikle YSA sisteminin toplam tepki süresi veya sisteme ait bir işlemci elemanın tepki süresi değeri temel alınır. Bunun yanında kapladığı hafıza ve zaman karmaşıklığı bazı uygulamalarda hesaplanmaktadır.

Bir YSA'nın uygun parametrelerle tasarlanması durumunda YSA sürekli olarak kararlı ve istikrarlı sonuçlar üretecektir. Ayrıca sistemin tepki süresinin kısa olabilmesi için de ağ büyüklüğünün yeterince küçük olması gerekir. İhtiyaç duyulan toplam hesaplama da bu sayede sağlanmış olacaktır.

2.4.1 YSA Ağ Yapısının Seçilmesi

YSA'nın tasarımı sürecinde ağ yapısı uygulama problemine bağlı olarak seçilmelidir. Hangi problem için hangi ağın daha uygun olduğunun bilinmesi önemlidir. Kullanım amacı ve o alanda başarılı olan ağ türleri Çizelge 2.4'te verilmiştir.

Çizelge 2. 4 Ağ türleri ve başarılı oldukları alanlar[19]

Kullanım Amacı	Ağ Türü	Ağın Kullanımı
Tahmin	• ÇKA	Ağın girdilerinden bir çıktı değerinin tahmin edilmesi
Sınıflandırma	• LVQ • ART • Counterpropagation • Olasılıklı Sinir Ağları	Girdilerin hangi sınıfa ait olduklarının belirlenmesi
Veri İlişkilendirme	• Hopfield • Boltzman Machine • Bidirectional As.Memory	Girdilerdeki hatalı bilgilerin bulunması ve eksiklerin tamamlanması

Uygun YSA yapısının seçimi, büyük ölçüde ağda kullanılması düşünülen öğrenme algoritmasına bağlıdır. Ağda kullanılacak öğrenme algoritması seçildiğinde, bu algoritmanın gerektirdiği mimari zorunlu olarak seçilmiş olacaktır. Örneğin geri yayılım algoritması ileri beslemeli ağ mimarisi gerektirir.

Bir YSA'nın karmaşıklığının azaltılmasında en etkin araç, YSA ağ yapısını değiştirmektir. Gereğinden fazla sayıda işlemci eleman içeren ağ yapılarında, daha düşük genelleme kabiliyeti ile karşılaşılır.

2.4.2 Öğrenme Algoritmasının Seçilmesi

YSA yapısının seçiminden sonra uygulama başarısını belirleyen en önemli faktör öğrenme algoritmasıdır. Genellikle ağ yapısı öğrenme algoritmasının seçiminde belirleyicidir. Bu nedenle seçilen ağ yapısı üzerinde kullanılabilecek öğrenme algoritmasının seçimi ağ yapısına bağlıdır. Yapay sinir ağının geliştirilmesinde kullanılacak çok sayıda öğrenme algoritması bulunmaktadır. Bunlar içinde bazı algoritmaların belli tip uygulamalar için daha uygun olduğu bilinmektedir.

2.4.3 Ara Katman Sayısının Belirlenmesi

YSA'nın tasarımı sürecinde tasarımcının yapması gereken diğer işlem, ağdaki katman sayısına karar vermektir. Nöronların aynı doğrultu üzerinde bir araya gelmeleriyle katmanlar oluşmaktadır. Çoğu problem için 2 veya 3 katmanlı bir ağ tatmin edici sonuçlar üretebilmektedir. Katmanların değişik şekilde birbirleriyle bağlanmaları

değişik ağ yapılarını oluşturur. Girdi ve çıktı katmanlarının sayısı, problemin yapısına göre değişir. Katman sayısını belirlemenin en iyi yolu, birkaç deneme yaparak en uygun yapıya karar vermektir.

2.4.4 Nöron Sayısının Belirlenmesi

Ağın yapısal özelliklerinden birisi her bir katmandaki nöron sayısıdır. Neden sonuç ilişkisine dayalı tahmin problemlerinde girdi nöron sayısı, değişken sayısına eşittir dolayısıyla girdi nöron sayısını belirlemek kolaydır. Ancak zaman serisi tahmin problemlerinde girdi nöron sayısı gecikme sayısı ile ilişkilidir ve bu sayıyı belirlemek için önerilen belirli bir yol yoktur.

Gizli nöron sayısı, girdi ile çıktı nöronları arasındaki doğrusal olmayan ilişkinin kurulmasını ve ağ ağırlıklarının belirlenmesini sağlar. Genellikle tek gizli katman kullanılması önerilir. Katmandaki nöron sayısının tespitinde deneme-yanılma yöntemi kullanılır. Bunun için, başlangıçtaki nöron sayısını istenilen performansa ulaşıncaya kadar arttırmak veya tersi şekilde istenen performansın altına inmeden azaltmak gerekir. Bir katmanda kullanılacak nöron sayısı olabildiğince az olmalıdır. Nöron sayısının az olması yapay sinir ağının "genelleme" yeteneğini artırırken, gereğinden fazla olması ağın verileri ezberlemesine neden olur. Ancak gereğinden az nöron kullanılması verilerdeki örüntünün ağ tarafından öğrenilememesi gibi bir sorun yaratabilir.

Nörondaki fonksiyonların karakteristik özellikleri de YSA tasarımıda önemli kararlardan biridir. Nöronun aktivasyon fonksiyonunun seçimi büyük ölçüde yapay sinir ağının verilerine ve ağın neyi öğrenmesinin istendiğine bağlıdır. Aktivasyon fonksiyonları içinde en çok kullanılan sigmoid ve hiperbolik tanjant fonksiyonlarıdır. Sigmoid fonksiyonunun çıktısı 0 ve 1 arasında olurken, hiperbolik tanjant fonksiyonunun çıktısı 1 ve (-1) arasında olmaktadır.

Eğer ağın bir modelin ortalama davranışını öğrenmesi isteniyorsa sigmoid fonksiyon, ortalamadan sapmanın öğrenilmesi isteniyorsa hiperbolik tanjant fonksiyon kullanılması gerekmektedir.

2.4.5 Normalizasyon

YSA'ların en belirgin özelliklerinden olan doğrusal olmama özelliğini anlamlı kılan yaklaşım, verilerin bir normalizasyona tabii tutulmasıdır. Verilerin normalizasyonu için

seçilen yöntem YSA performansını doğrudan etkileyecektir. Çünkü normalizasyon, giriş verilerinin transfer edilirken fonksiyonun aktif olan bölgesinden aktarılmasını sağlar. Veri normalizasyonu, işlemci elemanlarını verileri kümülatif toplamaların oluşturacağı olumsuzluklardan korur. Veri normalizasyonu, bu nedenle zorunludur ve aşırı değerlendirilmiş kümülatif toplamaların oluşturacağı olumsuzlukları bertaraf eder.

Genellikle verilerin $[0,1]$ veya $[-1,+1]$ aralıklarından birine ölçeklendirilmesi önerilmektedir. Ölçekleme verilerin geçerli eksen sisteminde sıkıştırılması anlamı taşıdığından veri kalitesi aşırı sınımlar içeren problemlerin YSA modellerini olumsuz yönde etkileyebilir. Bu olumsuzluk, kullanılacak öğrenme fonksiyonunu da başarısız kılabilir.

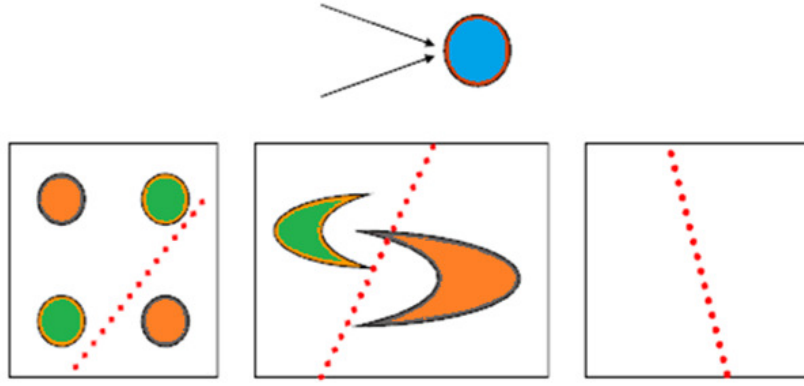
2.4.6 Performans Fonksiyonu Seçimi

Öğrenme performansını etkileyen önemli hususlardan biri performans fonksiyonudur. İleri beslemeli ağlarda genellikle kullanılan performans fonksiyonları karesel ortalama hata (MSE Mean Square Error) ve toplam karesel hatadır (SSE Sum Square Error).

2.5 Basit Algılayıcı Modeli (Perceptron)

İlk defa 1958 yılında Rosenblatt tarafından örüntü sınıflandırma amacı ile geliştirilmiştir. Algılayıcı öğrenme kuralı, Hebb öğrenme kuralına göre daha güçlü bir algoritmik yapıya sahiptir. Bir sinir hücresinin birden fazla girdiyi alarak bir çıktı üretmesi prensibine dayanmaktadır. Ağın çıktısı 1 veya 0'dan oluşan mantıksal bir değerdir. Basit olarak bir Algılayıcı ünitesi (hücresi), algılama ünitesi, ilişkilendirme ünitesi ve cevap ünitesi olmak üzere üç kısımdan oluşmaktadır. Algılayıcı, eğitilebilen tek bir yapay sinir hücresinden oluşmaktadır. Girdiler ağa sunulmaktadır ve her girdi setine karşılık gelen çıktı değerleri de ağa gösterilmektedir. Daha sonra öğrenme kuralına göre ağın çıktı değeri hesaplanır. Eğer ağın çıktısı olması gereken çıktıdan farklı ise ağırlıklar ve eşik değerler yeniden ayarlanır.

Basit algılayıcılar şekil 2.9'da görüldüğü üzere sadece doğrusal sınıflandırma yapabilir.



Şekil 2. 9 Basit algılayıcı modeli

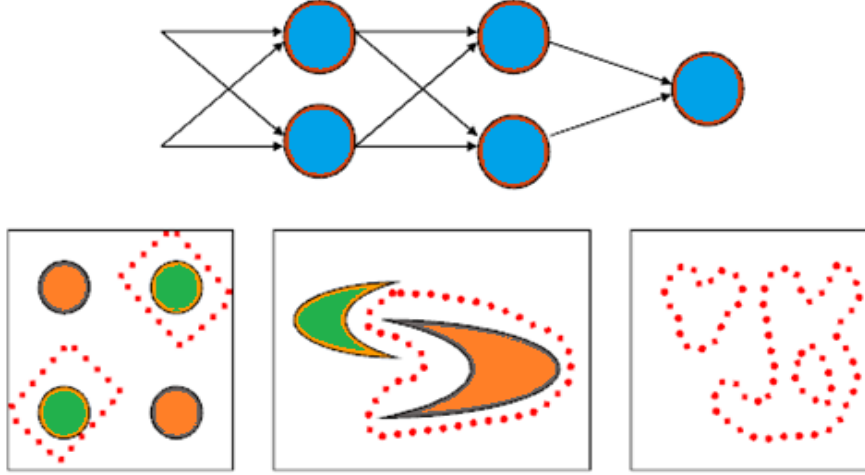
2.6 Çok Katmanlı Algılayıcılar ÇKA (Multi Layer Perceptron MLP)

Doğrusal olmayan bir ilişki gösteren XOR problemini çözmek amacıyla yapılan çalışmalar sonucunda geliştirilen bir ağıdır. Rumelhart ve arkadaşları tarafından tasarlanan bu modele Hata Yayma Modeli veya Geriye Yayılım Modeli (backpropagation network) de denilmektedir. ÇKA modeli yapay sinir ağlarına olan ilgiyi çok hızlı bir şekilde artırmış ve YSA tarihinde yeni bir dönem başlatmıştır. Bu ağ modeli özellikle mühendislik uygulamalarında en çok kullanılan sinir ağı modeli olmuştur. Birçok öğretim algoritmasının bu ağı eğitmede kullanılabilir olması, bu modelin yaygın olmasının sebebidir. Konveks alanları birleştirir. Şekil 2.10'da çok katmanlı bir ağ görülmektedir.

Bilgi akışı ileri doğru olup geri besleme yoktur. Bunun için ileri beslemeli sinir ağı modeli olarak adlandırılır. Giriş katmanında herhangi bir bilgi işleme yapılmaz. Buradaki işlem elemanı sayısı tamamen uygulanan problemlerin giriş sayısına bağlıdır. Ara katman sayısı ve ara katmanlardaki işlem elemanı sayısı ise, deneme-yanılma yolu ile bulunur. Çıkış katmanındaki eleman sayısı ise yine uygulanan probleme dayanılarak belirlenir.

MLP ağında katman sayısı ve her katmandaki düğüm sayısı dikkatle seçilmelidir. Bu sayıların ne olacağı hakkında kesin bir yöntem yoktur ve sadece takip edilecek genel kurallar bulunmaktadır.

Kural Bir: Girdi verisi ve istenilen çıktı arasındaki ilişkinin karmaşıklığı artınca, gizli katmanlardaki işleme elemanlarının sayısı da artmalıdır.



Şekil 2. 10 Çok katmanlı algılayıcı

Kural İki: Eğer ele alınan süreç birçok aşamalara ayrılabilirse, fazla sayıda gizli katman kullanılmalıdır. Eğer süreç aşamalara ayrılamıyorsa, fazla sayıda gizli katman kullanılırsa ağda yalnızca ezberleme ortaya çıkar ve yanlış sonuçlara yol açar.

Kural Üç: Ağda kullanılan eğitim verisinin miktarı, gizli katmanlardaki işleme elemanlarının sayısı için üst bir sınır oluşturmaktadır. Bu üst sınırı bulmak için önce eğitim kümesindeki girdi ve çıktı çiftlerinin sayısı bulunur. Bulunan bu sayı ağdaki toplam giriş ve çıkış düğümlerinin sayısına bölünür. Çıkan sonuç, beş ile on arasında bir dereceleme faktörüne bölünür. Daha büyük dereceleme faktörleri göreceli olarak işitsel veriler için kullanılmaktadır. İşitsel veriler 20-50 arasında bir faktör gerektirirken, çıktı ile belirgin bir ilişkisi olan verilerde faktör 2 civarına kadar düşürülebilir. Gizli katmanların az miktarda işleme elemanına sahip olması önemli bir faktördür.

ÇKA modelinin temel amacı, ağın beklenen çıktısı ile ürettiği çıktı arasındaki hatayı en aza indirmektir. Bu ağlara eğitim sırasında hem girdiler hem de o girdilere karşılık üretilmesi gereken çıktılar gösterilir. Ağın görevi her girdi için o girdiye karşılık gelen çıktıyı üretmektir. Örnekler giriş katmanına uygulanır, ara katmanlarda işlenir ve son katmandan da çıkışlar elde edilir.

Delta Öğrenme Kuralını kullanan bu ağ modeli, özellikle sınıflandırma, tanıma ve genelleme yapmayı gerektiren problemler için çok önemli bir çözüm aracıdır.

2.7 Radyal Tabanlı Sinir Ağı (Radyal Based Network RBN)

Ara katmandaki işlemci elemanlar girişlerin ağırlıklandırılmış şeklini kullanmamakta ve ara katmandaki işlemci elemanların çıkışları YSA girişleri ile temel fonksiyonun merkezi arasındaki uzaklığa göre belirlenmektedir. Radyal tabanlı fonksiyon ağı tasarımı çok boyutlu uzayda eğri uydurma yaklaşımıdır. Bu nedenle bu ağın eğitimi, çok boyutlu uzayda eğitim verilerine en uygun bir yüzeyi bulma problemine dönüşmektedir.

Radyal tabanlı fonksiyonlar, sayısal analizde çok değişkenli interpolasyon problemlerinin çözümünde kullanılmış ve YSA' nın gelişmesi ile birlikte bu fonksiyonlardan YSA tasarımında yararlanılmıştır. Bu ağ, ileri beslemeli YSA yapılarına benzer şekilde giriş, orta ve çıkış katmanından oluşur. Ancak, giriş katmanından orta katmana dönüşüm, radyal tabanlı aktivasyon fonksiyonları ile doğrusal olmayan sabit bir dönüşümdür. Radyal tabanlı oluşturulan MLP'de standart sigmoid fonksiyonlarına tercihen Gaussian transfer fonksiyonları kullanılır. Orta katmandan çıkış katmanına ise uyarlamalı ve doğrusal bir dönüşüm gerçekleştirilir.

RBN'lere uyarlanabilecek serbest parametreler; merkez vektörleri, radyal fonksiyonların genişliği ve çıkış katman ağırlıklarıdır. Çıkış katmanı doğrusal olduğundan ağırlıklar, eğim düşme ya da doğrusal en iyileme yöntemleri ile kolayca bulunabilir. Merkezler, girişler arasından rastgele ve sabit olarak seçilebilmekle birlikte ağın performansını iyileştirmek amacıyla merkez vektörlerinin ve genişliğin uyarlanması için çeşitli yöntemler geliştirilmiştir.

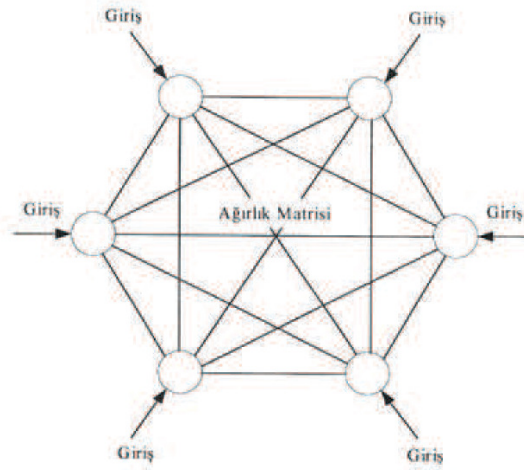
Merkez vektörleri, eğim düşme yöntemine göre eğiticili öğrenme algoritması ile uyarlanarak, dik en küçük kareler yöntemi ile ya da kendiliğinden düzenlemeli yöntemle giriş örneklerinden belirlenebilir.

2.8 Hopfield Ağı

Hopfield ağları; 1980'lerin başlarında John Hopfield tarafından sunuldu. CAM (Content Addressable Memories) denilen sinirsel fizyolojiden harekete geçen modellerden yararlanılan Hopfield sinir ağı, beyine benzer bir şekilde belirli hafızaları veya desenleri depolayan basit bir yapay sinir ağıdır.

Hopfield ağ topolojisi diğer ağlardan farklıdır. Farklı katmanlar yoktur; her birim diğer tüm birimlere bağlıdır. Ayrıca, bağlantılar çift yönlüdür (bilgi her iki yönde akar) ve

simetriktr. Bu ağ genellikle ikili (0 veya 1) ve bipolar (+1 veya -1) girişler kabul eder. Tek tabaka işlemci elemanları vardır ve her işlemci eleman bir diğerine bağlanmıştır. Hopfield ağı, bir geri yayımlı ağı eğitildiği gibi eğitilmez. Bunun yerine, örnek desen grupları seçilir, örnek desenler ağı ağırlıklarının başlangıç değerlerini saptamak için kullanılır. Bu bir kere yapıldıktan sonra herhangi bir desen ağı sunulur ve bu da giriş desenine en çok benzeyen örnek desenlerden biriyle sonuçlandırılır. Çıkış deseni, birimlerin durumlarına bakılarak ağıdan okunabilir. Şekil 2.11’de Hopfield ağı görülmektedir.



Şekil 2. 11 Hopfield ağı

2.9 Öğrenme Vektör Parçalama Ağı (LVQ Learning Vector Quantisation)

LVQ ağı 1984 yılında Kohonen tarafında geliştirilmiştir. YSA modellerinin bazılarında eğitim sırasında ağı hem girdi değerleri hem de o girdi değerleri için üretilecek çıktı değerlerinin ne olması gerektiği verilir. Fakat bazı durumlarda, ağı çıktının ne olduğunu vermek mümkün olmamaktadır. Yalnızca ağı üretmiş olduğu çıktının doğru veya yanlış olduğu belirtilebilmektedir. Destekleyici öğrenme olarak belirlenen bu yöntemi kullanan modellerin bir tanesi de, doğrusal vektör parçalama modeli olan LVQ modelidir.

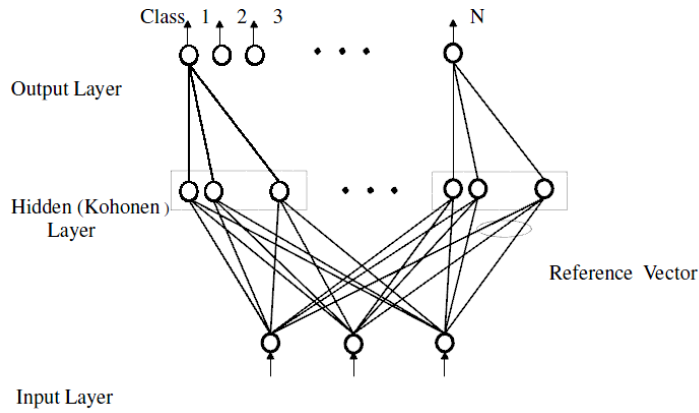
Sınıflandırma problemi için geliştirilmiş olan LVQ modelinin temel felsefesi, n boyutlu bir vektörü bir vektörler setine haritalamaktır. LVQ ağı, tamamen gizli bir şekilde bulunmaktadır. Her çıkış işlemci elemanı farklı bir gizli işlemci eleman kümesine bağlıdır. Gizli ve çıkış işlemci elemanları arasındaki ağırlıklar 1’e sabitlenmiştir.

Kohonen katmanı üzerine kurulmuştur. Bu katman; maddeleri, benzer objelerin bulunduğu uygun kategorilere ayırma özelliğine sahiptir.

LVQ ağının eğitilmesindeki amaç, her iterasyonda girdi vektörüne en yakın referans vektörünü bulmaktır. Referans vektörleri, Kohonen katmanındaki proses elemanlarını girdi katmanındaki proses elemanlarına bağlayan ağırlık değerleridir. Ağın öğretilmesi esnasında bunlar yeniden değer alırlar. Öğrenme esnasında sadece referans vektörlerin ağırlık değerleri değiştirilir. Girdi katman ile ara katman tam bağlantılı, ara katman ile çıktı katmanı kısmi bağlantılıdır. LVQ ağı Kohonen öğrenme kuralı denilen kazanan hepsini alır yaklaşımını kullanır.

Hem gizli işlemci elemanlar hem de çıkış işlemci elemanları ikili (binary) çıkışa sahiptir. Ağa bir giriş deseni verildiğinde referans vektörü “1”, diğerleri “0” üretir. “1” üreten çıkış işlemci elemanı giriş işaretini sınıflar ve her işlemci eleman ayrı bir sınıfa atanmış olur.

LVQ ağının bazı eksiklikleri vardır. Benzer nesneler veya girdi vektörleri ile ilgili karmaşık sınıflandırma problemlerinde ağ, her bir sınıf için pek çok işlem elemanına sahip geniş bir Kohonen katmanı gerektirir.



Şekil 2. 12 LVQ öğrenme vektör parçalama ağı

LVQ ağları, bazı işlem elemanlarının çok sık kazanmaya eğilimli olması, bazılarının ise etki itibarıyla hiçbir şey yapmaması gibi bir eksiklikten kötü yönde etkilenmektedir. Bu durum özellikle işlem elemanları eğitim vektörlerinden uzakta başladıklarında meydana gelir. Burada bazı elemanlar çok çabuk yakına çekilir ve diğerleri sürekli olarak uzakta kalır.

LVQ ağı, kazanan işleme elemanı yanlış sınıfta olduğunda ve ikinci en iyi işleme elemanı doğru sınıfta olduğunda yanlış sonuç verir. Eğitim vektörünün bu iki işleme elemanını birleştiren boşluğun orta noktası yakınlarında bulunması gerektiğinden, bu daha fazla sınırlama olur. Kazanan yanlış işleme elemanı eğitim vektöründen uzaklaşır ve ikinci sıradaki eleman eğitim vektörüne doğru hareket eder.

2.10 Elman ve Jordan Ağları

Bu ağlar, MLP' lere benzer bir yapıdadır ve çok katlıdırlar. Her iki ağda da gizli katmana ek olarak diğer bir “durum” katmanı denilen özel bir gizli katman daha vardır. Bu katman, gizli katmandan veya çıkış katmanından geri besleme işaretleri alır. Jordan ağının aynı zamanda durum katmanındaki her işlemci elemandan kendisine bağlantıları vardır. Her iki ağda da durum katmanındaki işlemci elemanların çıkışları ileriye doğru gizli katmana verilmektedir. Eğer sadece ileri doğru bağlantılar göz önüne alınır ve geri besleme bağlantılarına sabit değerler verilirse, bu ağlar sıradan ileri beslemeli ağlar haline gelirler.

2.11 MATLAB Programı (Matrix Laboratory)

Matlab matris işlenmesine, fonksiyonlara, veri çizilmesine, algoritma uygulanmasına, kullanıcı arayüzü oluşturulmasına ve diğer dillerle yazılmış programlar ile etkileşim oluşturulmasına izin veren bir bilgisayar programıdır. Matlab programında yapay sinir ağlarına yönelik çalışmalar yapılabilmektedir.

Tez çalışmasında Matlab R12 versiyonunda newff komutu kullanıldı. Bu komutun çalışmasını örneklerle açıklamanın yararlı olacağı düşünüldü.

Matlab R12’de çok katmanlı algılayıcı ağı oluşturmanın birkaç yolu vardır. Bunlardan bir tanesi newff komutu ile ileri beslemeli ağ nesnesi oluşturmaktır.

Bu komutun yazılışı aşağıdaki gibidir:

```
net = newff(PR,[S1 S2...SN],{TF1 TF2...TFN1},BTF,BLF,PF)
```

Komutta geçen ifadelerin tanımı:

- **PR:** R elemanlı giriş vektörünün minimum ve maksimum değerlerini içeren RX2 ‘lik matristir. Önceki bölümlerde anlatıldığı üzere aktivasyon fonksiyonlarının özelliğinden dolayı yapay sinir ağı normalize veriler kullanır. Veriler [-1,1] aralığında olmalı. Newff fonksiyonu verileri kendiliğinden normalize eder. Bunu yapabilmesi için tanımlanan matrisin her satırını ayrı ayrı değerlendirip minimum ve maximum değerleri yazılmalı veya bunun için minmax komutu kullanılmalı. Birinci sütun minimum, 2. sütun maksimumdur. R satırlı veri matrisinin olduğu varsayılırsa RX2’lik matris elde edilir. Minmax komutu kullanıldığında RX2 matris otomatik oluşur. Minmax komutu min ve max şeklinde 2 sütundan oluşan bir matris oluşturur. Verilen matrisin sırayla satırlarına bakar. İlgili satırın min değeri ilk sütuna, max değeri 2. sutuna yazılır.

A=

16	3	2	13
5	10	11	8
9	6	7	12
4	15	14	1

A=minmax(A)

A’nın son hali

2	16
5	11
6	12
1	15

- **Si:** İ'nci katmanda bulunan nöron sayısı.
- **TFi:** İ'nci katmanın transfer fonksiyonu, varsayılan '**tansig**'.
Burada seçilebilecek fonksiyonlar '**logsig**', '**tansig**' ve '**purelin**'dir. Gizli katmanda lineer olmayan fonksiyonlar '**logsig**' veya '**tansig**' kullanılmalı. Çıkış katmanında lineer fonksiyon da kullanılabilir.
- **BTF:** Çok katmanlı ağ eğitim algoritması, varsayılan '**trainlm**'dir.
Eğitim algoritması olarak seçilebilecek birçok fonksiyon bulunmaktadır. Bunlardan birkaçı şöyledir:

'**traingd**' gradyent azaltım metodunu kullanan eğitim algoritması

'**traingdm**' momentum değişkeni içeren eğitim algoritması

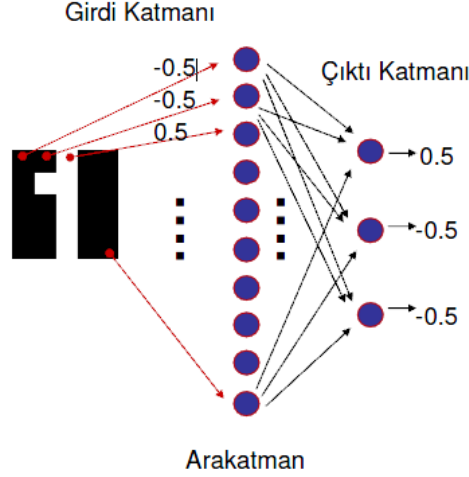
'**trainlm**' Levenberg Marquardt minimizasyon metotlu eğitim algoritması
- **BLF:** Geriye yayılım ağırlık/bias öğrenme fonksiyonu, varsayılan '**learnngdm**'dir.
'**learnngd**' gradyent azaltım metotlu öğrenme algoritması
'**learnngdm**' momentum değişkenli öğrenme algoritması
- **PF:** Performans fonksiyonu, varsayılan '**mse**' dir.

Newff komutu ağ nesnesi oluşturur ve aynı zamanda ağırlık ve bias değerlerinin ilklendirme (initialization) işlemini rastgele değerler atamak suretiyle gerçekleştirir. Haliyle ağ eğitime hazır olarak beklemektedir. Eğitim için train komutu, simulasyon için sim komutu kullanılır. Similasyon'dan kasıt, ağa bir giriş değeri verip ağın çıkışını hesaplatmaktır.

P giriş matrisini oluştururken daha sonra transpoze alınacağı göz önünde tutularak sütunlar değişken, satırlar ise o değişkenlere ait örnekleri içermelidir. Matrisin isminin üstüne kesme işareti konulursa A' gibi bu transpoze alınması anlamına gelir.

2.11.1 Örnek Program 1

5X5 boyutunda (25 değişkenli) 3 örnek karenin ağa tanıtılması ve ağın bu kareleri sınıflandırması.



Şekil 2. 13 25 girişli 3 çıkışlı örnek MLP ağı

```
%%%%%%%%%
```

```
clear all; close all; clc;
```

```
P = [-0.5 -0.5 0.5 -0.5 -0.5 -0.5 0.5 0.5 -0.5 -0.5 -0.5 -0.5 0.5 -0.5 -0.5 -0.5 0.5 -0.5  
-0.5 -0.5 -0.5 0.5 -0.5 -0.5;...  
-0.5 0.5 0.5 0.5 -0.5 0.5 -0.5 -0.5 -0.5 0.5 -0.5 -0.5 -0.5 0.5 -0.5 -0.5 -0.5 0.5 -0.5  
0.5 0.5 0.5 0.5 0.5;...  
-0.5 0.5 0.5 0.5 0.5 -0.5 -0.5 -0.5 0.5 -0.5 -0.5 -0.5 0.5 0.5 -0.5 -0.5 -0.5 -0.5 0.5 -  
0.5 0.5 0.5 0.5 -0.5]'  
T = [[0.5;-0.5;-0.5] [-0.5;0.5;-0.5] [-0.5;-0.5;0.5] ]
```

```
net=newff(minmax(P),[10,3],{'logsig','purelin'},'trainrp');  
net.trainParam.show = 50; %% kaç iterasyonda bir eğitim durumu ekrana gelecek.  
net.trainParam.epochs = 400; %% iterasyon sayısı  
net.trainParam.goal=1e-7; %% hedef  
[net,tr]=train(net,P,T);
```

```
c = [ -0.5 -0.5 0.5 -0.5 -0.5 -0.5 0.5 0.5 -0.5 -0.5 -0.5 -0.5 0.5 -0.5 -0.5 -0.5 0.5 -0.5  
-0.5 -0.5 0.5 0.5 0.5 -0.5 ]'
```

```
Y = sim(net,c);
```

```
msgbox(num2str(Y),'SONUC')
```

```
%%%%%%%%%
```

Örnek program 1’de 25 değişkenli 3 örnek P matrisi olarak tanımlanıyor. Boyutu ise 3X25 şeklinde. Yani başlangıçta matrisin satırları örnekleri, sütunları ise değişkenleri ifade etmektedir.

P matrisinin sonunda ' işareti olduğuna dikkat ediniz. Bu işaret Matlab'da transpoze anlamına gelir ve satır ile sütunların yer değiştirmesi işlemidir. P'nin transpozesi alınınca 25X3 boyutlu matrise dönüşüyor ve artık satırlar değişkenleri, sütunlar ise örnekleri temsil etmektedir.

Normalde T sonuç matrisinin de ilk başta 3X3 boyutunda satırları örnek, sütunları çıktı değişkeni olması gerekir. Newff komutuna almak için T'nin transpozesi alınmalı ve satırları değişken sütunları örnek şeklinde yeni 3X3'lük matris oluşturulmalı. Programda T'nin transpozesi alınmış şekilde veriler tanıtılmış o nedenle tekrar transpoze alınmıyor. Bu arada eğer tek çıktı yani tek değişken olsaydı sonuç matrisinde tek satır olacaktı. Örnek sonuçları yan yana aynı satırda yazılacaktı.

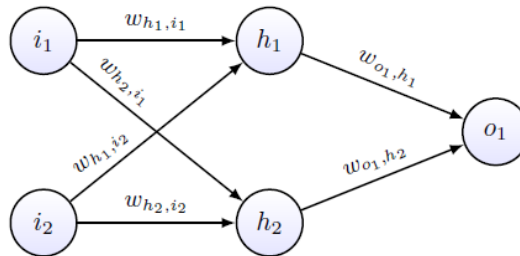
Newff komutunda normalizasyon işlemine gerek yoktur verilerin minimum ve maksimum değerleri yazıldığında komut otomatik olarak verileri normalize hale getirmektedir. Sonucu ise kullanıcıya normalizeden çıkararak gerçek değerde verir. Ancak özellikle çok veri ile çalışıldığında excelde normalize edilen veriler Matlab programına verilmelidir.

Yukarıdaki programla (25,10,3) yapısında yani 25 giriş nöronlu,10 gizli nöronlu ve 3 çıkış nöronlu bir MLP tasarlanmıştır. İterasyon sayısı ve hedef belirtilerek train komutuyla ağ eğitilmiştir.

Daha sonra 25 değişkenli başka bir örnek YSA'ya verilmiş ve çıktı elde edilmiştir.

2.11.2 Örnek Program 2

Çizelge 2.2'de giriş ve çıkışları gösterilen 2 girişli 1 çıkışlı XOR problemini çözen MLP tasarımının yapılması.



Şekil 2. 14 XOR problemi MLP ağı

```

%%%%%%%%%%
clear all;close all;clc;

A=[0 0;0 1;1 0;1 1]; B=[0;1;1;0];
A=A'; B=B';

net=newff(minmax(A),[5,1],{'logsig','purelin'},'trainlm','learnngdm','sse');
net=train(net,A,B);

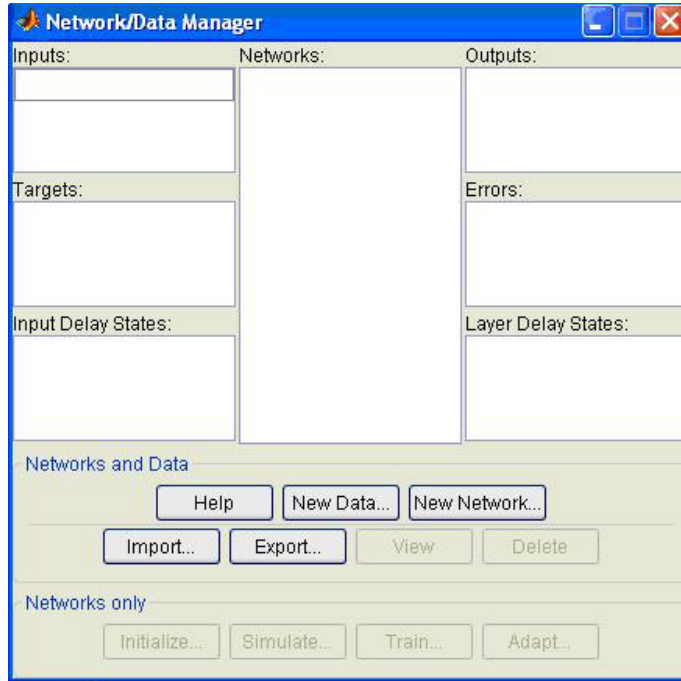
net.trainParam.show = 50;
net.trainParam.lr = 0.05;
net.trainParam.epochs = 300;
net.trainParam.goal = 1e-5;

%%%%%%%%%%

```

XOR problemi için (2,5,1) yapısında bir MLP ağı tasarlanmıştır.

Matlab programında nntool yardımıyla yapay sinir ağı tasarlanabilmektedir. Bunun için command penceresine nntool yazıldığında ekran gelecektir. Bu ekranda girdi değerleri, çıktı değerleri ve ağ yapısı tanıtılarak tasarım yapılabilir.



Şekil 2. 15 Matlab NNTOOL penceresi

UYGULAMA

3.1 Uygulama Yapılan Firma Hakkında Bilgi

İmalat sektöründe yapay sinir ağları uygulamaları konulu bu tez çalışmasında, kariyer hayatımın başlangıcı olan Malkan Makina Ltd. Şirketi'nde yapay sinir ağları metoduyla talep tahmini yapıldı. Şirkette yapısal bir üretim planlama birimi bulunmamaktadır. Üretim, çoğunlukla sipariş odaklıdır. Makine satış hedefi olarak pazarlama bölümü, önceki sene satış verilerine bakıp tecrübelerine de dayanarak yeni sene nihai ürün talep tahmini yapmaktadır. Bu hedef ise sadece pazarlama bölümünde kalmakta üretim için bir anlam ifade etmemektedir. Ana üretim planı mevcut değil. Şirket için öncelikli ihtiyaç, nihai ürün talep tahmin çalışmasıdır.

Malkan Makina sanayi tipi ütü makineleri üretmektedir. Müşteri portföyüne bakıldığında çoğunlukla tekstil sektöründe faaliyet gösteren firmalar, kuru temizleyiciler bulunmaktadır. Bunun dışında buhar kazanları satılan oteller, hastaneler, kuyumcular da firmanın müşterileri arasında yer almaktadır. Üretilen makine modeli 300'ün üzerindedir. Firmanın ürettiği makineler grup bazında Çizelge 3. 1'de görülmektedir.

Firma yöneticileri satışlarını etkileyen birçok belirsiz parametre olduğu, bu nedenle satışlarının hiçbir modele uymayacağı konusunda aynı fikirdeler. Firmanın müşterilerine verdiği uzun termin süresinden dolayı kaybettiği siparişler de bulunmaktadır.

Firmanın makine satışlarını etkileyen durumlar:

- Trend etkisi. Kot moda olduğunda kot ütüleme makinelerinin satılması.
- Mevsimsel dalgalanmalar. Yazın gömlek ütüleme, yapıştırma tela makinelerinin çok satılması
- Devlet teşviğinin olduğu bölgelerde tekstil firmalarının yatırım kararı alması
- Müşterilerin finansal durumu
- Satılan makinelerin ortalama ömürlerini tamamlamaları. Müşterinin yeni makine alımına karar vermesi
- Askeriyenin alımları
- İnovasyon etkisi
- Fuarlar. Fuar olan ay diğer aylara göre daha çok satış olmaktadır.

Çizelge 3. 1 Üretilen makine grupları

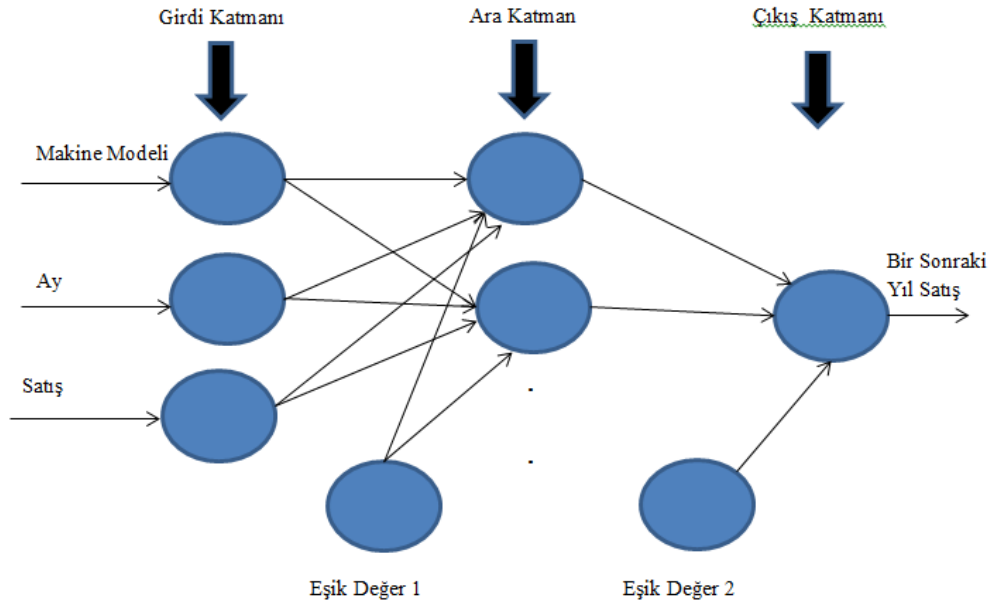
Grup	Makine Grup Adı
1	Vakumlu Ütü Paskalaları
2	Vakumlu Ütü Masaları
3	Kendinden Kazanlı Tezgâhlar
4	Seyyar ve Merkezi Sistem Buhar Jeneratörleri
5	Mekanik Ütü Presleri
6	Yapıştırma ve Transfer Baskı Presleri
7	Buharlı Şişirme Ütüleri
8	Pnömatik Ütü Presleri
9	Merkezi Sistem Tezgâhlar
10	Leke Çıkarma Makineleri

3.2 Farklı Yapıda MLP Yapay Sinir Ağlarının Kurulması

Yapay sinir ağlarında tahmin amacıyla genellikle MLP (Çok Katmanlı Algılayıcı) ağı kullanılmaktadır. Bu çalışmada MLP ağı kullanıldı. Oluşturulan ağ 3 girişli ve bir çıkışlıdır. Girdi parametleri olarak;

- Ürün modeli (1'den 10'a kadar Çizelge 3.1' de grup isimleri görülmektedir.)
- Satış ayı (1-12 arası değer alır. Örneğin 5, mayıs ayını ifade eder.)
- Belirlenen ürün modelinin o aydaki satış miktarı

seçilmiştir. Seçilen ürün modelinin bir sonraki sene aynı ayındaki satış miktarı olarak tanımlanmıştır. Tasarlanacak ağı sırasıyla ürün grubu, ilgili ay ve satış değerleri verilerek; ağın bir sonraki ilgili ayda satış tahmin etmesi amaçlanmaktadır. Şekil 3. 1'de oluşturulan ağın yapısı gösterilmiştir.



Şekil 3. 1 Tasarlanan MLP ağ yapısı

Çizelge 3.2' de programa girişi yapılan eğitim/ test verilerinden örnek gösterilmektedir. Çizelgedeki ilk örneğe göre 1. ürün gurup modelinden ocak ayında 79 adet, bir sonraki ocak ayında ise 35 adet satılmıştır. 1. Örnekte giriş 1 1 79, çıkış 35'tir.

Yapay sinir ağları 1-(0) veya 1 (-1) aralığında çalışmaktadır. Bu nedenle veriler öncelikle normalize edilir. Normalize formülü olarak 3.1 kullanıldı. Normalize veriler kullanıldığı için ağın çıktısı da normalize edilmiş olacaktır. Çıktı verilerin gerçek değerine ulaşmak için programda gerekli düzenleme yapılmalıdır.

$$V_n = 0,8 \cdot \left[\frac{V_r - V_{min}}{V_{max} - V_{min}} \right] + 0,1 \quad (3.1)$$

Çizelge 3. 2 Programa girişı yapılan örnek veri

GERÇEK GİRDİLER					ÇIKTI			
	ÜRÜN GRUBU	AY	SATIŞ	YENİ SATIŞ	ÜRÜN GRUBU	AY	SATIŞ	YENİ SATIŞ
1. Örnek	1	1 (Ocak)	79	35	0,1000	0,1000	0,4129	0,2386
2. Örnek	2	2 (Şubat)	82	130	0,1889	0,1727	0,2980	0,2584
3. Örnek	3	3 (Mart)	20	38	0,2778	0,2455	0,2030	0,1792
4. Örnek	4	4 (Nisan)	45	53	0,3667	0,3182	0,4089	0,2861
5. Örnek	5	5 (Mayıs)	17	12	0,4556	0,3909	0,1396	0,1198
6. Örnek	6	6 (Haziran)	4	12	0,5444	0,4636	0,1436	0,1158
7. Örnek	7	7 (Temmuz)	3	6	0,6333	0,5364	0,1554	0,1119
8. Örnek	8	8 (Ağustos)	14	11	0,7222	0,6091	0,1277	0,1238
9. Örnek	9	9 (Eylül)	36	26	0,8111	0,6818	0,1119	0,2426
10. Örnek	10	10 (Ekim)	3	15	0,9000	0,7545	0,1515	0,1158

Çizelge 3.2’de gösterilen normalize hale getirilen örnek verilerin ondalık kısmı virgöl ile ayrılmıştır. Matlab R12 programında ise sayıların ondalık kısmı nokta ile ayrılmaktadır. Bu nedenle verilerdeki virgüller noktaya çevrilir. Çizelgedeki normalize başlıklı kısmın 0,1 verisinden başlanarak 4 sütun ve 400 (eğitim örnek adedi) satır seçilip metin belgesi (.txt) türünde kaydedilir. Bunlar eğitim verilerini oluşturur. 400X4 matris boyutundadır. Sütunlar sırasıyla ürün grubu, ay, satış ve yeni satıştır. Eğitim veri seti Ek-A’da gösterilmiştir.

En güncel 2012 Mayıs – Aralık peryoduna ait 80 adet test verisi için de aynı işlemler yapılır. 80X4 matris boyutunda test verisi elde edilir. Test veri seti Ek-B’de gösterilmiştir.

Böylece tasarlanacak yapay sinir ağı için eğitim ve test verileri hazır hale gelir. Bu dosyalar Matlab R12 programında m.file şeklinde program yazımında ‘load’ komutuyla dosya ismiyle programa yüklenir.

Matlab R12’de talep tahmini amacıyla yapay sinir ağı tasarlamak için m.file oluşturulmuştur.

Talep tahmini problemine en iyi çözümü bulan ağ yapısını tespit etmek için;

- Gizli nöron sayısı,
- Aktivasyon fonksiyonu,
- Eğitim algoritması

üzerinde çeşitli değişiklikler yapılarak duyarlılık analizi ayrı ayrı yapıldı.

Çizelge 3. 3 Gizli nöron sayısı için duyarlılık analizi

No	Ağ Yapısı	Fonksiyon ve Öğrenme Algoritması	MSE	Sonuç
1	3,7,1	logsig' 'purelin','trainlm','learngdm','mse'	0.00322293/0	
2	3,8,1	logsig' 'purelin','trainlm','learngdm','mse'	0.00287374/0	
3	3,9,1	logsig' 'purelin','trainlm','learngdm','mse'	0.00275698/0	Seçilen Ağ Yapısı
4	3,11,1	logsig' 'purelin','trainlm','learngdm','mse'	0.00243763/0	Uyumsuz
5	3,15,1	logsig' 'purelin','trainlm','learngdm','mse'	0.00222388/0	Uyumsuz

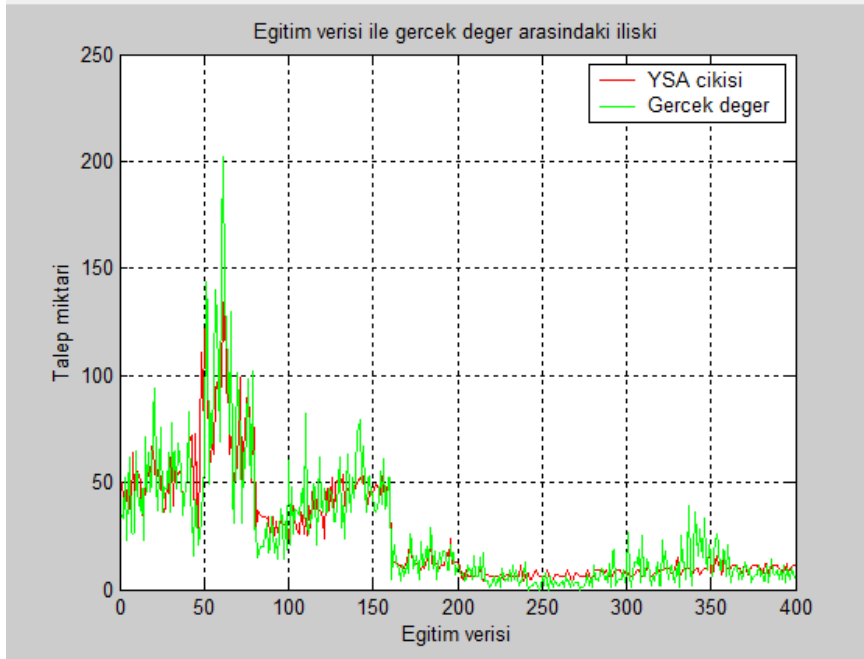
7, 8, 9, 11, 15 adet gizli nörona göre ağ yapısı sorgulanarak duyarlılık analizi yapıldı. Çizelge 3.3’te görülmektedir. Ağ tasarımı, duyarlılık analizinde 9 gizli nörona sahip yapının seçilmesine karar verildi. Seçilen bu yapı minimum hatayı vermemektedir. Ancak çalışmamızın devamında diğer parametlerle birlikte problem değerlendirildiğinde birinci katmanda 9 gizli nöronun optimum çözümü verdiği görülmüş ve bu nedenle seçilmiştir.

Çizelge 3. 4 Fonksiyon ve eğitim algoritması için duyarlılık analizi

No	Ağ Yapısı	Fonksiyon ve Öğrenme Algoritması	MSE	Sonuç
1	3,9,1	logsig' 'purelin','trainlm','learngdm','mse'	0.00296912/0	min. MSE
2	3,9,1	logsig' 'purelin','trainbfg','learngdm','mse'	0.00389429/0	
3	3,9,1	logsig' 'purelin','trainrp','learngdm','mse'	0.00452462/0	
4	3,9,1	tansig' 'purelin','trainbfg','learngdm','mse'	0.00488269/0	
5	3,9,1	tansig' 'purelin','trainrp','learngdm','mse'	0.00523543/0	
6	3,9,1	tansig' 'purelin','trainlm','learngdm','mse'	0.00313225/0	
7	3,9,1	logsig' 'logsig','trainbfg','learngdm','mse'	0.0043698/0	
8	3,9,1	logsig' 'logsig','trainlm','learngdm','mse'	0.00299124/0	
9	3,9,1	logsig' 'logsig','trainrp','learngdm','mse'	0.00550057/0	
10	3,9,1	tansig' 'tansig','trainlm','learngdm','mse'	0.00300116	
11	3,9,1	tansig' 'tansig','trainbfg','learngdm','mse'	0.00436302/0	
12	3,9,1	tansig' 'tansig','trainrp','learngdm','mse'	0.00508331/0	

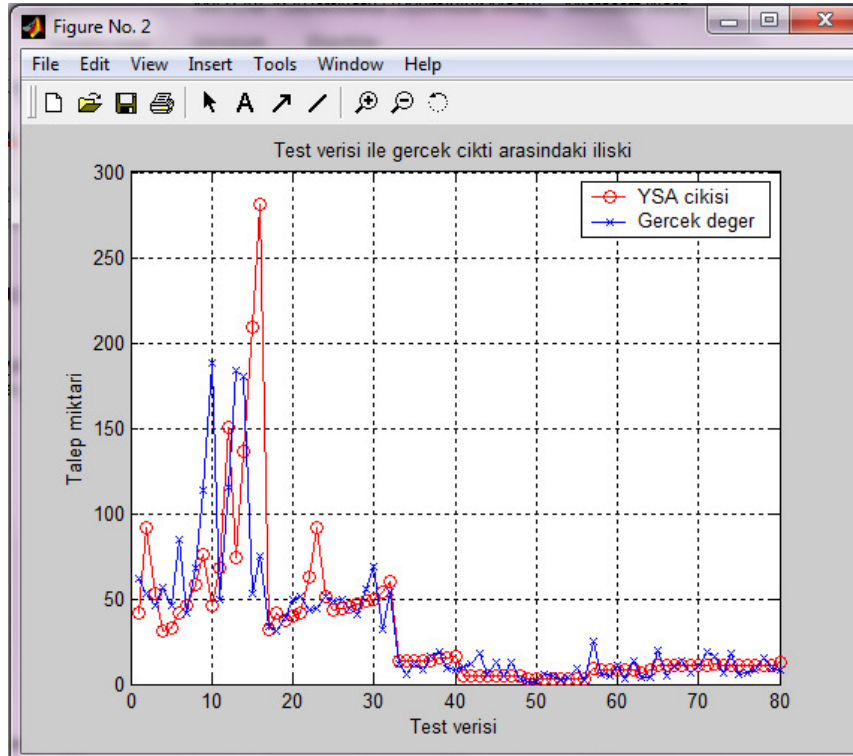
Giriş katmanında 3 nöron, gizli katmanda 9 nöron ve çıkış katmanında 1 nöron bulunan (3,9,1) yapısındaki sinir ağı için aktivasyon fonksiyonu ve eğitim algoritması duyarlılık analizleri yapıldı. Aktivasyon fonksiyonları olan logsig, tansig ve purelin fonksiyonlarının hepsi değişik kombinasyonlarla denendi.

Eğitim algoritmalarından ise, trainlm (geri yayılım algoritması), trainrp (esnek yayılım algoritması) ve trainbfg kullanılarak ağı duyarlılığı test edilmiştir. Sonuçlar Çizelge 3.4'te gösterilmektedir. Elde edilen sonuçlar incelendiğinde gizli katmanda “logsig” aktivasyon fonksiyonu, çıkış katmanında “purelin” aktivasyon fonksiyonu olan ve 'trainlm' eğitim algoritmasıyla çalışan ağı minimum MSE'yi verdiği görülmüştür. Ağı yapısının (3,9,1) logsig' 'purelin'}, 'trainlm','learngdm','mse' gibi olmasına karar verilmiştir.



Şekil 3. 2 'logsig', 'purelin','trainlm','learngdm','mse' (3,9,1) ağı eğitim

Şekil 3.2’de seçilen (3,9,1) logsig' 'purelin','trainlm','learngdm','mse' yapısındaki ağıın eğitim sonucu 400 örnek için ayrı ayrı gösterilmiştir. YSA tahminleri kırmızı, gerçek değerler yeşil renkli çizilmiştir. Eğitim durumunda, ağı girdi değişken değerleri ve bunun karşılığı çıkması gereken çıkış değerleri tanıtılır. Eğitim verileri ışığında, ağıın verilen girdilere karşılık çıkışları doğru bir şekilde tahmin etmesi için ağıdaki ağırlıklar kendiliğinden güncellenir. Böylece tasarlanan ağı öğrenme eylemini gerçekleştirmiş olur. Ağıın tahmin ettiği sonuç ile gerçekte olması gereken sonuç çizimlerden de görüleceğı üzere birbirine yakındır.



Şekil 3. 3 'logsig' 'purelin','trainlm','learngdm','mse' (3,9,1) ağı test

Tasarlanan ve eğitime tabii tutulan ağ artık test aşamasına gelmiştir. Sinir ağına eğitim esnasında hiç görmediği 3 girdi değişkeninden oluşan 80 örnek verilmiş ve ağın bu örneklerle ait çıktıyı tahmin etmesi beklenmiştir. Şekil 3.3'te gösterilen test sonuçlarından YSA tahminleri kırmızı, gerçekte olması gereken değer ise mavidir.

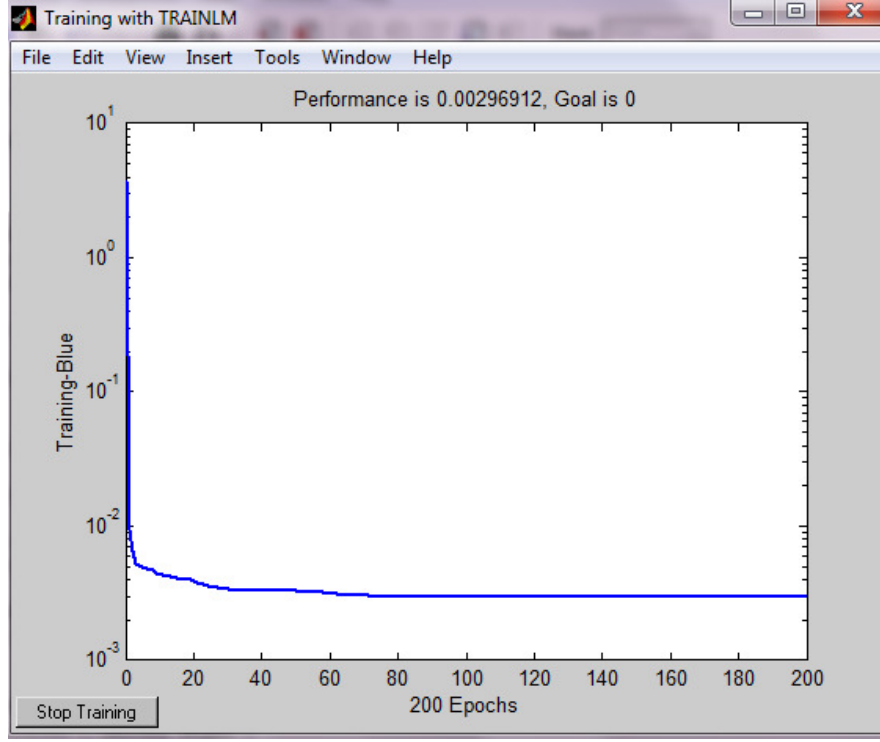
Test verilerinin büyük çoğunluğunun örtüştüğü görülmektedir. Ağın tahmin ettiği sonuç ile gerçekte olması gereken sonuç çizimlerden de görüleceği üzere arada sapmalar olmasında rağmen birbirine yakındır. Çizelge 3.5'te test verileri, ağın tahminleri ve aradaki fark gösterilmiştir.

Çizelge 3. 5 Ağın test sonuçları

YIL	ÜRÜN GRUBU	AY	SATIŞ GERÇEKLEŞEN	AĞIN TAHMİNİ	FARK
2012	1	5	62	45	-17
2012	1	6	53	65	12
2012	1	7	46	37	-9
2012	1	8	57	58	1
2012	1	9	46	47	1
2012	1	10	85	57	-28
2012	1	11	42	51	9
2012	1	12	68	56	-12
2012	2	5	114	81	-33
2012	2	6	188	25	-163
2012	2	7	50	94	44
2012	2	8	115	101	-14
2012	2	9	184	129	-55
2012	2	10	180	142	-38
2012	2	11	53	138	85
2012	2	12	75	114	39
2012	3	5	34	34	0
2012	3	6	31	1	-30
2012	3	7	39	32	-7
2012	3	8	50	31	-19
2012	3	9	51	39	-12
2012	3	10	43	40	-3
2012	3	11	44	49	5
2012	3	12	51	51	0
2012	4	5	49	44	-5
2012	4	6	50	45	-5
2012	4	7	44	47	3
2012	4	8	41	48	7
2012	4	9	56	56	0
2012	4	10	69	69	0
2012	4	11	32	32	0
2012	4	12	54	54	0
2012	5	5	12	11	-1
2012	5	6	6	10	4
2012	5	7	12	12	0
2012	5	8	8	13	5
2012	5	9	16	14	-2
2012	5	10	19	13	-6
2012	5	11	9	14	5
2012	5	12	8	20	12
2012	6	5	10	7	-3

Çizelge 3.5 Ağın test sonuçları (devamı)

YIL	ÜRÜN GRUBU	AY	SATIŞ GERÇEKLEŞEN	AĞIN TAHMİNİ	FARK
2012	6	6	12	7	-5
2012	6	7	18	6	-12
2012	6	8	6	6	0
2012	6	9	13	6	-7
2012	6	10	5	6	1
2012	6	11	13	6	-7
2012	6	12	3	6	3
2012	7	5	1	5	4
2012	7	6	1	7	6
2012	7	7	6	7	1
2012	7	8	5	6	1
2012	7	9	2	6	4
2012	7	10	4	6	2
2012	7	11	9	6	-3
2012	7	12	2	6	4
2012	8	5	25	10	-15
2012	8	6	6	6	0
2012	8	7	5	8	3
2012	8	8	11	11	0
2012	8	9	3	8	5
2012	8	10	14	6	-8
2012	8	11	4	6	2
2012	8	12	4	6	2
2012	9	5	20	10	-10
2012	9	6	5	10	5
2012	9	7	10	7	-3
2012	9	8	14	11	-3
2012	9	9	7	16	9
2012	9	10	10	14	4
2012	9	11	19	7	-12
2012	9	12	16	6	-10
2012	10	5	7	11	4
2012	10	6	18	10	-8
2012	10	7	6	10	4
2012	10	8	7	8	1
2012	10	9	8	9	1
2012	10	10	15	10	-5
2012	10	11	9	11	2
2012	10	12	8	8	0



Şekil 3. 4 'logsig' 'purelin', 'trainlm', 'learnqdm', 'mse' (3,9,1) ağı öğrenme eğrisi

Test aşamasından sonra ağı tahmin ederek bulduğu sonuçlar ile gerçek sonuçlar karşılaştırılarak ağı performansı bulunur. Performans ölçütü olarak tasarlanan ağda hata karelerinin ortalaması olan MSE kullanılmıştır. Ağda hesaplanan MSE yaklaşık olarak 0,003'tür.

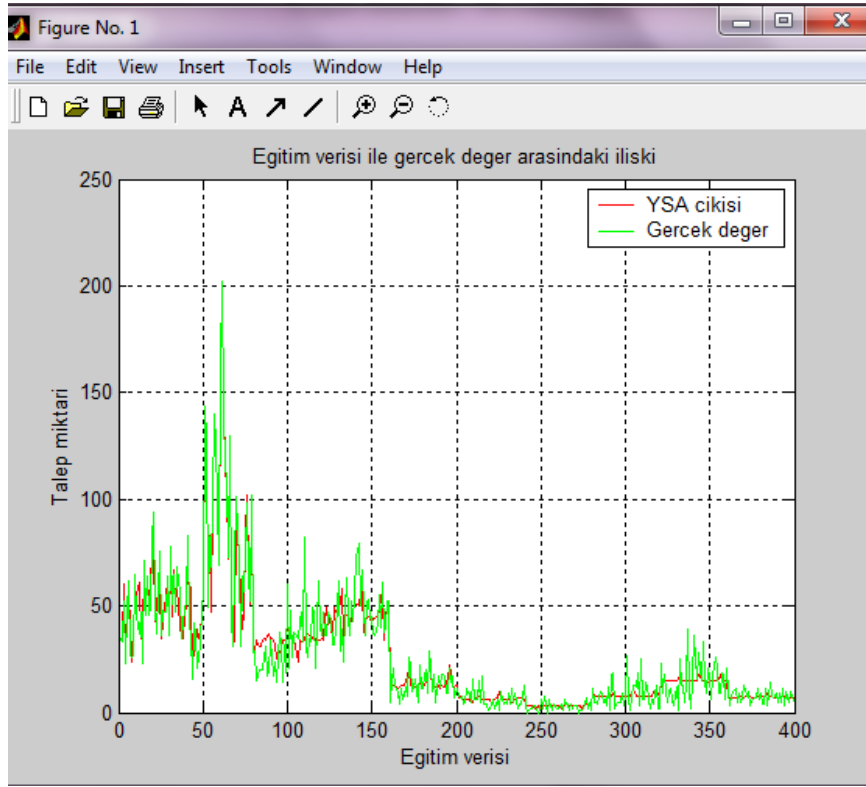
Tek gizli katmanlı ağ için çeşitli denemeler yapılarak 3,9,1 yapısındaki ağ 0,003 MSE ile seçildi. 2 gizli katmanlı bir ağ tasarlanırsa acaba hata azaltılabilir mi? Bu sorunun cevabı için 2 gizli katmanlı ağ tasarlanarak performans değerleri karşılaştırıldı. Çizelge 3.6'da sonuçlar görülmektedir.

Çizelge 3. 6 İkinci gizli katmandaki gizli nöron sayısı için duyarlılık analizi

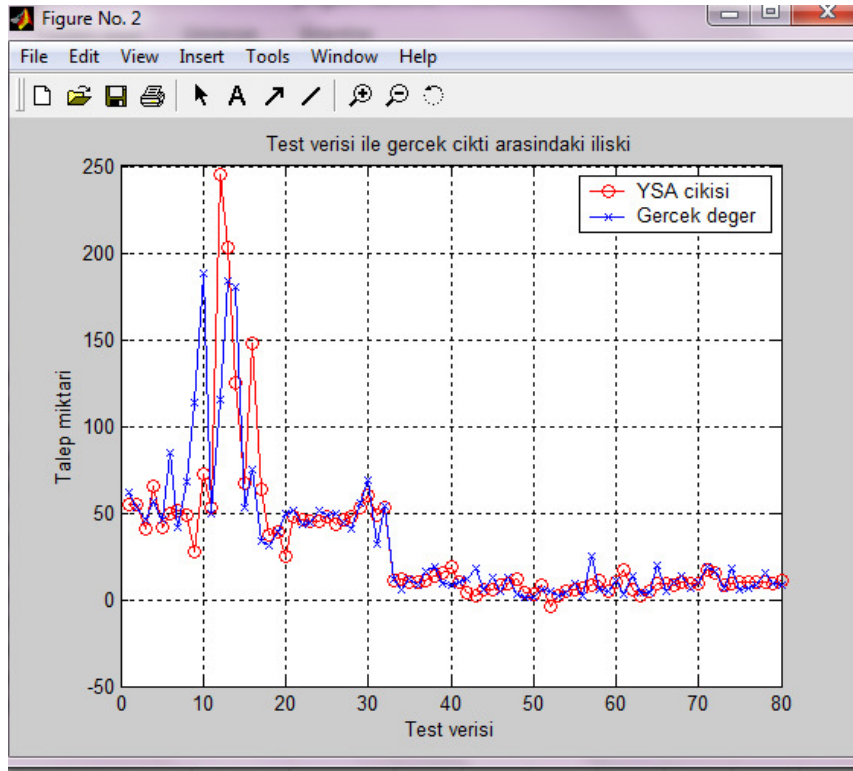
No	Ağ yapısı	Fonksiyon ve Öğrenme Algoritması	MSE	Sonuç
1	3,9,2,1	'logsig', 'tansig', 'purelin', 'trainlm', 'learngdm', 'mse'	0.0021719/0	
2	3,9,2,1	'logsig', 'logsig', 'purelin', 'trainlm', 'learngdm', 'mse'	0.00196337/0	
3	3,9,3,1	'logsig', 'logsig', 'purelin', 'trainlm', 'learngdm', 'mse'	0.00254948/0	
4	3,9,4,1	'logsig', 'logsig', 'purelin', 'trainlm', 'learngdm', 'mse'	0.00140099/0	
5	3,9,4,1	'logsig', 'tansig', 'purelin', 'trainlm', 'learngdm', 'mse'	0.00156592/0	
6	3,9,4,1	'logsig', 'tansig', 'purelin', 'trainbfg', 'learngdm', 'mse'	0.00453015/0	
7	3,9,5,1	'logsig', 'logsig', 'purelin', 'trainlm', 'learngdm', 'mse'	0.00126267/0	
8	3,9,5,1	'logsig', 'tansig', 'purelin', 'trainlm', 'learngdm', 'mse'	0.00154322/0	
9	3,9,5,1	'logsig', 'tansig', 'purelin', 'trainbfg', 'learngdm', 'mse'	0.00497893/0	
10	3,9,6,1	'logsig', 'logsig', 'purelin', 'trainlm', 'learngdm', 'mse'	0.00104653/0	
11	3,9,7,1	'logsig', 'tansig', 'purelin', 'trainlm', 'learngdm', 'mse'	0.00188898/0	
12	3,9,8,1	'logsig', 'tansig', 'purelin', 'trainlm', 'learngdm', 'mse'	0.00090877/0	

2 gizli katmanı bulunan ağ yapıları MSE'ye göre incelendiğinde 10. denemede yer alan (3, 9, 6, 1) yapısındaki ağın 0,001 performans verdiği gözlemlenmiştir. Bu ağ yapısına ait sonuçlar şekillerde verilmiştir. Gizli nöron sayısı 8'e çıktığında hatanın daha da azaldığı görülmüş ancak ağın ezberleme ihtimalinden dolayı 2. gizli katmanda 6 nöron yeterli görülmüştür. 2. gizli katmanın aktivasyon fonksiyonu logsig olarak belirlenmiş. Seçtiğimiz ağ yapısı ve ilgili parametler şöyledir:

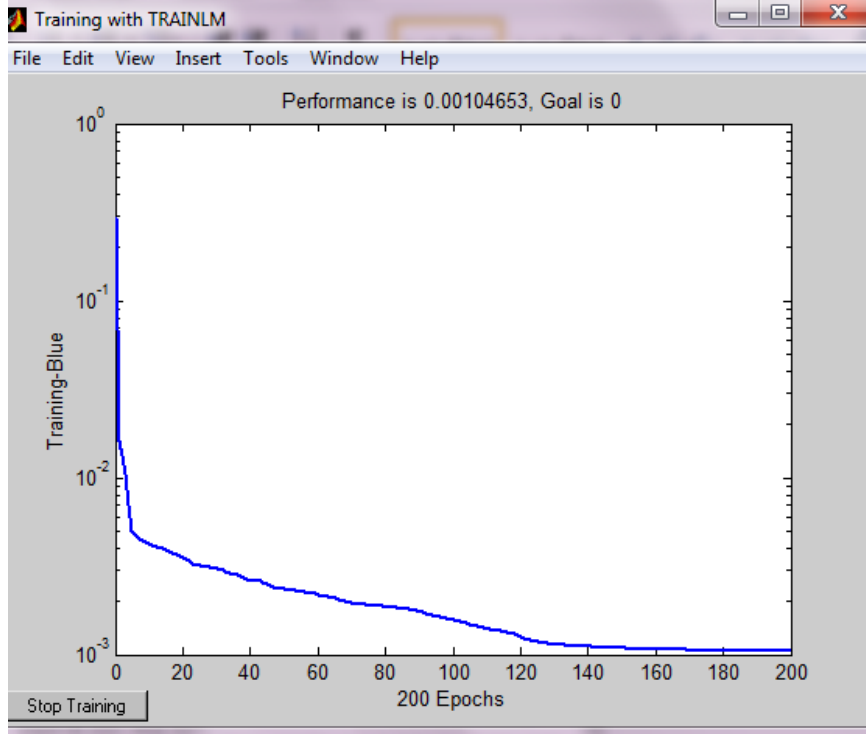
'logsig' 'logsig' 'purelin', 'trainlm', 'learngdm', 'mse' (3,9,6,1)



Şekil 3. 5 'logsig','logsig','purelin','trainlm','learngdm','mse' (3,9,6,1) ağı eğitim



Şekil 3. 6 'logsig' 'logsig','purelin','trainlm','learngdm','mse' (3,9,6,1) ağı test



Şekil 3. 7 'logsig' logsig 'purelin','trainlm','learngdm','mse' (3,9,6,1) ağı öğrenme eğrisi

Tasarlanan ağda tüm ürün grupları bir ağda işlem görmektedir. Acaba ürün gurupları bazında farklı ağlar olsa her ağ bir ürün grubunun tahminini yapsa daha iyi bir performans elde edilebilir mi?

Ay parametresi göz önüne alınmadan sadece satışların girdi olarak değerlendirildiği bir girdisi ve bir çıktısı olan yapay sinir ağı tasarlanarak vakumlu ütü paskalaları için bu ağda çeşitli denemeler yapıldı. Vakumlu ütü paskalalarının tüm satışları kendi arasında normalize edilerek programa girildi. Bir girdili ve bir çıktılı yapay sinir ağındaki denemelerin sonuçları çizelge 3.7'de gösterilmiştir. Ancak performans yani hata değerleri yüksek çıkmıştır. Bu nedenle talep tahmini için uygun olmadığına karar verildi. 1 girdili ve bir çıktılı olan ağda eğitim seti 40, test seti 8 örnekten oluşmuştur. Örneklerin az olması ağı öğrenmesini azaltabilir ve performansı etkileyebilir.

Çizelge 3. 7 Bir girişli ve bir çıkışlı ağı gizli nöron sayısı için duyarlılık analizi

No	Ağ yapısı	Fonksiyon ve Öğrenme Algoritması	MSE
1	1,9,1	'logsig','purelin','trainlm','learngdm','mse')	0.00582589/0
2	1,1,1	'logsig','purelin','trainlm','learngdm','mse')	0.00658378/0
3	1,2,1	'logsig','purelin','trainlm','learngdm','mse')	0.00652157/0
4	1,3,1	'logsig','purelin','trainlm','learngdm','mse')	0.00655973/0
5	1,3,1	'tansig','purelin','trainlm','learngdm','mse')	0.00651939/0
6	1,3,1	'tansig','tansig','trainlm','learngdm','mse')	0.00798493/0
7	1,4,1	'logsig','purelin','trainlm','learngdm','mse')	0.00636044/0
8	1,20,1	'logsig','purelin','trainlm','learngdm','mse')	0.00511896/0
9	1,10,5,1	'logsig','tansig','purelin','trainrp','learngdm','mse')	0.0161501/0
10	1,10,5,1	'logsig','tansig','purelin','trainbfg','learngdm','mse')	0.0110476/0

Tasarlanan yapay sinir ağı 3,9,6,1 yapısında 4 katmanlı bir MLP ağıdır.

- Girdiler: Makine modeli, ay, ilgili makine modeli satış
- Çıktı: Gelecek sene aynı aydaki makine satışı
- Gizli katmanlardaki aktivasyon fonksiyonu: logsig, logsig, purelin
- Eğitim algoritması: trainlm
- Öğrenme algoritması: learngdm
- Öğrenme katsayısı: 0,5
- Momentum katsayısı: 0,5
- İterasyon sayısı:200
- Her 100 iterasyonda hata ekrana yazdırılıyor.
- Eğitim hedefi: Minimum hata belirtilerek hata o aşamaya geldiğinde sistem durdurulabilir. Minimum hata arayışında olduğu için hedef kısıtı verilmedi.
- MSE 0,001.

3,9,6,1 yapısında, 2 gizli katmanlı ve logsig, logsig, purelin aktivasyon fonksiyonları ile çalışan yapay sinir ağıyla makine imalatı yapan bir firmanın talep tahmini yapılmıştır. 0,001 düzeyinde hata ile karşılaşılmıştır. Bu sinir ağının talep tahmini için beklentimizi karşıladığı görülmüş ve 0,001 hata ile ağ modeli kabul edilmiştir.

SONUÇ VE ÖNERİLER

Aralarında lineer olmayan ilişkilerin olduğu birçok değişkene bağlı problemlerin çözümünde beyin sinir sisteminden esinlenerek oluşturulan yapay sinir ağları metodu olumlu sonuç vermektedir. Önemli olan değişkenlerin doğru tanımlanması ve doğru öğrenmeyi sağlayacak bir ağ yapısının kurulmasıdır.

Her sistemde onu etkileyen nedenler vardır. Etkilere göre sistem farklı davranış sergilemektedir. Bu nedenler çoğu zaman bilinmese bile sistem ona göre bir çıktı üretir. Yapay sinir ağlarında, sistemi etkileyen parametrelerin ne yönde nasıl bir etki yaptığı sinir ağının yapısına kodlanmaktadır. Böylece o parametre değişkeni ağı girdiğinde, parametrenin değerine bağlı olarak yapay sinir ağı, ağda kodlanmış bulunan etkiyi oluşturarak sistemin aynı cevabı vermesini sağlar.

Talep tahmini şirketler için çok kritik bir önem taşımaktadır. Çünkü satın alma, üretim, pazarlama vb. ile ilgili kararlar tahmin sonuçlarına göre verilmektedir. Gerçeğe yakın bir talep tahmini şirketlerin başarısını artırır. Talebi etkileyen birçok faktör vardır; ancak belirsizliklerin fazla olduğu bir pazarda bu faktörler ve talep üzerindeki etkisinin tespiti oldukça güçtür.

Talep tahmini problemi literatürde birçok örneğinin de bulunduğu matematiksel modellerle çözülebilmektedir. Ancak talep tahmin probleminde talebi etkileyen tüm faktörlerin tam olarak belirlenememesi, hiç hesapta olmayan bir parametreden ötürü talebin olumlu ya da olumsuz yönde değişebilmesinden dolayı belirsizlik fazladır. Talebin hiç öngörülemeyen farklı faktörlerden etkilenmesi, belirsizliğin fazla olması, vb. sebeplerden dolayı talep tahmini problemine yapay sinir ağı optimum çözüm sunabilir. Doğru tasarlanan bir yapay sinir ağıyla talep tahmininde başarılı sonuçlar elde etmek mümkündür.

Bu tez çalışmasında makine imalatı yapan bir firmanın talep tahmini, çeşitli yapay sinir ağı tasarlanarak minimum hatayı veren sinir ağı tespit edilmeye çalışıldı. Makine modeli, ay ve ilgili makine modeli satışı girdi değişkeni; bir sene sonra aynı ayda satılan makine adedi çıktı değişkeni olarak tanımlanmıştır. Bir gizli katmana sahip sinir ağı yapısında; gizli nöron sayısı, aktivasyon fonksiyonu ve eğitim algoritması için duyarlılık analizleri yapıldı. 3 girişli, 1 çıkışlı, gizli katmanında 9 nöron içeren 3,9,1 yapısında, gizli katmanında 'logsig', çıkış katmanında 'purelin' aktivasyon fonksiyonuna sahip, 'trainlm' eğitim algoritmasıyla çalışan ağ, 0,003 performansla seçildi. Daha düşük performans elde etmek için 2 gizli katmanlı sinir ağı üzerinde denemelere devam edildi. 3,9,6,1 yapısında, 2 gizli katmanlı ve 1. gizli katmanında logsig, 2. gizli katmanında logsig, sonuç katmanında purelin aktivasyon fonksiyonuna sahip, 'trainlm' eğitim algoritmasıyla çalışan yapay sinir ağının talep tahmini problemine 0,001 düzeyinde hata ile çözüm bulduğu görülmüş ve kullanımına karar verilmiştir. Tez çalışmasında bir makine grubuna yönelik, eğitim için 40 aylık satış verisi, test için 8 aylık satış verisi kullanılmıştır. Bir makine grubu için eğitim ve testte kullanılmak üzere 48 aylık yani 4 senelik satış verisinden yararlanılmıştır. Talep tahmini için tasarlanacak ağda, ağın yıllık veya mevsimsel dalgalanmaları öğrenmesine olanak sağlayacak sayıda veri ile çalışılmalıdır.

Üretilen ürün çeşidine, satışı etkileyen faktörlere göre farklı yapay sinir ağları tasarlanabilir. Her yapay sinir ağı kendine özgüdür. Tez çalışmasında 0,001 performansla seçilen yapay sinir ağı başka bir ürün modelinin talep tahmini için olumlu sonuç vermeyebilir.

Yapay sinir ağı tasarımında her ne kadar nöron sayısı, gizli katman sayısı, aktivasyon fonksiyonu önemliyse sinir ağını eğitmede kullanılacak örnek sayısı da önemlidir. Yeterince örnek sunulmadan eğitime çalışılan bir sinir ağı ne kadar düzgün tasarlanmış olsa bile doğru bağlantıyı öğrenmediğinden yanlış sonuç verme ihtimali artar. Çok fazla örnek verilerek eğitime çalışılan bir sinir ağı da öğrenmemiş sadece ezberlemiş olabilir. Bundan dolayı ağı eğitmede kullanılacak örnek sayısına doğru karar verilmelidir.

Tez çalışmasında tasarlanan ağ çeşitli şekillerde geliştirilebilir. Tasarlanan yapay sinir ağında ortalama 300 farklı model makine, 10 makine grubu şeklinde sınıflandırılmış ve makine grubu bazında talep tahmini yapılmıştır. Yapay sinir ağına makine modelleri

ayrı ayrı tanıtılıp makine modeli bazında talep tahmini yapması sağlanabilir. Yıllık tahminin yanısıra program kullanıcılarına talep tahmini yapacağı bir makine modeli ve hangi ay için tahmin yapacağı bilgisi sorularak programın sadece model ve ay bazında talep tahmini yapması sağlanabilir. Kullanıcının girdiği bilgiler doğrultusunda yapay sinir ağı çalışarak ilgili modelin istenilen ayının satış tahminini ekrana getirebilir. Ayrıca yazılan programda satışı artan makine modellerinin ve satışı azalan makine modellerinin ayrı ayrı sınıflandırılması yapılabilir.

Uygulama yapılan firmada bazı makine modeli diğer model makine satışını da tetiklemektedir. Kendinden kazanı olmayan ütü masası veya paskalası alan müşteri bunları çalıştırmak için elinde yoksa merkezi bir buhar jeneratörü almalıdır. Ürün modellerinin birbirine olan etkisi yapay sinir ağına tanıtılarak model bazında gerçeğe daha yakın talep tahmini elde edilebilir.

Ağımızda girdilerimiz makine modeli, ay ve ilgili makine modeli satışı şeklindedir. Bu girdilere ilave olarak talebi yakından etkileyen ve gözlemlenebilen piyasanın ekonomik durumunu gösteren bir girdi değişkeni, gelecek yılın ilgili ayında fuar olup olmaması, devlet teşviğinin olup olmaması gibi yeni girdi değişkenleri ilave edilerek farklı bir ağ yapısı elde edilebilir.

Tasarlanan yapay sinir ağıımız çok katmanlı MLP ağıdır. Tahmin problemlerinde bu ağ genel kabul görmüştür. Ancak talep tahmini problemimize radyal tabanlı ağ (RBN)'ın nasıl bir çözüm sağladığı araştırılabilir. RBN ağları MLP ağları ile birçok bilimsel çalışmada karşılaştırılmıştır. [5]'te Çin'in 2020 enerji talepleri için RBN tabanlı oluşturulan ağın daha iyi sonuç verdiği belirtilmiştir. [9]'da ise tedarik zinciri çalışmasında depo ve fabrika talep tahmini için tasarlanan yapay sinir ağıında MLP ve RBN ağları karşılaştırılmış MLP'nin daha iyi sonuç verdiği tespit edilmiştir. Bu çalışmalar ışığında tezin uygulaması RBN yapay sinir ağı ile yapılarak sonuçlar karşılaştırılabilir.

KAYNAKLAR

- [1] Khashei, M. ve Bijari, M. , (2010). “An Artificial Neural Network (P,D,Q) Model for Time Series Forecasting “, ScienceDirect Journal of Expert System with Applications, 37: 479-489.
- [2] Wang, F.K. , Chang, K.K ve Tzeng, C.W., (2011). ”Using Adaptive Network-based Fuzzy Inference System to forecast Automobile Sales”, ScienceDirect Journal of Expert System with Applications, 38: 10587-10593.
- [3] Khah, M.S. , Moghaddam, M. P. ve Sheikh-El-Eslami, M.K., (2011). “Price Forecasting of Day-ahead Electricity Markets Using a Hybrid Forecast Method”, ScienceDirect Journal of Energy Conversion and Management, 52: 2165-2169.
- [4] Martins ve Werner, L. , (2012). ” Forecast Combination in Industrial Series: A Comparison Between Individual Forecasts and its Combinations with and without Correlated Errors”, ScienceDirect Journal of Expert System with Applications, 39: 11479-11486.
- [5] Yu, S., Wei, Y.M. ve Wang, K. , (2012). “China’s Primary Energy Demands in 2020: Predictions from an MPSO–RBF Estimation Model”, ScienceDirect Journal of Energy Conversion and Management, 61: 59-66.
- [6] Antonino, A. ve Messineo, A. , (2012). “Using Recurrent Artificial Neural Networks to Forecast Household Electricity Consumption”, ScienceDirect Journal of Energy Procedia, 14: 45-55.
- [7] Teixeira, J.P. ve Fernandes, P. O. , (2012). “Tourism Time Series Forecast - Different ANN Architectures with Time Index Input”, ScienceDirect Journal of Procedia Technology, 5: 445-454.
- [8] Akiner, M. E. ve Akkoyunlu, A. , (2012). “Modeling and forecasting river flow rate from the Melen Watershed, Turkey”, ScienceDirect Journal of Hydrology, 456-457:121-129.
- [9] Efendigil, T. ve Öñüt, S. , (2012). ” An Integrated Methodology Based on Fuzzy Inference Systems and Neural Approaches for Multi-Stage Supply-Chains”, ScienceDirect Journal of Computers & Industrial Engineering, 62:554-569.

- [10] Lau, H.C.W., Ho, G.T.S. ve Zhao, Y. , (2013). “A Demand Forecast Model Using a Combination of Surrogate Data Analysis and Optimal Neural Network Approach”, ScienceDirect Journal of Decision Support Systems, 54: 1404-1416.
- [11] Kulkarni, S., Sishaj, P., Simon ve Sundareswaran, K., (2013). “A spiking neural network (SNN) forecast engine for short-term electrical load forecasting”, ScienceDirect Applied Soft Computing, 13:3628-3635.
- [12] Rodger, J.A. , (2013). “A Fuzzy Nearest Neighbor Neural Network İstatistical Model for Predicting Demand for Natural Gas and Energy Cost Savings in Public Buildings”, ScienceDirect in Press of Expert System with Applications .
- [13] Jaipuria, S. ve Mahapatra, S.S. , (2013). “An Improved Demand Forecasting Method to Reduce Bullwhip Effect in Supply Chains”, ScienceDirect in Press of Expert System with Applications.
- [14] Kialashaki, A. ve Reisel, J.R. , (2013). “Modeling of the Energy Demand of the Residential Sector in the United States Using Regression Models and Artificial Neural Networks”, ScienceDirect Applied Energy, 108:271-280.
- [15] Belayneh, A. , Adamowski, J. , Khalil, B. ve Ozga-Zielinski, B. , (2013). “Long-term SPI Drought Forecasting in the Awash River Basin in Ethiopia Using Wavelet Neural Networks and Wavelet Support Vector Regression Models”, ScienceDirect In Press Journal of Hydrology.
- [16] Kheirkhah, A. , Azadeh, A. , Saberi, M., Azaron, A. ve Shakouri, H. , (2013). “Improved Estimation of Electricity Demand Function by Using of Artificial Neural Network, Principal Component Analysis and Data Envelopment Analysis”, ScienceDirect Journal of Computers & Industrial Engineering, 64:425-441.
- [17] Taşpınar, F. , Çelebi, N.ve Tutkun, N. , (2013). “Forecasting of Daily Natural Gas Consumption on Regional Basis in Turkey Using Various Computational Methods”, ScienceDirect Journal of Energy and Buildings, 56:23-31.
- [18] Claveria, O. ve Torra, S. , (2014). “Forecasting Tourism Demand to Catalonia: Neural Networks vs. Time Series Models”, ScienceDirect Journal of Economic Modelling, 36:220-228.
- [19] Öztemel, E. , (2003). Yapay Sinir Ağları, Papatya Yayıncılık, İstanbul.
- [20] Tebelkıs, J., (1995). Speech Recognition using Neural Networks, Thesis (PhD), School of Computer Science Carnegie Mellon University, Pennsylvania.
- [21] Zontul, M., (2004). Som Tipinde Yapay Sinir Ağlarını Kullanarak Türkiye ile Ticareti Olan Ülkelerin Kümelenmesi Üzerine Bir Çalışma, Doktora Tezi, Cumhuriyet Üniversitesi Fen Bilimleri Enstitüsü, Sivas.
- [22] Aydoğmuş, Z. ve Çöteli, R., (2005). Yapay Sinir Ağları Yardımıyla İzolatör Yüzeyinde Potansiyel Tahmini, F.Ü.Fen ve Müh.Bil. Dergisi, 17(2), 239-246.
- [24] Yurtoğlu, H., (2005).”YSA Metodolojisi ile Öngörü Modellemesi: Bazı Makroekonomik Değişkenler İçin Türkiye Örneği”, DPT Yayın No:2683,7-30.
- [25] Elmas, Ç., (2003). Yapay Sinir Ağları, Seçkin Yayınevi, Ankara.

- [26] Çuhadar, M., (2006). Turizm Sektöründe Talep Tahmini İçin Yapay Sinir Ağları Kullanımı ve Diğer Yöntemlerle Karşılaştırmalı Analizi (Antalya İlinin Dış Turizm Talebinde Uygulama), Süleyman Demirel Üniversitesi Sosyal Bilimler Enstitüsü İşletme Anabilim Dalı, Isparta.
- [27] Çetin, M., Uğur, A ve Bayzan Ş., (2006). İleri Beslemeli Yapay Sinir Ağlarında Backpropagation (Geriye Yayılım) Algoritmasının Sezgisel Yaklaşımı, Pamukkale Üniversitesi Bilgisayar Müh. Bölümü, Denizli.
- [28] Civalek Ö.ve Çatal, H. H., (2004). Geriye Yayılma Yapay Sinir Ağları Kullanılarak Elastik Kirişlerin Statik Ve Dinamik Analizi, DEÜ Mühendislik Fakültesi Fen ve Mühendislik Dergisi, Cilt. 6, Sayı 1, 1-16.
- [29] Efe, M.Ö. ve Kaynak, O., (2000). Yapay Sinir Ağları ve Uygulamaları, Boğaziçi Üniversitesi Yayınevi, İstanbul.
- [30] Anderson, D. ve Mc. Neill, G., (1992). Artificial neural networks, Technology, Kaman Sciences Corporation, New York.
- [31] Alavala, C. R. ,(2008). Fuzzy Logic and Neural Networks: Basic Concepts and Application. Delhi: New Age International.
- [32] Hebb, D.O., (1949). The organization of behavior: a neuropsychological theory, John Wiley&Sons, New York.
- [33] Çorumluğlu, Ö. ve Özbay, Y., (2005).” GPS Yüksekliklerinden Ortometrik Yüksekliklerin Elde Edilmesinde Yapay Sinir Ağı (YSA) Tekniğinin Kullanımı” , İTÜ Har. ve Kad. Müh.Odası, Mühendislik Ölçmeleri STB Komisyonu 2. Mühendislik Ölçmeleri Sempozyumu, 62-74.

YSA EĞİTİM SETİ

NO	GİRDİLER			ÇIKTI TAHMİN
	ÜRÜN GRUBU	AY	SATIŞ	
1	1	1	79	35
2	1	2	57	33
3	1	3	26	52
4	1	4	67	23
5	1	5	93	47
6	1	6	74	62
7	1	7	73	27
8	1	8	22	26
9	1	9	68	27
10	1	10	51	65
11	1	11	40	46
12	1	12	34	37
13	1	1	35	47
14	1	2	33	23
15	1	3	52	71
16	1	4	23	46
17	1	5	47	64
18	1	6	62	46
19	1	7	27	49
20	1	8	26	81
21	1	9	27	94
22	1	10	65	50
23	1	11	46	37
24	1	12	37	76
25	1	1	47	36
26	1	2	23	48
27	1	3	71	47
28	1	4	46	47
29	1	5	64	64
30	1	6	46	36
31	1	7	49	78
32	1	8	81	51
33	1	9	94	65
34	1	10	50	47
35	1	11	37	68
36	1	12	76	56

NO	GİRDİLER			ÇIKTI TAHMİN
	ÜRÜN GRUBU	AY	SATIŞ	
37	1	1	36	35
38	1	2	48	43
39	1	3	47	47
40	1	4	47	54
41	2	1	72	83
42	2	2	50	40
43	2	3	40	38
44	2	4	90	16
45	2	5	134	42
46	2	6	88	37
47	2	7	112	21
48	2	8	39	32
49	2	9	16	49
50	2	10	36	55
51	2	11	86	144
52	2	12	16	128
53	2	1	83	49
54	2	2	40	82
55	2	3	38	68
56	2	4	16	100
57	2	5	42	140
58	2	6	37	127
59	2	7	21	69
60	2	8	32	163
61	2	9	49	202
62	2	10	55	141
63	2	11	144	115
64	2	12	128	111
65	2	1	49	73
66	2	2	82	130
67	2	3	68	44
68	2	4	100	31
69	2	5	140	69
70	2	6	127	101
71	2	7	69	72
72	2	8	163	31
73	2	9	202	64
74	2	10	141	57
75	2	11	115	76
76	2	12	111	98
77	2	1	73	63
78	2	2	130	52
79	2	3	44	102
80	2	4	31	28
81	3	1	20	29
82	3	2	34	15
83	3	3	26	20
84	3	4	26	20
85	3	5	35	20
86	3	6	39	26

NO	GİRDİLER			ÇIKTI TAHMİN
	ÜRÜN GRUBU	AY	SATIŞ	
87	3	7	32	31
88	3	8	31	17
89	3	9	17	19
90	3	10	20	34
91	3	11	21	27
92	3	12	10	17
93	3	1	29	24
94	3	2	15	14
95	3	3	20	38
96	3	4	20	33
97	3	5	20	14
98	3	6	26	32
99	3	7	31	19
100	3	8	17	60
101	3	9	19	21
102	3	10	34	48
103	3	11	27	26
104	3	12	17	38
105	3	1	24	37
106	3	2	14	35
107	3	3	38	43
108	3	4	33	47
109	3	5	14	33
110	3	6	32	82
111	3	7	19	30
112	3	8	60	26
113	3	9	21	35
114	3	10	48	33
115	3	11	26	49
116	3	12	38	47
117	3	1	37	21
118	3	2	35	62
119	3	3	43	40
120	3	4	47	36
121	4	1	67	47
122	4	2	45	36
123	4	3	52	36
124	4	4	78	47
125	4	5	83	46
126	4	6	108	44
127	4	7	125	32
128	4	8	36	32
129	4	9	59	39
130	4	10	66	62
131	4	11	72	29
132	4	12	44	39
133	4	1	47	24
134	4	2	36	48
135	4	3	36	63
136	4	4	47	45

NO	GİRDİLER			ÇIKTI
	ÜRÜN GRUBU	AY	SATIŞ	TAHMİN
137	4	5	46	36
138	4	6	44	52
139	4	7	32	43
140	4	8	32	55
141	4	9	39	73
142	4	10	62	79
143	4	11	29	50
144	4	12	39	67
145	4	1	24	48
146	4	2	48	36
147	4	3	63	52
148	4	4	45	53
149	4	5	36	44
150	4	6	52	45
151	4	7	43	37
152	4	8	55	36
153	4	9	73	41
154	4	10	79	55
155	4	11	50	40
156	4	12	67	61
157	4	1	48	38
158	4	2	36	40
159	4	3	52	52
160	4	4	53	52
161	5	1	11	5
162	5	2	10	15
163	5	3	9	21
164	5	4	12	12
165	5	5	19	10
166	5	6	16	4
167	5	7	6	11
168	5	8	6	6
169	5	9	10	8
170	5	10	7	12
171	5	11	21	8
172	5	12	18	26
173	5	1	5	22
174	5	2	15	10
175	5	3	21	11
176	5	4	12	16
177	5	5	10	5
178	5	6	4	22
179	5	7	11	11
180	5	8	6	11
181	5	9	8	17
182	5	10	12	20
183	5	11	8	9
184	5	12	26	29
185	5	1	22	22
186	5	2	10	10

NO	GİRDİLER			ÇIKTI TAHMİN
	ÜRÜN GRUBU	AY	SATIŞ	
187	5	3	11	18
188	5	4	16	5
189	5	5	5	17
190	5	6	22	12
191	5	7	11	18
192	5	8	11	18
193	5	9	17	17
194	5	10	20	9
195	5	11	9	9
196	5	12	29	21
197	5	1	22	14
198	5	2	10	17
199	5	3	18	7
200	5	4	5	11
201	6	1	3	8
202	6	2	12	7
203	6	3	10	8
204	6	4	11	4
205	6	5	14	5
206	6	6	3	11
207	6	7	4	9
208	6	8	10	9
209	6	9	8	5
210	6	10	11	16
211	6	11	3	6
212	6	12	2	6
213	6	1	8	15
214	6	2	7	5
215	6	3	8	17
216	6	4	4	4
217	6	5	5	4
218	6	6	11	4
219	6	7	9	1
220	6	8	9	2
221	6	9	5	4
222	6	10	16	6
223	6	11	6	2
224	6	12	6	6
225	6	1	15	1
226	6	2	5	10
227	6	3	17	6
228	6	4	4	4
229	6	5	4	5
230	6	6	4	4
231	6	7	1	9
232	6	8	2	4
233	6	9	4	11
234	6	10	6	5
235	6	11	2	9
236	6	12	6	2

NO	GİRDİLER			ÇIKTI TAHMİN
	ÜRÜN GRUBU	AY	SATIŞ	
237	6	1	1	4
238	6	2	10	4
239	6	3	6	8
240	6	4	4	13
241	7	1	6	4
242	7	2	3	0
243	7	3	1	1
244	7	4	9	2
245	7	5	6	2
246	7	6	6	3
247	7	7	14	3
248	7	8	1	0
249	7	9	5	1
250	7	10	3	6
251	7	11	2	2
252	7	12	1	8
253	7	1	4	4
254	7	2	0	0
255	7	3	1	6
256	7	4	2	3
257	7	5	2	1
258	7	6	3	4
259	7	7	3	2
260	7	8	0	2
261	7	9	1	3
262	7	10	6	4
263	7	11	2	1
264	7	12	8	6
265	7	1	4	4
266	7	2	0	4
267	7	3	6	1
268	7	4	3	4
269	7	5	1	2
270	7	6	4	3
271	7	7	2	3
272	7	8	2	0
273	7	9	3	1
274	7	10	4	1
275	7	11	1	2
276	7	12	6	4
277	7	1	4	7
278	7	2	4	2
279	7	3	1	5
280	7	4	4	1
281	8	1	14	11
282	8	2	5	5
283	8	3	5	4
284	8	4	8	7
285	8	5	18	10
286	8	6	9	9

NO	GİRDİLER			ÇIKTI
	ÜRÜN GRUBU	AY	SATIŞ	TAHMİN
287	8	7	3	5
288	8	8	7	6
289	8	9	2	3
290	8	10	1	12
291	8	11	17	1
292	8	12	13	18
293	8	1	11	19
294	8	2	5	2
295	8	3	4	5
296	8	4	7	5
297	8	5	10	5
298	8	6	9	9
299	8	7	5	3
300	8	8	6	8
301	8	9	3	27
302	8	10	12	7
303	8	11	1	1
304	8	12	18	7
305	8	1	19	13
306	8	2	2	12
307	8	3	5	19
308	8	4	5	3
309	8	5	5	25
310	8	6	9	6
311	8	7	3	6
312	8	8	8	14
313	8	9	27	13
314	8	10	7	6
315	8	11	1	2
316	8	12	7	6
317	8	1	13	4
318	8	2	12	13
319	8	3	19	2
320	8	4	3	2
321	9	1	11	23
322	9	2	9	9
323	9	3	13	18
324	9	4	6	10
325	9	5	21	5
326	9	6	12	10
327	9	7	12	11
328	9	8	1	6
329	9	9	6	3
330	9	10	23	12
331	9	11	23	1
332	9	12	17	25
333	9	1	23	13
334	9	2	9	13
335	9	3	18	5
336	9	4	10	14

NO	GİRDİLER			ÇIKTI
	ÜRÜN GRUBU	AY	SATIŞ	TAHMİN
337	9	5	5	39
338	9	6	10	10
339	9	7	11	2
340	9	8	6	16
341	9	9	3	36
342	9	10	12	16
343	9	11	1	28
344	9	12	25	20
345	9	1	13	15
346	9	2	13	33
347	9	3	5	15
348	9	4	14	9
349	9	5	39	21
350	9	6	10	17
351	9	7	2	6
352	9	8	16	17
353	9	9	36	26
354	9	10	16	21
355	9	11	28	9
356	9	12	20	9
357	9	1	15	19
358	9	2	33	18
359	9	3	15	6
360	9	4	9	21
361	10	1	8	10
362	10	2	7	3
363	10	3	0	8
364	10	4	15	10
365	10	5	7	11
366	10	6	8	5
367	10	7	13	9
368	10	8	5	5
369	10	9	10	7
370	10	10	13	13
371	10	11	5	11
372	10	12	3	8
373	10	1	10	10
374	10	2	3	3
375	10	3	8	5
376	10	4	10	6
377	10	5	11	7
378	10	6	5	11
379	10	7	9	6
380	10	8	5	9
381	10	9	7	4
382	10	10	13	4
383	10	11	11	11
384	10	12	8	3
385	10	1	10	8
386	10	2	3	14

NO	GİRDİLER			ÇIKTI
	ÜRÜN GRUBU	AY	SATIŞ	TAHMİN
387	10	3	5	8
388	10	4	6	7
389	10	5	7	9
390	10	6	11	5
391	10	7	6	9
392	10	8	9	5
393	10	9	4	11
394	10	10	4	3
395	10	11	11	10
396	10	12	3	8
397	10	1	8	5
398	10	2	14	10
399	10	3	8	8
400	10	4	7	5

EK-B**YSA TEST SETİ**

NO	GİRDİLER			ÇIKTI
	ÜRÜN GRUBU	AY	SATIŞ	TAHMİN
1	1	5	64	62
2	1	6	36	53
3	1	7	78	46
4	1	8	51	57
5	1	9	65	46
6	1	10	47	85
7	1	11	68	42
8	1	12	56	68
9	2	5	69	114
10	2	6	101	188
11	2	7	72	50
12	2	8	31	115
13	2	9	64	184
14	2	10	57	180
15	2	11	76	53
16	2	12	98	75
17	3	5	33	34
18	3	6	82	31
19	3	7	30	39
20	3	8	26	50
21	3	9	35	51
22	3	10	33	43
23	3	11	49	44
24	3	12	47	51
25	4	5	44	49
26	4	6	45	50
27	4	7	37	44
28	4	8	36	41
29	4	9	41	56
30	4	10	55	69
31	4	11	40	32
32	4	12	61	54
33	5	5	17	12

NO	GİRDİLER			ÇIKTI
	ÜRÜN GRUBU	AY	SATIŞ	TAHMİN
34	5	6	12	6
35	5	7	18	12
36	5	8	18	8
37	5	9	17	16
38	5	10	9	19
39	5	11	9	9
40	5	12	21	8
41	6	5	5	10
42	6	6	4	12
43	6	7	9	18
44	6	8	4	6
45	6	9	11	13
46	6	10	5	5
47	6	11	9	13
48	6	12	2	3
49	7	5	2	1
50	7	6	3	1
51	7	7	3	6
52	7	8	0	5
53	7	9	1	2
54	7	10	1	4
55	7	11	2	9
56	7	12	4	2
57	8	5	25	25
58	8	6	6	6
59	8	7	6	5
60	8	8	14	11
61	8	9	13	3
62	8	10	6	14
63	8	11	2	4
64	8	12	6	4
65	9	5	21	20
66	9	6	17	5
67	9	7	6	10
68	9	8	17	14
69	9	9	26	7
70	9	10	21	10
71	9	11	9	19
72	9	12	9	16
73	10	5	9	7
74	10	6	5	18
75	10	7	9	6
76	10	8	5	7
77	10	9	11	8
78	10	10	3	15
79	10	11	10	9
80	10	12	8	8

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Emine SEVİNÇTEKİN
Doğum Tarihi ve Yeri : 10.05.1982 Gerger Adıyaman
Yabancı Dili : İngilizce
E-posta : emine_sevintekin@yahoo.com

ÖĞRENİM DURUMU

Derece	Alan	Okul/Üniversite	Mezuniyet Yılı
Y. Lisans	Sistem Mühendisliği	Yıldız Teknik Üniversitesi	2014
Lisans	Endüstri Mühendisliği	Kocaeli Üniversitesi	2005
Lise	Pertevniyal Lisesi		2001

İŞ TECRÜBESİ

Yıl	Firma/Kurum	Görevi
2014 -	Devlet Malzeme Ofisi (DMO)	Satın Alma Uzman Yrd.
2012 - 2013	PTT Aş.	Şube Memuru
2005 - 2011	Malkan Makine Ltd. Şti	Kalite Yönetim Müdürü

YAYINLARI

Makale

1. Alcan, P., Özdemir, Y., Başlıgil H., Sevinçtekin E., (2013). “Mobbing Davranışlarının Etkilerinin Araştırılması “ Sigma Mühendislik ve Fen Bilimleri Dergisi, 31: 222-233