

**YAPAY SİNİR AĞLARI İLE MERMER KESME
MAKİNESİNDE SPESİFİK ENERJİNİN TAHMİNİ
YÜKSEK LİSANS TEZİ**

Emre AKARSLAN

DANIŞMAN

Yrd. Doç. Dr. Murat CANER

ELEKTRİK EĞİTİMİ ANABİLİM DALI

Nisan 2009

**AFYON KOCATEPE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

YÜKSEK LİSANS TEZİ

**YAPAY SINIR AĞLARI İLE MERMER KESME MAKİNESİNDE SPESİFİK
ENERJİNİN TAHMİNİ**

Emre AKARSLAN

DANIŞMAN

Yrd. Doç. Dr. Murat CANER

ELEKTRİK EĞİTİMİ ANABİLİM DALI

Nisan 2009

ONAY SAYFASI

Yrd. Doç. Dr. Murat CANER danışmanlığında,
Emre AKARSLAN tarafından hazırlanan
YAPAY SİNİR AĞLARI İLE MERMER KESME MAKİNESİNDE SPESİFİK ENERJİNİN
TAHMİNİ

başlıklı bu çalışma, lisansüstü eğitim ve öğretim yönetmeliğinin ilgili maddeleri uyarınca
29/04/2009

tarhinde aşığıdaki jüri tarafından
Elektrik Eğitimi Anabilim Dalında
Yüksek Lisans tezi olarak oy birliğı ile kabul edilmiştir.

Ünvanı, Adı, SOYADI	İmza
Başkan Prof. Dr. Hasan ÇİMEN	
Üye Yrd. Doç. Dr. Murat CANER	
Üye Doç. Dr. Kubilay ASLANTAŞ	

Afyon Kocatepe Üniversitesi
Fen Bilimleri Enstitüsü Yönetim Kurulu'nun
...../...../..... tarih ve
..... sayılı kararıyla onaylanmıştır.

Doç. Dr. Zehra BOZKURT
Enstitü Müdürü

ÖZET

YÜKSEK LİSANS TEZİ YAPAY SİNİR AĞLARI İLE MERMER KESME MAKİNESİNDE SPESİFİK ENERJİNİN TAHMİNİ

Emre AKARSLAN

Afyon Kocatepe Üniversitesi

Fen Bilimleri Enstitüsü

Elektrik Eğitimi Anabilim Dalı

Danışman: Yrd. Doç. Dr. Murat CANER

Bu çalışmada iki farklı yapay sinir ağı (YSA) uygulaması gerçekleştirilmiştir. Her iki uygulamada da yapay sinir ağı uygulaması için Matlab programı kullanılmıştır. Birinci uygulamada YSA metodu kullanılarak, mermer kesme işleminde spesifik enerji (SE) tahmin edilmeye çalışılmıştır. Bu amaçla, Matlab’da önce kod tabanlı program yazılmış, sonra “nftool” araç kutusu kullanılmıştır. Kesme deneylerinde kullanılan, kesme hızı, testere dönüş hızı ve mermerin sertliği giriş verileri olarak alınmış, bu parametrelere göre spesifik enerji faktörü tahmin edilmiştir. Yapılan uygulama sonucunda, YSA ile yapılan tahminlerin, deneyler sonucu elde edilen verilere oldukça yakın olduğu görülmüştür.

İkinci uygulamada, yine spesifik enerji YSA kullanılarak tahmin edilmiştir. Ancak, bu uygulamada spesifik enerjiyi etkileyen dört parametre dikkate alınmıştır. Bu amaçla, kesme işleminde kullanılan motorun gücü de spesifik enerji değerinin tahmini için hesaba katılmıştır. Ayrıca YSA’nın eğitiminde verilerin birbirine yakınlığının eğitimi kolaylaştıracağı düşünüldüğünden, bazı parametrelerde normalizasyon işlemi gerçekleştirilmiştir. Yapılan uygulama sonucunda, YSA ile yapılan tahminlerin, deneyler sonucu elde edilen verilere oldukça yakın olduğu görülmüştür.

2009, 64 sayfa

Anahtar Kelimeler: Yapay sinir ağları, spesifik enerji, mermer kesme, nftool, Matlab

ABSTRACT

M. Sc. Thesis

THE ESTIMATION OF SPECIFIC ENERGY WITH ARTIFICIAL NEURAL NETWORK IN MARBLE CUTTING MACHINE

Emre AKARSLAN

Afyon Kocatepe University

Graduate School of Natural and Applied Sciences

Department of Electric Education

Supervisor: Yrd. Doç. Dr. Murat CANER

In this study, two different artificial neural network (ANN) applications were realized. Matlab programme was used for Artificial neural network application in both implementations. In first application, the specific energy factor (SE) was tried to estimate in marble cutting operation, by using ANN method. For this purpose, code based program was written first, then “nftool” toolbox was used in Matlab. Cutting speed, speed of circular saw rotation and marble’s hardness, which were used in cutting experiments, were taken as inputs, and according to these parameters, the specific energy factor was estimated. As conclusion, it was reported that estimated data were quite close to experimental result.

In second application, spesific energy was estimated by using ANN, again. But, in this application, four different parameter were taken into account which effect the spesific energy. For this purpose, motor power using for cutting process was allowed for to estimation of spesific energy. In addition, normalization transaction was applied in some parameter, because of that could expedite to ANN’s training. As conclusion, it was reported that estimated data were quite close to experimental result.

2009, 64 pages

Key Words: Artificial neural network, specific energy, marble cutting, nftool, Matlab

TEŐEKKÜR

Çalıřmalarımda bana yardımcı olan ve yönlendiren danıřman hocam Yrd. Doç. Dr. Murat CANER'e, bölüm başkanımız Prof. Dr. Hasan Çimen'e, gerektiğinde yardımlarını esirgemeyen bölüm hocalarıma, çalışma arkadaşlarıma, her zaman beni destekleyen, yanımda olan ve beni bugünlere getiren başta annem ve rahmetli babam olmak üzere, tüm aileme sonsuz teşekkürlerimi sunarım.

Emre AKARSLAN
AFYONKARAHİSAR, Nisan 2009

İÇİNDEKİLER

ÖZET	iv
ABSTRACT	v
TEŞEKKÜR	vi
SİMGELER ve KISALTMALAR DİZİNİ	ix
ŞEKİLLER DİZİNİ	x
RESİMLER DİZİNİ	xi
ÇİZELGELER DİZİNİ	xii
1. GİRİŞ	1
2. S/T MERMER KESME MAKİNESİ	3
2.1. Mermer Kesme Makinesi Yapısı	4
2.2. Kesme Parametreleri	9
2.3. Kesme Kinematığı	12
3. YAPAY SİNİR AĞLARI (YSA)	16
3.1. Yapay Sinir Ağlarının Tarihçesi	16
3.2. Yapay Sinir Ağlarında Olumlu – Olumsuz yönler	18
3.3. Yapay Sinir Ağlarını Oluşturan Temel Elemanlar	19
3.3.1. Girdiler	19
3.3.2. Ağırlıklar	19
3.3.3. Toplama Fonksiyonu	19
3.3.4. Aktivasyon Fonksiyonu	20
3.3.5. Çıktılar	21
3.4. Yapay Sinir Ağlarında Katmanlar	21
3.5. Yapay Sinir Ağlarında Öğrenme Prensipleri	22
3.5.1. Denetimlerine Göre Öğrenme Şekilleri	22
3.5.1.1. Denetimli (Öğretmenli) Öğrenme	23
3.5.1.2. Destekleyici (Reinforcement) Öğrenme	23
3.5.1.3. Denetimsiz (Öğretmensiz) Öğrenme	24
3.5.1.4. Karma Stratejiler	24
3.6. Başlıca Öğrenme Kuralları	24

3.6.1. Hebb Kuralı	24
3.6.2. Hopfield Kuralı	24
3.6.3. Kohonen Kuralı	25
3.6.4. Delta Kuralı	25
3.7. Çok Katmalı Algılayıcılar	25
3.8. Hızlı Algoritmalar	26
3.8.1. Levenberg-Marquart Algoritması	26
3.9. Matlab'ta YSA Modellenmesi	29
4. UYGULAMA	35
4.1. Mermer Kesme İşleminde Spesifik Enerjinin (SE) YSA ile Tahmini	35
4.1.1. Ağın eğitimi için gerekli verilerin hazırlanması	35
4.1.2. Yapay sinir ağının eğitilmesi	38
4.1.3. Yapay sinir ağının test edilmesi	40
4.1.4. Spesifik Enerjinin 'nftool' ile tahmini	43
4.2. Spesifik Enerjinin dört farklı parametreye bağlı tahmini	46
4.2.1. Ağın eğitimi için gerekli verilerin hazırlanması	47
4.2.2. Yapay sinir ağının eğitilmesi	49
4.2.3. Yapay sinir ağının test edilmesi	51
4.2.4. Spesifik Enerjinin 'nftool' ile tahmini	53
5. SONUÇ VE DEĞERLERNDİRME	58
6. KAYNAKLAR	60
6.1. İnternet Kaynakları	63
7. ÖZGEÇMİŞ	64

SİMGELER ve KISALTMALAR DİZİNİ

1. Simgeler

F_X	Disk kesme yönünde oluşan yatay kuvvet	[N]
F_Y	Disk kesme yönünde oluşan dikey kuvvet	[N]
F_Z	Disk kesme yönüne dik oluşan aksenal kuvvet	[N]
F_N	Normal (Radyal) kuvvet	[N]
F_T	Teğetsel (Tanjantiyel) kuvvet	[N]
F_K	Kesme kuvveti	[N]
H_K	Kesme derinliği	[mm]
P_C	Motor gücü (kuvvetlerden hesaplanan)	[Nm/sn]
q	Kesme hızları oranı	
Q_W	Birim zamanda kesilen hacim	[mm ³ /sn]
SE_m	Özgül kesme enerjisi (Mekaniksel)	[J/mm ³]
SE_e	Özgül kesme enerjisi (Elektriksel)	[Wh/mm ³]
V_k	İlerleme hızı	[m/dk]
$V_ç$	Testere çevresel hızı	[m/sn]
V_d	Testere dönüş hızı	[d/dk]
α	Normal kuvvetin etki (aşındırma) açısı	[derece]
μ	Kesme kuvvetleri oranı	
φ_k	Disk in kayaçla temas (kavrama) açısı	[derece]
Φ_D	Disk (testere) çapı	[mm]

ŞEKİLLER DİZİNİ

	Sayfa No
Şekil 2.1. Makinenin perspektif gösterimi (Büyüksağış 1998)	5
Şekil 2.2. Veri toplama sistemi blok şeması (Çınar 2007)	8
Şekil 2.3. Aynı yönlü kesme işlemi kinematığı (Büyüksağış 1998)	12
Şekil 2.4. Zıt yönlü kesme işlemi kinematığı (Büyüksağış 1998)	13
Şekil 3.1. Kullanılan bazı önemli sigmoid fonksiyonları	21
Şekil 3.2. Yapay sinir ağlarında katmanlar	22
Şekil 4.1. Hatanın iterasyona bağlı değişim grafiği	40
Şekil 4.2. Deneyler sonucu elde edilen değer ile tahmini değer arasındaki farkı gösteren Grafik	41
Şekil 4.3. Oluşturulan ağa ait prensip şema	43
Şekil 4.4. 14 iterasyona göre toplam MSE hata	44
Şekil 4.5. Çıkış ve hedefin regresyon analizi	44
Şekil 4.6. Gerçek ve tahmini değerlerin karşılaştırılması	45
Şekil 4.7. Her bir hata için çubuk grafiği	45
Şekil 4.8. Hatanın histogramı	46
Şekil 4.9. Hatanın iterasyona bağlı değişim grafiği	51
Şekil 4.10. Deneyler sonucu elde edilen değer ile tahmini değer arasındaki farkı gösteren grafik	53
Şekil 4.11. Oluşturulan ağa ait prensip şema	54
Şekil 4.12. 166 iterasyona göre toplam MSE hata	55
Şekil 4.13. Gerçek ve tahmini değerlerin karşılaştırılması	55
Şekil 4.14. Çıkış ve hedefin regresyon analizi	56
Şekil 4.15. Her bir hata için çubuk grafiği	56
Şekil 4.16. Hatanın histogramı	57

RESİMLER DİZİNİ

	Sayfa No
Resim 2.1. İki kolonlu S/T mermer kesme makinesi (İnt. Kyn. 1)	3
Resim 2.2. Dört kolonlu S/T mermer kesme makinesi (İnt. Kyn. 2)	4
Resim 2.3. ST makinesinin görünüşü	5

ÇİZELGELER DİZİNİ

	Sayfa No
Çizelge 2.1. Kesme deneyleri için varyasyon sırası	11
Çizelge 4.1. Varyasyon deneyleri işlem sırası	36
Çizelge 4.2. YSA'nın eğitimi için kullanılan veriler	37
Çizelge 4.3. Deney sonucu elde edilen veriler ile tahmin sonuçlarını karşılaştıran tablo	42
Çizelge 4.4. YSA'nın eğitimi için kullanılan veriler	48
Çizelge 4.5. Deney sonucu elde edilen veriler ile tahmin sonuçlarının karşılaştırılması	52

1. GİRİŞ

Mermer bloklarının, son ürün elde edilmesi için işlendiği yerler olan mermer işleme tesislerinde amaç, mümkün olan en düşük maliyetle piyasa tarafından ihtiyaç duyulan miktarda ürünün üretilmesinin sağlanmasıdır. Bu amaç doğrultusunda mermer işleme tesislerinde verimli bir biçimde işlemlerin sürdürülmesi gerekmektedir. Literatürde mermer kesme parametreleriyle enerji tüketimi ilişkisi üzerine farklı deneysel çalışmalar yapılmıştır.

Polini ve Turchetta (2004) elmas mil ile mermer kesim işleminde kesme parametreleri ve kesici optimizasyonu üzerine bir çalışma gerçekleştirmişlerdir. Çalışmalarında mermere etkiyen kuvvetlerden hareketle hesaplanabilen spesifik enerji modelini kullanmışlardır. Chen (2000) orta setlikteki çekikten yapılmış CNB kesici kullanarak yaptığı kesme deneylerinde oluşan kesme kuvvetleri ve kesilen mermer yüzeylerindeki deformatsiyonu incelemiştir.

Büyüksağış ve Gökten (2005) kesim sırasında mermere etkiyen kuvvetlerden hareketle harcanan enerjiyi hesaplamışlardır. Çalışmalarında tamamen bilgisayar kontrollü bir deney düzeneği kullanmışlardır. Farklı tipte mermer kullanılarak gerçekleştirilen çalışmada iki farklı parametreye göre kesim enerjisi incelenmiştir.

Xu vd. (2001) gerçekleştirdikleri deneysel çalışmalarında dairesel testereli bir mermer makinesinde gri granitin kesimi için farklı çalışma koşullarında kesme kuvvetleri ve enerjisinin değişimini incelemiştir. Kuvvet ve kesim gücü ölçümlerini bilgisayar ortamına aktarmışlar ve verileri işlemek için Matlab yazılımı kullanmışlardır.

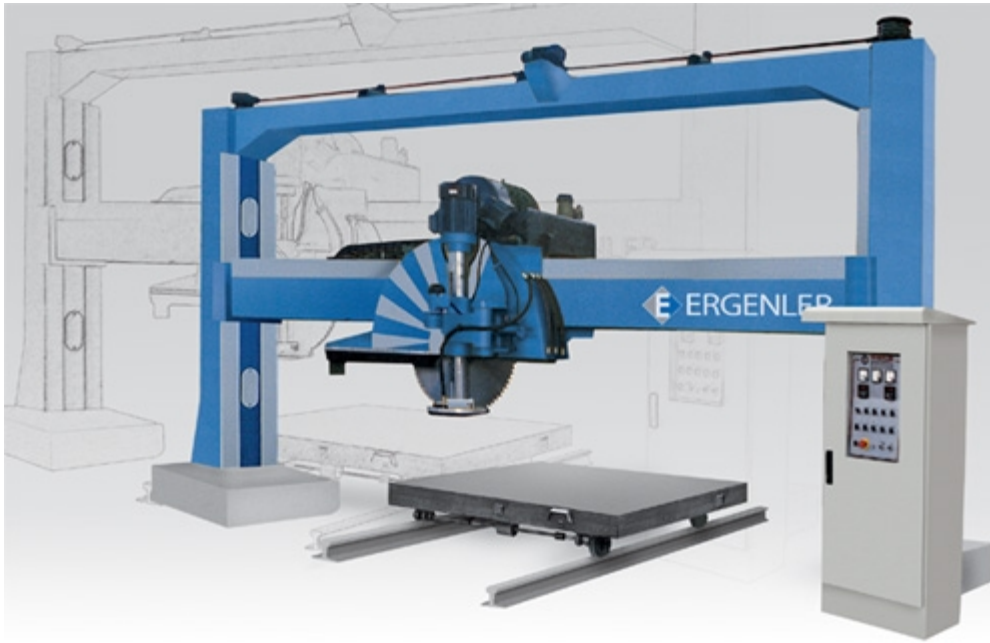
Çınar(2007) mermer kesme işleminde kullanılan elektrik enerji tüketimi ve kesme parametrelerin bu tüketime etkisi incelemiştir. Kesme deneyleri için bilgisayar tabanlı dairesel testereli blok kesme makinesi kullanmış ve Bilecik Bej mermeri ile Denizli traverteni için kesme hızı ve testere dönüş hızının enerji tüketimine etkisini gözlemiştir.

Sonuç olarak uygun kesme parametreleri belirlendiđi takdirde enerji tüketiminin optimize edilebileceđini görmüştür.

Büyüksađıř (1998), blok kesme makinelerinin alıřma prensiplerine uygun olarak tasarımılandırılan laboratuvar ölekli bir mermer kesme deney seti kullanarak hakiki mermer grubuna ait 7 deđiřik mermer örneđinin kesilebilirlik özelliklerini incelemiřtir. Yaptıđı deneyler sonucu elde ettiđi parametreler ile spesifik enerjii hesaplayan Büyüksađıř, daha sonra ok deđiřkenli regresyon analizi yöntemiyle bu spesifik enerjii tahmin etmeye alıřmıřtır.

2. S/T MERMER KESME MAKİNESİ

Makineye S/T adını Stripper ve Trimmer kelimeleri vermektedir. Stripper kelimesi “dilimleyerek ayırmak”, Trimmer kelimesi ise “keserek ayırmak” anlamına gelmektedir. Makine üzerinde bulunan iki testere bu fonksiyonları yerine getirmektedir. Dilimleme düşey testere ile yapılırken kesme yatay testere tarafından gerçekleştirilmektedir. S/T mermer makineleri yapı itibarıyla iki veya dört kolonlu olarak imal edilmektedir. Resim 2.1’de görülen iki kolonlu makinede testereleri taşıyan platform iki boyutta hareket etmektedir. Platformun ileri-geri hareketi ile kesme ilerlemesi ve yukarı-aşağı hareketiyle kesme derinliği değişmektedir. Dilim kalınlıkları ise kesilecek mermeri taşıyan vagonun öne-arkaya hareketi ile değiştirilebilmektedir. İki ayaklı(lolonlu) S/T lerde, hem vagon hareketli hem de köprü üzerindeki testere hareketlidir. (Bağcıvandemir 1997)



Resim 2.1. İki kolonlu S/T mermer kesme makinesi (İnt. Kyn.1)

Resim 2.2’de görülen dört kolonlu makinede ise testereleri taşıyan platform üç boyutta hareket edebilmekte ve dilim kalınlığını değiştirmek için testere öne-arkaya hareket etmektedir. Dolayısıyla iki kolonlu makinelerin tersine dilim kalınlığının vagon hareketi ile

ayarları söz konusu değildir. Vagon sadece kesilecek mermeri makinenin ortasına taşımak için hareket etmektedir.



Resim 2.2. Dört kolonlu S/T mermer kesme makinesi (İnt.Kyn.2)

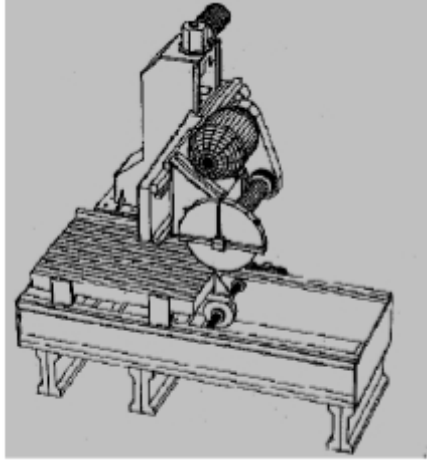
Aşağıda, bu çalışmada kullanılan verileri elde etmek için, Said Mahmut Çınar'ın tez çalışmasında kullandığı, mermer kesme makinesinin yapısı anlatılmıştır.

2.1. Mermer Kesme Makinesi Yapısı

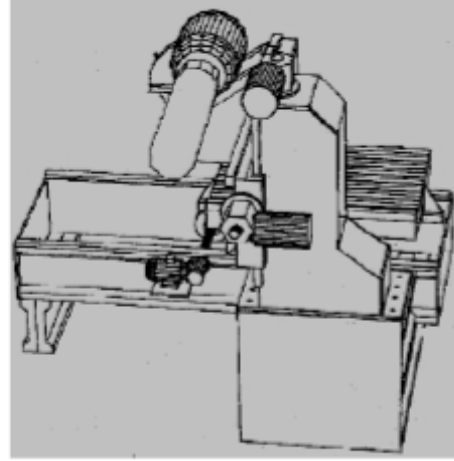
Afyon Kocatepe Üniversitesi Laboratuvarlarında bulunan mermer kesme makinesi, mermer endüstrisinde “Yan Kesme” olarak bilinen mermer makinesine benzemektedir. Endüstride kullanılan S/T mermer makinelerinde düşey ve yatay olmak üzere iki kesici testere bulunmasına rağmen, Resim 2.3’de görüldüğü gibi makine üzerinde sadece düşey testere bulunmaktadır. Yatay testerenin kesme işlemine etkisi ihmal edilebildiğinden makinede düşey testerenin bulunması yeterli görülmüştür.



Resim 2.3. S/T makinenin görünüşü



Ön görünüş



Arka görünüş

Şekil 2.1. Makinenin perspektif gösterimi (Büyüksağış 1998)

Şekil 2.1’de verilen perspektiflerde görülebileceği üzere testerenin yatay eksendeki öne arkaya hareketi ile dilim kalınlığı, düşey eksendeki yukarı-aşağı hareketiyle kesme derinliği ayarlanmaktadır. Testere yatay ve düşey eksende hareket edebilen bir platform üzerine monte edilmiş ve kızaklarla yataklanmıştır. Kızakların hareketi sonsuz dişlilere bağlı 0.75kW motorlu redüktör ile sağlanmaktadır. Redüktör milinin bir ucu sonsuz dişliye bağlı iken diğer ucu altı kanatlı bir pervaneye bağlanmıştır. Pervane üzerindeki kanatlar

sayesinde hareket miktarı milimetre hassasiyetinde belirlenebilmektedir. Vagonun ileri-geri hareketi, çark zincir sistemi ile gerçekleştirilmekte ve sistemin hareketi 0.75kW motorlu redüktör ile sağlanmaktadır. Çark zincir sistemi sayesinde vagonun ileri-geri hareketinin kesintisiz ve darbesiz olması sağlanmaktadır. Testerenin dönme hareketi kasnak kayış sistemi ile gerçekleştirilir. Testere 400mm çapa sahiptir ve 7.5kW gücünde motorla kontrol edilmektedir. Motorun mili kasnak-kayış sistemindeki, kasnaklardan küçük olanına bağlıdır. Diğer kasnak platforma yataklanmış testere miline bağlıdır. Motor ve testere arasındaki güç aktarımı kasnaklar arasında çalışan üç kayış ile sağlanmaktadır. Ayrıca testere miline bağlı olan kasnakla akuple çalışan tako-generatör ile testere dönüş hızı ölçülmektedir.

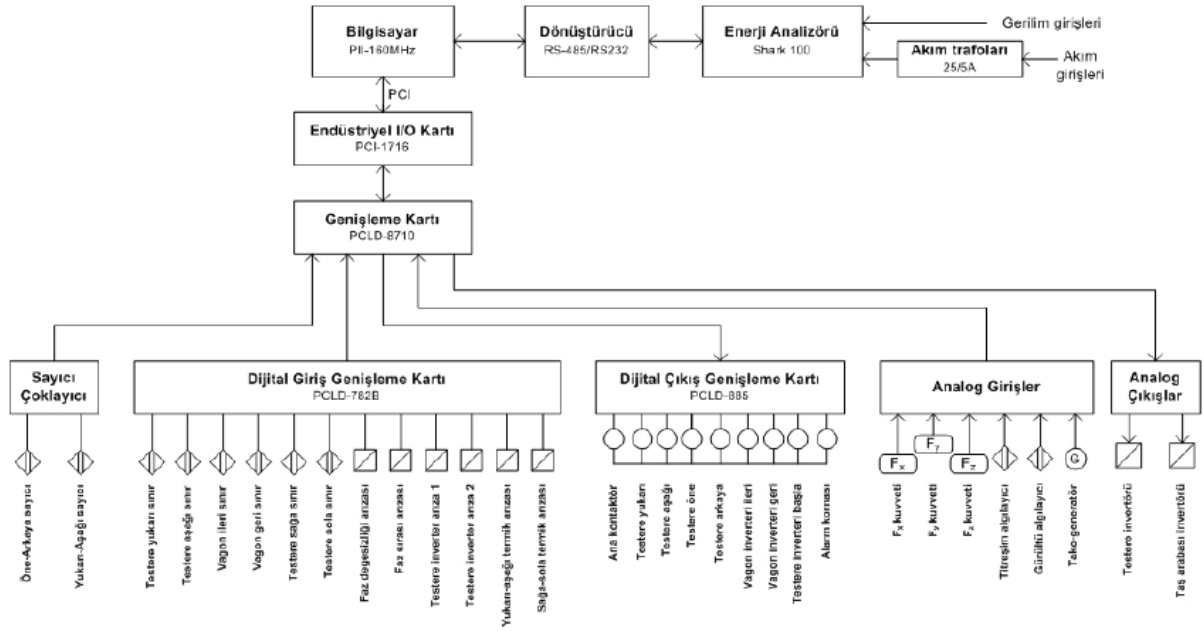
Makine üzerindeki motorların şebeke bağlantıları, elektrik kumanda sistemi ile gerçekleştirilir. Söz konusu motorlar; testere motoru (7.5kW), testere platformu motorları (0.75kW), vagon motoru (0.75kW) ve soğutma suyu besleme motorlarından (0.5kW ve 0.75kW) oluşur. Kullanılan altı motordan üçü invertör ile kontrol edilirken diğer üçü doğrudan şebeke gerilimi ile çalıştırılır. Öncelikle kullanılan motorların güvenli çalışmalarını temin etmek için özel koruma röleleri kullanılmıştır. Faz dengesizliği tespiti faz koruma rölesi ile fazların sırasının tespiti ise faz sırası rölesi ile gerçekleştirilmektedir.

Testere motoru 7.5kW'lık VERİKON marka invertörle kontrol edilmektedir. İntertör ile motor gerilimi frekansı 0-90Hz aralığında değiştirilebilmekte dolayısıyla motor devir sayısı 0-5400d/dk arasında ayarlanabilmektedir. İntertörün hız ayarı için analog girişi kullanılırken start stop ve arıza sinyalleri için dijital giriş-çıkışları kullanılmaktadır. Testere motoru elektrik tüketimi SHARK 100 marka harmonik analizi yapabilen bir enerji analizörü ile ölçülmektedir. Analizörün gerilim bağlantıları üç faz ve bir nötr olmak üzere doğrudan yapılmaktadır. Ancak invertör akımının, analizör ölçme sınırını aşması sebebiyle akım bağlantıları doğrudan yapılamamaktadır. Bu nedenle akım ölçme işleminde 25/5A, 10VA ve 0.5 sınıfına sahip üç adet akım trafosu kullanılmıştır. Vagon motoru 0.75kW'lık SIEMENS marka invertörle kontrol edilmektedir. Vagonun ilerleme hızı 0-4m/dk aralıkta

değiştirilebilmektedir. İvertörün hız ayarı için analog girişi kullanılırken ileri-geri ve arıza sinyalleri için dijital giriş-çıkışları kullanılmaktadır.

Testere platformunun hareketi için kullanılan 0.75kW gücündeki asenkron motorlar, ikişer adet kontaktör ile gerçekleştirilen devir yönü değiştirme devreleri ile kontrol edilmektedir. Kontaktörler üzerinde üç güç kontağı ve bir normalde kapalı kumanda kontağı bulunmaktadır. Normalde kapalı kumanda kontakları iki kontaktörün aynı anda çekmesini önlemek için kullanılmaktadır. Ayrıca platform hareket motorlarının herhangi bir sebeple mekanik olarak sıkışması veya faz kesilmesi (motorun iki faza kalması) gibi arızalardan korunabilmesi için, motor besleme hattına 1.6–2.5A çalışma aralığına sahip, aşırı akım rölesi (termik röle) kullanılmıştır. Termik rölelerin birer adet normalde açık ve normalde kapalı olmak üzere iki adet kumanda kontağı bulunmaktadır. Normalde kapalı kontak, kontaktör bobinlerinin besleme hattına seri bağlanmıştır ve arıza durumunda açılarak çalışan kontaktörün enerjisini kesip motorun durmasını sağlamaktadır. Normalde açık kontak ise arıza durumunda kapanarak bilgisayara aşırı akım arıza sinyali göndermektedir. Soğutma suyu besleme birimi, makinenin elektrik kumanda sistemindeki son birimi oluşturmaktadır. Mermer kesme işlemi esnasında testerenin ısınmasını önlemek için soğutma suyu kullanılmaktadır. Soğutma suyu besleme birimi testereye verilen suyun miktarını (debisini) ayarlayabilecek şekilde tasarlanmıştır. Ayrıca suyun kesintiye uğramasını önlemek için bir su deposu kullanılmaktadır. (Çınar 2007)

Makinenin bilgisayarla kontrolü için endüstriyel I/O kartı, izoleli giriş-çıkış genişleme kartları, çeşitli algılayıcılar, enerji analizörü, güç ve kontrol röleleri gibi çeşitli donanımlardan oluşan veri toplama sistemi oluşturulmuştur. Aşağıda Şekil 2.2'de makinenin veri toplama sistemini özetleyen bir blok şema görülmektedir.



Şekil 2.2. Veri toplama sistemi blok şeması (Çınar 2007)

Veri toplama sistemi, ADVANTECH firmasının PCI-1716 endüstriyel I/O kartı etrafında kurulmuştur. Endüstriyel I/O kartı bilgisayarın PCI yuvasına monte edilmiş olup çevre birimlerle olan bağlantıları dijital I/O ve analog I/O genişleme kartları ile gerçekleştirilmektedir. Genişleme kartları, I/O kartının çevre donanımlarla bağlantılarını sağlamanın yanı sıra bilgisayara zarar verebilecek olan elektriksel tehditlere karşı yalıtımı ve korumayı da sağlamaktadır. Veri toplama sisteminin diğer bir parçası bilgisayarın RS232 terminalinden haberleşen enerji analizörüdür. Ancak enerji analizörünün haberleşme için RS485 haberleşme standardını kullanması analizör ile bilgisayar arasında RS485-RS232 dönüştürücü kullanılması zorunlu kılmıştır.

I/O kartı ve enerji analizörü tarafından toplanan veriler ara yüz yazılımı ile işlenmekte ve makinenin bilgisayar tabanlı kontrolü gerçekleştirilmektedir. Ara yüz yazılımı ile makinenin tüm hareketleri bilgisayar ile kontrol edilebildiği gibi bütün veriler bilgisayar monitöründen gözlenebilmektedir. Ayrıca deney sonuçları istenilen formatta kaydedilmekte ve analiz programları ile kolayca işlenebilmektedir.

2.2. Kesme Parametreleri

Kesme işleminde birbirleri ile etkileşimde bulunan birçok parametre bulunmaktadır.

Bunlar:

- kesici testerenin özellikleri
- kesilecek kayacın fiziko-mekanik yapısı ve özellikleri
- kesme hızı
- testere dönüş hızı
- kesme derinliği
- kesme işleminde kullanılan su miktarı

Bu parametrelerin değişimi ile mermer kesiminde tüketilen enerji de değişiklik göstermektedir. Ancak bunların tamamının etkileşimini incelemek yüzlerce kesme deneyinin gerçekleştirilmesini beraberinde getirmektedir. Bu ise önemli bir bütçeye tekabül etmektedir.

Birim hacim başına harcanan enerji ifadesi olan spesifik enerji faktörünün (SE) hesaplanabilmesi için gerekli parametreler, kesme parametrelerinin varyasyon deneyleri ile elde edilmektedir. (Çınar 2007) da yaptığı çalışmada bu amaçla, Kesme derinliği (H_K), İlerleme Hızı (V_K) ve Testere Dönüş Hızı (V_d) parametrelerinin aşağıda verilen aralıklarında, Bilecik Bej mermeri ve Denizli traverteni numuneleri üzerinde toplam 40 adet deney gerçekleştirilmiştir.

Kesme Derinliği (H_K) : 50mm

Kesme Hızı (V_K) : 0.3, 0.4, 0.5, 0.6 ve 0.7m/dk

Testere Dönüş Hızı (V_d) : 1432, 1910, 2387 ve 2864d/dk

Ayrıca (Dursun 2008) yaptığı çalışmada, Uşak Yeşil mermeri üzerinde, 5,5 kW'lık asenkron motor kullanılarak, Kesme Derinliği (H_K), Kesme Hızı (V_K) ve Testere Dönüş

Hızı (V_d) parametrelerinin aşağıda verilen aralıklarında, toplam 9 adet deney gerçekleştirilmiştir.

Kesme Derinliği (H_K) : 50mm

Kesme Hızı (V_K) : 0.5, 0.6 ve 0.7m/dk

Testere Dönüş Hızı (V_d) : 2387, 2626 ve 2864d/dk

Her iki çalışmada dikkate alındığında, makine parametrelerinden Kesme Derinliği (H_K) 50mm değerinde sabit tutulmaktadır. Kesme hızı (V_K) 0.3-0.7m/dk aralığında 5 değer almaktadır. Testere Dönüş Hızı (V_d) ise 2387-2864d/dk aralığında 5 değer almaktadır.

Mermencilik üzerine yapılan akademik çalışmalarda Testere Dönüş Hızı yerine testerenin Çevresel Hızı [V_c] kullanılmakta ve birimi [m/sn] olarak belirlenmektedir. Ancak elektrik mühendisliği açısından testerenin çevresel hızı değil dakikada yaptığı devir sayısı önem arz etmektedir. Bu nedenle kesme deneylerinin literatür ile uyumlu olması açısından testere Çevresel hız değerinin 30, 40, 50,55 ve 60m/sn değerlerine karşılık gelen 1432, 1910, 2387,2626 ve 2864d/dk değerlerindeki Testere dönüş hızları kullanılmıştır.

Kesme derinliği, İlerleme hızı ve Testere dönüş hızı için seçilen parametrelere karşılık yapılacak deney sayısı 20 adet olmaktadır. Ancak iki tip kayaç için deneyler tekrar edildiğinde toplam deney sayısı 40 adede çıkmaktadır. Bu değerler 5.5 kW'lık motor ile yapılan 9 deney de eklenince, deney sayısı 49'a çıkmaktadır. Çizelge 2.1 kesme deneylerinin varyasyon sırasını göstermektedir.

Çizelge 2.1'de sıralaması verilen varyasyon deneyleri Denizli traverteni, Bilecik Bej ve Uşak Yeşil mermerleri için tekrarlanmıştır. Varyasyon deneylerinde söz konusu kayaçların kesiminde iki farklı testere kullanılmıştır. Seçilen kayaç numunelerinin farklı fizikokimyasal özellikte olması dolayısıyla aynı tipte testere ile kesilmeye uygun olmaması her kayaç için özel testere kullanımını zorunlu kılmıştır.

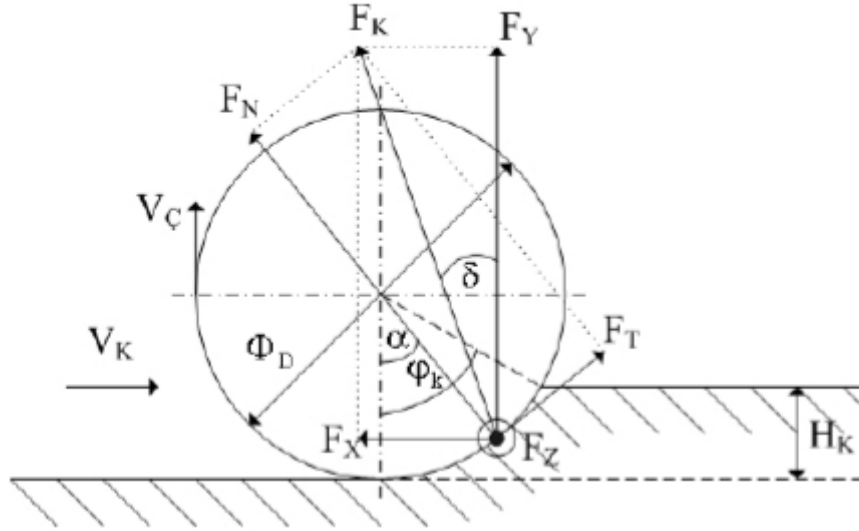
Çizelge 2.1. Kesme deneyleri için varyasyon sırası

Deney No	Motor Gücü(kW)	Kesme Derinliği [mm]	Testere Dönüş Hızı [d/dk]	İlerleme Hızı [m/dk]
1	7,5	50	1432	0,3
2	7,5	50	1432	0,4
3	7,5	50	1432	0,5
4	7,5	50	1432	0,6
5	7,5	50	1432	0,7
6	7,5	50	1910	0,3
7	7,5	50	1910	0,4
8	7,5	50	1910	0,5
9	7,5	50	1910	0,6
10	7,5	50	1910	0,7
11	7,5	50	2387	0,3
12	7,5	50	2387	0,4
13	7,5	50	2387	0,5
14	7,5	50	2387	0,6
15	7,5	50	2387	0,7
16	7,5	50	2864	0,3
17	7,5	50	2864	0,4
18	7,5	50	2864	0,5
19	7,5	50	2864	0,6
20	7,5	50	2864	0,7
21	7,5	50	1432	0,3
22	7,5	50	1432	0,4
23	7,5	50	1432	0,5
24	7,5	50	1432	0,6
25	7,5	50	1432	0,7
26	7,5	50	1910	0,3
27	7,5	50	1910	0,4
28	7,5	50	1910	0,5
29	7,5	50	1910	0,6
30	7,5	50	1910	0,7
31	7,5	50	2387	0,3
32	7,5	50	2387	0,4
33	7,5	50	2387	0,5
34	7,5	50	2387	0,6
35	7,5	50	2387	0,7
36	7,5	50	2864	0,3
37	7,5	50	2864	0,4
38	7,5	50	2864	0,5
39	7,5	50	2864	0,6
40	7,5	50	2864	0,7
41	5,5	50	2387	0,5
42	5,5	50	2626	0,5
43	5,5	50	2864	0,5
44	5,5	50	2387	0,6
45	5,5	50	2626	0,6
46	5,5	50	2864	0,6
47	5,5	50	2387	0,7
48	5,5	50	2626	0,7
49	5,5	50	2864	0,7

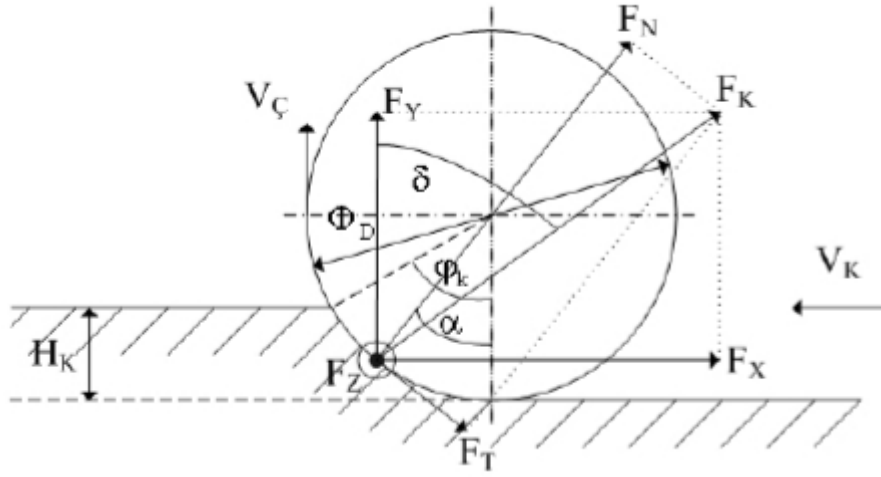
Spesifik enerji faktörünün hesaplanabilmesi için kesme kinematığının iyi bilinmesi gerekir. Kullanılan testere çapı, kesme derinliği, kesme hızı, eksenel kuvvetler gibi bir takım parametreler ile spesifik enerji faktörünün nasıl hesaplandığı, aşağıda “Kesme kinematığı” başlığı altında anlatılmıştır.

2.3. Kesme kinematığı

Diskli kesicilerle kayaç kesme prosesinde, diskin kayacı kesmesi esnasında mekanik etkileşimler sonucu Şekil 2.3 ve Şekil 2.4’te görülen kuvvetler ve bileşenleri oluşur.



Şekil 2.3. Aynı yönlü kesme için kesme işlemi kinematığı (Tönshoff et al. 1993)



Şekil 2.4. Zıt yönlü kesme için kesme işlemi kinematiği (Tönshoff et al. 1993)

Burada;

F_X = Disk kesme yönünde oluşan yatay (x) kuvvet (N)

F_Y = Disk kesme yönünde oluşan düşey (y) kuvvet (N)

F_Z = Disk kesme yönüne dik oluşan aksenal (z) kuvvet (N)

F_N = Normal (Radyal) kuvvet (N)

F_T = Teğertsel (Tanjantiyel) kuvvet (N)

F_K = Kesme kuvveti (N)

V_C = Diskin çevresel hızı (m/sn)

V_K = Kesme (ilerleme) hızı (m/sn)

Φ_D = Disk (testere) çapı (mm)

H_K = Kesme derinliği (mm)

φ_k = Diskin kayaçla temas (kavrama) açısı (Derece)

α = Normal kuvvetin etki (aşındırma) açısı (Derece)

Yatay (F_X), düşey (F_Y), aksenal (F_Z) kuvvetler ölçümler sonucu bulunan kuvvetlerdir. Bu kuvvetlerin yardımıyla aynı yönlü kesme işlemi için aşağıdaki bağıntılar kullanılarak normal, teğetsel ve kesme kuvvetleri de hesaplanabilir. (Tönshoff et al. 1993)

Aynı yönlü kesme durumunda Normal kuvvet F_N ve Teğetsel kuvvet F_T hesaplanırken 2.3 ve 2.5 formüllerinden yararlanılmaktadır.

$$\cos\delta = \frac{F_Y}{F_K}, \quad \sin\delta = \frac{F_X}{F_K} \quad (2.1)$$

$$F_N = F_K \cdot \cos(\alpha - \delta) \quad (2.2)$$

$$F_N = F_Y \cos\alpha + F_X \sin\alpha \quad (2.3)$$

$$F_T = F_K \sin(\alpha - \delta) \quad (2.4)$$

$$F_T = F_Y \sin\alpha - F_X \cos\alpha \quad (2.5)$$

Benzer şekilde zıt yönlü kesme durumunda Normal kuvvet F_N ve Teğetsel kuvvet F_T hesaplanırken de 2.6 ve 2.7 formüllerinden yararlanılmaktadır.

$$F_N = F_Y \sin\alpha + F_X \cos\alpha \quad (2.6)$$

$$F_T = F_X \cos\alpha - F_Y \sin\alpha \quad (2.7)$$

Normal ve Teğetsel kuvvetler belli olduğunda Kesme kuvveti (F_K) 2.8 trigonometrik bağıntısı ile hesaplanabilmektedir.

$$F_K = \sqrt{F_N^2 + F_T^2} \quad (2.8)$$

Testerinin kayacı kavrama açısı ve normal kuvvetin etki açısı denklem 2.9 ve 2.10 yardımıyla hesaplanabilir.

$$\varphi_k = \cos^{-1}\left(1 - \frac{2Hk}{\phi_D}\right) \quad (2.9)$$

$$\alpha = w\varphi_k \quad (2.10)$$

Aşındırma açısına ait w faktörünün, Tobias (1965) tarafından 0.7 alınması önerilmiştir. Motor gücü P_c denklem 2.11 ile belirlenebilmektedir.

$$P_c = F_T \cdot V_C \quad (2.11)$$

Motor gücünün belirlenmesinin ardından birim zamanda kesilen hacim başına harcanan enerji miktarı olan Özgül kesme enerjisi (SE_m) 2.12 denklemi ile hesaplanabilmektedir.

$$SE = \frac{P_c}{Q_w} = \frac{F_T V_C}{H_K G_s V_K} \quad (2.12)$$

Denklem 2.12'de G_s , elmas soket genişliğidir.

Spesifik enerji hesaplamalarında kullanılan diğer iki önemli parametre kesme kuvvetleri oranı (μ) ve kesme hızları oranı (q) olarak denklem 2.13 ve 2.14'de verilmektedir.

$$\mu = \frac{F_T}{F_N} \quad (2.13)$$

$$q = \frac{V_K}{V_C} \quad (2.14)$$

Yukarıda 2.1'den 2.14'e verilen formüller yardımıyla hesaplanan F_N , F_T , F_K , P_c , μ , q , ve SE_m değerleri deney verileri dosyasına kaydedilmektedir.

3. YAPAY SİNİR AĞLARI (YSA)

İnsan beyninin çalışma prensibini taklit etmek suretiyle geliştirilen sistemlerdir. Bu ağlarda birbirlerine belli ağırlıklar ile bağlanmış sinir hücreleri bulunmaktadır. Sinir hücreleri kendilerine gelen girdileri basit bir işleme tutup, kendinden sonra gelen hücreye gönderirler. Hücreleri birbirlerine bağlayan ağırlıklar, çeşitli öğrenme metodlarıyla eğitilmiş ağlarda sürekli olarak değişmekte ve genel anlamda ağı en yakın sonuca götürmeye çalışmaktadır.

1999 yılında Haykin tarafından yapılan tanım şu şekildedir: Yapay sinir ağı basit işlemler yapılan birimlerden oluşan, kendisine sunulan örnekleri hafızasına alıp daha sonra kullanan, yoğun olarak paralel dağıtılmış bir işlemcidir. Çevreden edinilen bilgi sinaplar arasındaki ağırlıklar sayesinde korunur. Bir başka tanımda insan beyninin çalışmasından esinlenerek bir programlama mantığı ile bu çalışma sistemini taklit etmeye çalışmaktır (Civalek 1998).

Yapay sinir ağlarını klasik bilgisayar programlarından ayıran temel özellik; kendisine verilen dataları hafızasına alıp bunları işlemesi, yorumlaması ve kendisine daha önceden gösterilmemiş olaylar hakkına fikir yürütebilmesidir. Ezberleme ve yorumlama yeteneği sayesinde insan beyninin çok basit bir taklidi niteliği taşımaktadır. Bilgisayar programlarının doğal sinir sistemlerinden etkilenmesi sonucu ortaya çıkan yapay sinir ağları ile çözümleri analitik olarak mümkün olmayan, bir başka deyişle doğadaki fiziksel mekanizmasının matematik modeli tam olarak kurulamayan sistemlerin çözümü kolaylaşmıştır.

3.1. Yapay Sinir Ağlarının Tarihçesi

İlk yapay sinir ağı çalışmaları 1940'lı yıllarda başlamıştır. 1943 yılında Warren McCulloch ve Pitts yapay sinir hücresini geliştirmişler, beynin çalışma prensiplerinden yararlanan elektronik devreler kurmayı başarmışlardır. Ağın ürettiği sonuçlar basit gruplandırma şeklindedir. Daha sonra çalışmalar bu ağların öğrenmesi ile ilgili metotların geliştirilmesi

yönünde devam etmiştir. 1949 yılında Donalı Hebb “Hebbian Öğrenme Kuralı” diye adlandırdığı bir yöntem geliştirmiştir. 1958 yılında Frank Rosenblatt’in basit algılayıcı modeli (perceptron) bulması çok katmanlı yapay sinir ağlarının temelini teşkil etmiştir.

1960 yılında ise yeni bir sistem olan Adaline (Adaptive Linear Element) Widrow ve Hoff tarafından geliştirilmiştir. Yine adaline ünitelerinin birleştirilmesiyle Madaline sistemi geliştirilmiştir.

1960’lı yılların sonunda Minsky ve Pappert yapay sinir ağlarında bir dönem duraklamaya sebep olacak “ Perceptrons” adlı kitabı yayınlamışlardır. Bu kitabın ana fikri perceptronların non-lineer sistemler için çözüm üretemediğidir. Bunu fikirlerini XOR adını verdikleri problemle kanıtlanmıştır. Böylelikle birçok araştırmacı yapay sinir ağları ile ilgilenmekten vazgeçmiş, araştırma enstitüleri bu konuda yapılan çalışmalarını desteklemeyi bırakmıştır. Konuya alâka en az seviyeye düşmüştür. Buna rağmen az da olsa birkaç araştırmacı çalışmalarına devam etmiş ve XOR problemini çözmüştür (Öztemel 2003).

1980’li yıllara gelindiğinde araştırmacılar çalışmalarına hız vermişlerdir. Bunlardan en önemlileri Kohonen, Hopfield ve Rummelhart’dır. Kohonen 1982’de “self organizing maps” adlı yayınında öğretmensiz(denetimsiz) öğrenme konusunda başarı sağlamıştır. Aynı yıllarda Rummelhart, Hinton ve Williams çok katmanlı yapay sinir ağlarını geliştirmişler ve XOR probleminin çözümüne büyük ölçüde katkıda bulunmuşlardır (Rummelhart vd, 1986). Yine Hopfield yapay sinir ağlarının bilgisayarda kullanılması sonucu mühendislik problemlerine katkıda bulunduğunu fikrini kanıtlamıştır (Hopfield 1982). Yine bu yıllarda Broomhead ve Love Radyal Basis Functions (Radyal tabanlı fonksiyonlar) modelini geliştirmişlerdir. Böylelikle yapay sinir ağlarına olan ilgi ve paylaşım tekrar artmıştır. Başka bir deyişle 1980’li yıllar gelişim sürecinde altın yıllar olarak değerlendirilebilir.

3.2. Yapay Sinir Ağlarında Olumlu – Olumsuz yönler

Yapay sinir ağları henüz gelişim sürecinde olan bir sistem olduğundan, karanlıkta kalan ve aydınlatılmayı bekleyen sırlarla doludur. Öncelikle nöronlar arası ağırlıkların ilk değerlerinin neye göre belirleneceği bilinmemektedir. Uygun gizli katman sayısı ve gizli katmanda ya da katmanlarda kullanılacak nöron sayısını belirleyen bir kural yoktur. Bunlar tamamen deneme-yanılma sonucu tespit edilebilmektedir.

Hangi problemde hangi ağ mimarisinin kullanılacağı da sadece tecrübeler yoluyla kestirilebilmektedir. Yani her ne kadar seçilen ağ mimarisi doğru sonuçlar verse dahi her zaman için kullanılanıdan daha doğru sonuç veren bir ağ olabilir.

Ağın eğitiminin ne zaman biteceğine dair bir kural da yoktur. Yapay sinir ağlarını kesin çözüm beklenen veya çok karmaşık işlemler gerektiren problemlerde kullanmak uygun değildir. Ağa gösterilecek olan dataların gösterim şekli kullanıcının kabiliyetine bağlıdır. Problem doğru şekilde aktarılmazsa ağ düzgün cevaplar üretemez. Sadece nümerik bilgiler girdi olarak kullanılabilir (Hiçyılmaz 2007).

Buna karşın yapay sinir ağları non-lineer problemlere çözüm üretebilirler. Tabiattaki birçok problem non-lineer yapıda olmasına rağmen bazı kabuller yapılarak lineer (doğrusal) yapıya dönüştürülmektedir. Yapay sinir ağlarında böyle bir problem yoktur.

Sinir ağları eksik bilgi ile çalışabilir. Girdi setinde kullanılan verilerin bir kısmı kaybedilmiş veya sınırlı sayıda olabilir. Bilinen veriler kullanılarak hatanın tolare edilebilme şansı vardır. Kendi kendini organize etme ve öğrenebilme yetenekleri, pozitif yönleridir (Tuzcuoğlu 2003).

Yapay sinir ağlarında nöronların birbirinden bağımsız çalışabilme yetenekleri vardır. Bu sebepten her bir nöron kendi başına işlem yapabilir ve yerel hatalar olsa dahi hata diğer nöronlar tarafından telafi edilebilir. Böylelikle esnek bir yapı gösterirler. Ağın bir bölümü

zarar gördüğünde performans düşüklüğü olsa dahi ağ çalışmaya devam edecektir. Tamamen göçme söz konusu olmayacaktır. Ayrıca örüntü tanıma, genelleme yapabilme ve adaptasyon kabiliyetleri vardır.

3.3. Yapay Sinir Ağlarını Oluşturan Temel Elemanlar

Yapay bir nöron (proses elemanı) beş temel öğeden oluşur. Bunlar girdiler, ağırlıklar, toplama fonksiyonu, aktivasyon formülü ve çıktılardır. Bu elemanlara kısaca değinmek gerekirse:

3.3.1. Girdiler

Proses elemanın dış ortamdan bilgileri (verileri) alan elemanlarıdır. Veriler bu safhada bir işleme tabi tutulmadan aynen iletilirler.

3.3.2. Ağırlıklar

Bir nörona eş zamanlı birçok veri girişi olabilir. Bu veriler nörona gelirken kendine ait olan ağırlık değeriyle çarpılır. Ağırlık değerleri pozitif, negatif ya da sıfır olabilir. Ağırlık değerleri kullanılan birçok öğrenme kurallarına ve ağın mimarisine göre değişebilmektedir. Girdilerin toplama fonksiyonu üzerindeki etkileri ağırlıkları miktarınca olur.

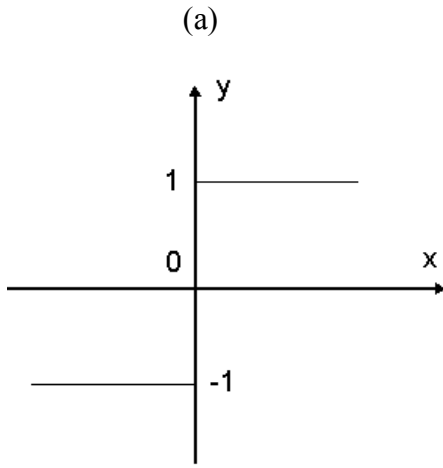
3.3.3. Toplama Fonksiyonu

Proses elemanında girdiler ağırlıklarla çarpıldıktan sonra toplama fonksiyonuna gönderilirler. Girdiler ve ağırlıklar, aktivasyon fonksiyonuna gitmeden önce birçok şekilde bütümleştirilebilirler. Genel olarak girdiler ve ağırlıkların çarpılmasıyla bulunan değerler toplanmak suretiyle aktivasyon fonksiyonuna gönderilir. Kimi zaman kullanıcının tercihinine göre bu değerlerin en büyüğü, en küçüğü ya da kümülatif toplamı vb. kullanılabilir. Hangi problemde hangi toplama fonksiyonunun kullanılacağına dair henüz bulunmuş bir formül

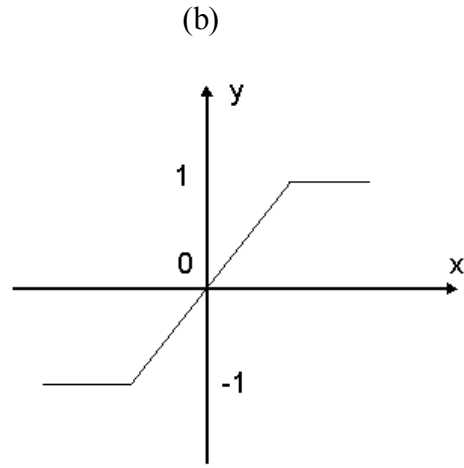
yoktur. Her nöron aynı toplama fonksiyonunu kullanabileceği gibi ayrı ayrı fonksiyonlarda kullanabilir.

3.3.4. Aktivasyon Fonksiyonu

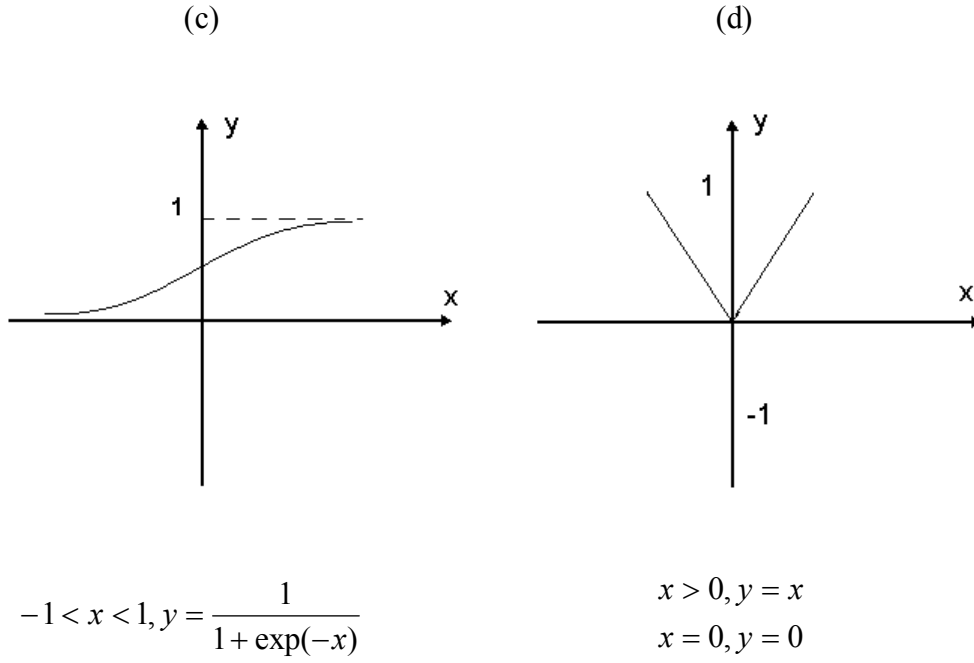
Toplama fonksiyonunun çıktısı aktivasyon formülünde girdi olarak kullanılır. Ancak bu girdileri belirli bir seviyenin üstünde tutmak için bir eşik değeri seçilmelidir. Toplam fonksiyonundan gelen değerler bu eşik değerinden yukarıda ise işleme tabi olurlar. Aktivasyon fonksiyonu girdileri algoritma ile gerçek bir çıktıya dönüştürür. Aktivasyon fonksiyonunda genel olarak türevi alınabilen fonksiyonlar kullanılır. En çok kullanılan sigmoid fonksiyonudur. Sıkça kullanılan aktivasyon fonksiyonları ve şekilleri aşağıda verilmiştir.



$$\begin{aligned}x > 0, y &= 1 \\x < 0, y &= -1 \\x = 0, y &= 0\end{aligned}$$



$$\begin{aligned}0 < x < a, y &= x \\x > a, y &= 1 \\-a < x < 0, y &= x \\x < -a, y &= -1\end{aligned}$$



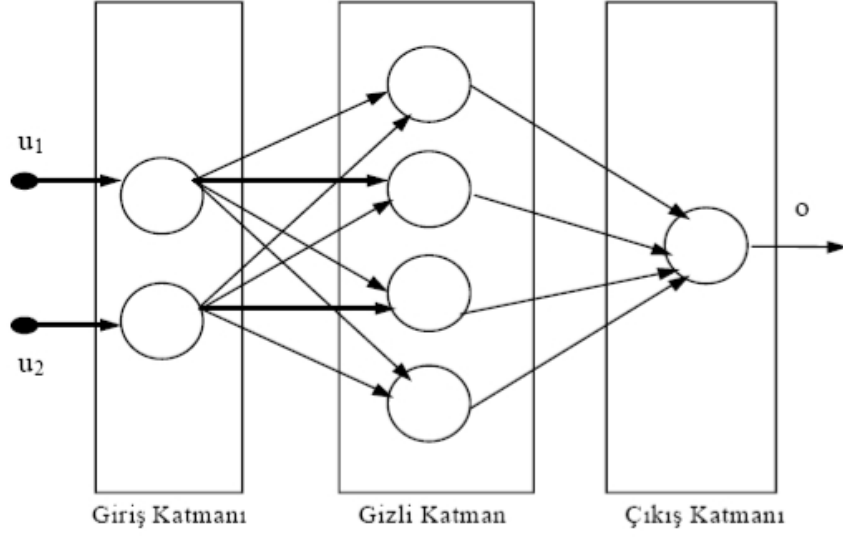
Şekil 3.1. Kullanılan bazı önemli sigmoid fonksiyonları

3.3.5. Çıktılar

Yapay nöronlarda da biyolojik nöronlara benzer şekilde birçok girdi olmasına rağmen tek bir çıktı değeri vardır. Genel anlamda çıktı, aktivasyon fonksiyonunun çıktısıdır. Bazı ağ yapılarında çıktılar komşu proses elemanlarının çıktılarıyla birleştirilerek kullanılabilir. Bazılarında ise nöronlar arası yarış vardır. Yarışmayı kazanan nöron çıktı olarak kullanılır.

3.4. Yapay Sinir Ağlarında Katmanlar

Temelde yapay sinir ağları bir giriş, bir çıkış ve bir ya da daha fazla sayıdan oluşabilen gizli katmandan meydana gelmektedir. Nöronlar katmanlarda bulunurlar. Dış dünyadan gelen veriler ilk olarak girdi katmanına gelirler. Girdi katmanındaki nöron adedi problemde verilen değişkenlere bağlı olarak değişebilmektedir. Yine bu katmanda veriler bir işleme tabi tutulmadan gizli katmanlara iletilirler.



Şekil 3.2. Yapay sinir ağlarında katmanlar

Gizli katmanlar yapay sinir ağlarının işlem merkezleridir. Bir ağda bir tane gizli katman olabileceği gibi birden fazla sayıda gizli katman da olabilir. Bu sayı kullanıcının sezgilerine ve tecrübelerine bağlı olarak seçilir. En yaklaşık sonuçları veren gizli katman adedini belirleyen bir formülasyon henüz geliştirilememiştir.

Çıktı katmanı ise işlemde geçen dataların sonuçlarının toplandığı katmandır. Girdi katmanında olduğu gibi çıktı katmanında da probleme bağlı olarak bir ya da daha fazla nöron bulunabilir.

3.5. Yapay Sinir Ağlarında Öğrenme Prensipleri

3.5.1. Denetimlerine Göre Öğrenme Şekilleri

Yapay sinir ağlarında genel olarak üç çeşit öğrenme şekli vardır. Birincisi denetimli (öğretmenli) öğrenme, ikincisi destekleyici öğrenme, üçüncüsü ise denetimsiz (öğretmensiz) öğrenmedir. Bu üç öğrenme türünü kısaca inceleyelim.

3.5.1.1. Denetimli (Öğretmenli) Öğrenme

Sıkça kullanılan bir öğrenme şeklidir. Öğretmenli öğrenme diye de adlandırılabilir. Bu öğrenme şeklinde ağın ürettiği çıktı değerleri ile eldeki mevcut çıktı değerleri karşılaştırılır. Veriler ağa sunuldukça başlangıçta seçilen ağırlık değerleri iterasyon sonucu değişerek en yakın sonuca yakınsamaya çalışır. Toplam hata minimizasyonu (kullanıcı tarafından yeterli görülene kadar) iterasyonlarla sağlanmaya çalışılır.

Eğitim safhası geçtikten sonra ağın kullanılabilirliğinin kontrol edilmesi gerekir. Buna test süreci denir. Test sürecinde eğitim seti içinden ağın eğitiminde kullanılmamış veriler alınarak ağa sunulur. Yani eğitim setinin verilerinin tümü ağı eğitmek için kullanılmaz, bir kısmı da ağın yeterli derecede öğrenip öğrenmediğini kontrol etmek için kullanılır. Bu safhada eğitim setinde kullanılmamış olan datalar (girdi-çıkıtı değerleri) ağa gösterilerek performans değerlendirmesi yapılır. Test seti olarak seçilen verilerin geniş kapsamlı olması gerekir. Yani eğitim setini en iyi şekilde temsil etmelidir. Elde edilen sonuçlar kabul edilebilir hata sınırları içindeyse ağın eğitimi tamamlanmıştır denir ve yeni veriler gösterilerek ağın tahminde bulunması istenir. Şayet deney setine ağın verdiği cevaplar hata sınırlarını aşıyorsa ağ yeni baştan eğitilmelidir. Farklı sonuçlara ulaşabilmek için nöronlar arası ağırlık değerleri, gizli katman sayısı ya da katmanlardaki nöron sayıları, iterasyon sayısı vb. değişiklikler yapılabilir (Sağiroğlu vd. 2003).

3.5.1.2. Destekleyici (Reinforcement) Öğrenme

Bu tür stratejide de öğrenen sisteme bir öğretmen yardımcı olur. Fakat öğretmen her girdi seti için olması gereken (üretilmesi gereken) çıktı setini sisteme göstermek yerine sistemin kendisine gösterilen girdilere karşılık çıktısını üretmesini bekler ve üretilen çıktının doğru veya yanlış olduğunu gösteren bir sinyal üretir. Sistem, öğretmenden gelen bu sinyali dikkate alarak öğrenme sürecini devam ettirir (Öztemel 1993).

3.5.1.3. Denetimsiz (Öğretmensiz) Öğrenme

Bu tür stratejide sistemin öğrenmesine yardımcı olan herhangi bir öğretmen yoktur. Sisteme sadece girdi değerleri gösterilir. Örneklerdeki parametreler arasındaki ilişkileri sistemin kendi kendisine öğrenmesi beklenir. Bu, daha çok sınıflandırma problemleri için kullanılan bir stratejidir. Yalnız sistemin öğrenmesi bittikten sonra çıktıların ne anlama geldiğini gösteren etiketlendirmenin kullanıcı tarafından yapılması gerekmektedir (Öztemel 1993).

3.5.1.4. Karma Stratejiler

Yukarıdaki 3 stratejiden birkaçını birlikte kullanarak öğrenme gerçekleştiren ağlar da vardır. Burada kısmen öğretmenli, kısmen ise öğretmensiz olarak öğrenme yapan ağlar kastedilmektedir. Radial tabanlı yapay sinir ağları (RBN) ve olasılık tabanlı ağlar (PBNN) bunlara örnek olarak verilebilir.

3.6. Başlıca Öğrenme Kuralları

3.6.1. Hebb Kuralı

Bugün araştırmacıların kullandığı birçok öğrenme kuralı vardır. En eski ve en iyi bilinen öğrenme kuralı Donald Hebb tarafından 1949 yılında “The Organization Of Behaviour ” adlı kitabında bahsettiği Hebb kuralıdır. Temel çalışma prensibi şudur; Eğer bir nöron diğer bir nörondan bir bilgi alırsa ve ikiside matematiksel olarak aynı işarete sahipse nöronlar arası bağlantı kuvvetlenir.

3.6.2. Hopfield Kuralı

Hebb kuralı ile benzerlik göstermektedir. Eğer beklenen girdi ve çıktıların her ikiside aktif ya da pasifse hücreler arası bağlantı öğrenme katsayısı kadar kuvvetlendirilir ya da zayıflatılır. Öğrenme katsayısının belirli bir değeri yoktur, kullanıcı tarafından atanır.

3.6.3. Kohonen Kuralı

Bu kural Teuvo Kohonen tarafından biyolojik sinir ağlarından esinlenerek oluşturulmuştur. En büyük çıktıya sahip olan hücre çevresindeki rakipleri sınırlama yetkisine sahip olur. Sadece kazanan hücre ve komşu hücreler bağlantı ağırlıklarını düzenleyebilirler.

3.6.4. Delta Kuralı

Hebb kuralının geliştirilmiş halidir. En çok kullanılan öğrenme kurallarından birisidir. Kuralın dayandığı temel düşünce, beklenen değerler ile ağın ürettiği değerler arasındaki farkın (delta) en aza indirilebilmesi için ağırlık değerlerinin değiştirilmesidir. Bağlantı ağırlıkları değiştikçe ağın hatası da (ortalama karesel hata) azalmaktadır. Toplam hatanın geri yayılımı aktivasyon fonksiyonunun türevi alınarak yapılmaktadır. Dolayısıyla hata “eğimli iniş” (gradient descent) adı verilen bir sistemle en aza indirgenmeye çalışılmaktadır. Her bir nörona belli bir oranda hata paylaştırılmakta ve bir sonraki iterasyonun katsayıları bulunmaktadır. Bu işlem giriş katmanındaki nöronlara kadar tüm ağ boyunca yapılır. Burada yanlış anlamaya sebebiyet vermemek için bir açıklama yapmak gerekirse, hatanın geriye doğru yayılması ağın geri-yayımlı (back-propagation) bir ağ olduğunu göstermez. Ağın ileri ya da geri yayımlı olması nöronlar arası bağlantı geometrisiyle alakalıdır. Hatanın geri yayılımı ise öğrenme kuralına bağlı olarak gelişmektedir (Elmas 2003).

3.7. Çok Katmanlı Algılayıcılar

Bugün için yapay sinir ağlarından beklenen non-lineer (doğrusal olmayan) problemlere çözüm bulabilmeleridir. Çünkü mühendislik problemlerinin çok büyük bir kısmı doğrusal olmayan yapıdadır. Doğrusal olmayan problemlerin yapay sinir ağları ile çözülebilmesi için “çok katmanlı algılayıcılar” (MLP: ‘Multi Layered Perceptrons’) geliştirilmiştir. Bir MLP

modeli bir giriş, bir ya da daha fazla ara katman, bir de çıkış katmanından oluşan ve her katmanda bir yada daha fazla sayıda nöron bulunduran bir yapıdan ibarettir. Bir katmandaki bütün nöronlar önceki katmanla bağlantılıdır ve bilgi akışı ileri yöndedir. İleri beslemeli sinir ağı diye adlandırılmasının nedeni budur. Ara katman sayısını ya da katmanlarda kullanılacak nöron sayılarını kullanıcı kendi tecrübelerine göre ya da deneme yanılma metodu ile bulur (Sağiroğlu vd. 2003).

3.8. Hızlı Algoritmalar

Hızlı algoritmalar genel olarak iki kategoriye ayrılabilir. İlk kategorideki algoritmalar, deneme yanılma tekniklerini kullanarak, standart gradyen azalması (steepest descent) yönteminden daha iyi sonuçlar verebilir. Deneme-yanılma işlemlerini kullanan geriye yayılım algoritmaları; momentum terimli geriye yayılım, öğrenme hızı değişen geriye yayılım ve esnek geriye yayılım algoritmalarıdır. Hızlı algoritmaların ikinci kategorisindeki algoritmalar, standart sayısal optimizasyon yöntemlerini kullanır. Bu algoritmalar:

- 1- Eşlenik gradyen öğrenme algoritması
- 2- Newton öğrenme algoritmaları
- 3- Levenberg – Marquardt öğrenme algoritması

3.8.1. Levenberg-Marquardt Algoritması

Geriye yayılım algoritması, yapay sinir ağlarında en çok kullanılan algoritmadır. Geriye yayılım öğrenmesi sırasında ağ, her giriş örüntüsünü, çıkış nöronlarında sonuç üretmek üzere gizli katmanlardaki nöronlardan geçirir. Daha sonra çıkış katmanındaki hataları bulabilmek için, beklenen sonuçla, elde edilen sonuç karşılaştırılır. Bundan sonra, çıkış hatalarının türevi çıkış katmanından geriye doğru gizli katmanlara geçilir. Hata değerleri bulunduğundan sonra, nöronlar kendi hatalarını azaltmak için ağırlıklarını ayarlar. Ağırlık değiştirme denklemleri, ağdaki performans fonksiyonunu en küçük yapacak şekilde düzenlenir.

İleri beslemeli ağlarda kullanılan öğrenme algoritmaları, performans fonksiyonunu en küçük yapacak ağırlıkları ayarlayabilmek için, performans fonksiyonunun gradyenini kullanırlar. Geriye yayılım algoritması da, ağ boyunca gradyen hesaplamalarını geriye doğru yapar. En basit geriye yayılım öğrenme algoritması gradyen azalması algoritmasıdır. Bu algortmada ağırlıklar, performans fonksiyonunun azalması yönünde ayarlanır. Fakat bu yöntem, pek çok problem için çok yavaş kalmaktadır. Bu algortmadan daha hızlı, daha yüksek performanslı algortmalar da vardır.

Öğrenme algoritmaları, kendisinden önce geliştirilen algortmalara alternatif olarak ortaya çıkmıştır ve önceki algortmaların iyi yönlerini geliştirip, kötü yönlerini azaltmaya yönelmiştir. Levenberg – Marquardt algortması da, Newton ve Gradyen Azalması algortmalarının en iyi özelliklerinden oluşur ve kısıtlamalarını ortadan kaldırır.

Eşlenik gradyen öğrenme algortmasına alternatif olarak sunulan Newton yöntemlerinde, temel adım Hessian matrisini elde etmektir. Hessian matrisi, performans fonksiyonunun ağırlıklara göre ikinci dereceden türevlerinden oluşan bir matristir. Hessian matrisi, ağırlık uzayının farklı doğrultularındaki gradyen değişimini gösterir.

$$H(n) = \frac{\partial^2 E(n)}{\partial w^2(n-1)} \quad (1)$$

Burada H Hessian matrisi, E performans fonksiyonu, w ağıın sinaptik ağırlığıdır. Performans fonksiyonu, duruma göre toplam ani hata veya ortalama karesel hata olarak alınabilir.

$$E(n) = E_{ort}(n) = \frac{1}{N} \sum_{n=1}^N \frac{1}{2} \sum_{j \in C} e_j^2(n) \quad (2)$$

Burada N eğitim kümesindeki toplam örüntü sayısını, e_j hata işaretini, C ağın çıkış katmanındaki bütün nöronları içeren kümeyi göstermektedir. d_j istenen değer, y_j ağın çıkışı olmak üzere hata işareti,

$$e_j(n) = d_j(n) - y_j(n) \quad (3)$$

olarak bulunabilir. Hessian matrisi hesaplandıktan sonra, tersi bulunarak ağırlıklar yenilenebilir. Ancak Hessian matrisi çok karmaşık ve ileri beslemeli bir yapay sinir ağı için hesaplanması zor bir matristir. Newton yöntemlerinin içinde, ikinci dereceden türevlerin hesaplanmadan işlem yapılan bir sınıf vardır. Bu sınıftaki yöntemler, quasi – Newton yöntemleri olarak adlandırılırlar. Quasi – Newton yöntemleri, algoritmanın her iterasyonunda, Hessian matrisinin yaklaşık bir şeklini kullanır.

Levenberg – Marquardt algoritması da quasi – Newton yöntemleri gibi, Hessian matrisinin yaklaşık değerini kullanır. Levenberg – Marquardt algoritması için Hessian matrisinin yaklaşık değeri şu şekilde bulunabilir:

$$H(n) = J^T(n)J(n) + \mu I \quad (4)$$

(4) denklemindeki μ Marquardt parametresi, I ise birim matristir. Burada J matrisi, Jakobien matrisi olarak adlandırılır ve ağ hatalarının ağırlıklara göre birinci türevlerinden oluşur:

$$J(n) = \frac{\partial e(n)}{\partial w(n-1)} \quad (5)$$

(5) denkleminde e, ağ hataları vektörüdür. Jakobien matrisi, hesaplamada Hessian matrisinden daha kolay olduğu için tercih edilir. Ağın gradyeni,

$$g(n) = J^T(n)e(n) \quad (6)$$

olarak hesaplanır ve ağırlıklar (7) denkleminde göre değiştirilir:

$$w(n+1) = w(n) - [H(n)]^{-1} g(n) \quad (7)$$

Marquardt parametresi, μ , skaler bir sayıdır. Eğer μ sıfırsa, bu yöntem yaklaşık Hessian matrisini kullanan Newton algoritması; eğer μ büyük bir sayı ise, küçük adımlı gradyan azalması yöntemi haline gelir. Newton yöntemleri, en küçük hata yakınlarında daha hızlı ve kesindir. Her başarılı adımdan sonra, yani performans fonksiyonunun azalmasında μ azaltılır ve sadece deneme niteliğindeki bir adım performans fonksiyonunu yükseltecekse μ arttırılır. Bu yöntemle, algoritmanın her iterasyonunda, performans fonksiyonu daima azaltılır.

Genel olarak Levenberg – Marquardt algoritması yavaş yakınsama probleminden etkilenmez. Burada hedef, performans fonksiyonun en küçük yapacak ağırlık değerini bulmaktır. (Bolat ve Kalenderli 2003)

3.9. Matlab’da YSA Modellenmesi

Bu kısımda, Matlab programı kullanılarak, ileri beslemeli çok katmanlı bir ağırlık tasarımının nasıl yapılacağı anlatılmıştır. İleri beslemeli bir yapay sinir ağırlık tasarımının nasıl oluşturulduğu ve nasıl eğitildiği, ilgili program komutları verilerek anlatılmıştır.

Matlab’da ileri beslemeli bir ağırlık oluşturmak için “newff” komutu kullanılır. newff komutunun yazım biçimi aşağıda verilmiştir:

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

Burada;

PR - R elemanlı giriş vektörünün minimum ve maksimum değerlerini içeren $R \times 2$ 'lik matris

S_i - i'nci katmanda bulunan nöron sayısı.

TF_i - i'nci katmanın transfer fonksiyonu, varsayılan= 'tansig'.

BTF - Geriye yayılım ağ eğitim fonksiyonu, varsayılan = 'trainlm'.

BLF - Geriye yayılım ağırlık/bias öğrenme fonksiyonu, varsayılan = 'learnsgdm'.

PF - Performans fonksiyonu, varsayılan = 'mse' dir.

Aşağıdaki kod ile 2 katmanlı bir ağ oluşturmaktayız. Bu problemde iki elemanlı bir giriş vektörü vardır. Giriş vektörünün ilk parametresinin değer aralığı -1 ve 2 olarak belirtilmiştir. İkinci parametrenin değer aralığı ise 0 – 5 olarak verilmiştir. İlk katmanda 3 nöron, ikinci katmanda bir diğer ifadeyle çıkış katmanında ise 1 nöron bulunmaktadır. İlk katmanın transfer fonksiyonu tanjant sigmoid, ikinci katmanındaki ise lineer'dir. Öğrenme algoritması, gradyent azaltım algoritması traingd'dir. Problem iki girişli bir çıkışlı bilinmeyen bir sistem olarak düşünülebilir. Amacımız örnek giriş ve çıkış verileri vererek sistemi karakterize etmektir.

```
net=newff([-1 2; 0 5],[3,1],{'tansig','purelin'},'traingd');
```

Bu komut ağ nesnesi oluşturur ve aynı zamanda ağırlık ve bias değerlerinin ilkendirme (initialization) işlemini rastgele değerler atamak suretiyle gerçekleştirir. Haliyle ağ eğitime hazır olarak beklemektedir. Bazı durumlarda ağı tekrar ilkendirmek (reset'lemek) gerekebilir. Bu durumda;

```
net = init(net);
```

komutu kullanılabilir.

Tasarlanan ağın simülasyonu için “sim” komutu kullanılır. Simülasyon’dan kasıt, ağa bir giriş değeri verip ağın çıkışını hesaplatmaktır.

```
p = [1;2];  
a = sim(net,p)  
a =  
-0.1011
```

Burada p ağın giriş verisini temsil etmektedir. Ağın iki girişi olduğu için p matrisi iki satırdan meydana gelmektedir. Görüleceği üzere ağ bir çıkış üretmektedir. Ancak bu, ağ henüz eğitilmediği ve ağırlık değerleri rasgele atandığı için arzu edilen çıkış değildir. Diğer bir yandan ağı çoklu örneklerle de simule edebiliriz.

```
p = [1 3 2;2 4 1];  
a=sim(net,p)  
a =  
-0.1011 -0.2308 0.4955
```

Burada iki girişli ağ için 3’er adet değer verilmiştir. Haliyle ağ her bir örnek için bir çıkış üretmiştir. Buradan açıkça görülüyor ki ağın girişi için (simülasyon veya eğitim aşamasında) organize edilecek matrisin satırları parametrelere, sütunları ise örneklere karşılık gelmektedir.

Ağırlıklar ve bias değerleri ilklendirildikten sonra ağ eğitime hazırdır. Eğitim işleminde ağın davranışını ortaya koyacak giriş (p) ve çıkış (t) verileri gereklidir. Bunlar eğitim verisi olarak adlandırılır. Eğitim verileri çözüm uzayının tamamını veya en azından büyük bölümünü temsil edecek şekilde seçilmelidir. Eğitim esnasında ağırlık ve bias değerleri iteratif olarak hata fonksiyonunu minimize edecek şekilde güncellenir. Hata fonksiyonunun MSE (ortalama karesel hata) olması durumunda hata, arzu edilen çıkış (t) ile ağın o anki çıkışı arasındaki farkın karesinin ortalamasıdır.

YSA'da kullanılan en temel öğrenme algoritması geriye yayılım (back propagation) algoritmasıdır. Basit anlamda ağırlıklar negatif gradyent yönünde güncellenir. Pek çok geriye yayılım algoritması vardır. En temel yapısı şu şekildedir:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \mathbf{g}_k$$

burada, x_k , ağırlık vektörünün k anındaki değerlerini, g_k , k anındaki gradyenti, α_k ise öğrenme oranını temsil etmektedir. Toplu gradyent azaltım (batch gradient descent) eğitim algoritması Matlab'da traingd ile temsil edilir.

Verdiğimiz problemin eğitim verilerini oluşturup ağın eğitimini şu şekilde gerçekleştirebiliriz:

$$p = [-1 -1 2 2; 0 5 0 5];$$

$$t = [-1 -1 1 1];$$

burada, p matrisi ağın eğitimi için kullanılacak giriş verisini temsil etmektedir. Görüldüğü üzere 2 satırdan ve 4 sütundan oluşmaktadır. Problem iki parametrelili olduğu için satır sayısı 2'dir. Ağın eğitimi için ise 4 örnek verilmektedir. Ağın çıkışını t matrisi temsil etmektedir. Her bir örnek için birer çıkış verilmiştir. Daha açık bir ifadeyle verilen sistemin doğruluk tablosu şu şekildedir:

p1	p2	t
-1	0	-1
-1	5	-1
2	0	1
2	5	1

Ağın eğitimi ile ilgili temel parametreleri şu şekilde düzenleyebilirsiniz:

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

Burada, “show” parametresi kaç iterasyonda bir eğitim durumunun Matlab ekranına aktarılacağını belirlemekte, “epoch” parametresi iterasyon sayısını, “goal” parametresi hedeflenen hata değerini, “lr” parametresi ise geriye yayılım algoritması için öğrenme oranını belirtmektedir. “epoch” veya “goal” parametrelerinden birisi sağlandığında eğitim işlemi durdurulacaktır. Ağın eğitime başlamak için train komutu kullanılır.

```
[net,tr]=train(net,p,t);
```

```
TRAINGD, Epoch 0/300, MSE 1.59423/1e-05, Gradient 2.76799/1e-10  
TRAINGD, Epoch 50/300, MSE 0.00236382/1e-05, Gradient 0.0495292/1e-10  
TRAINGD, Epoch 100/300, MSE 0.000435947/1e-05, Gradient 0.0161202/1e-10  
TRAINGD, Epoch 150/300, MSE 8.68462e-05/1e-05, Gradient 0.00769588/1e-10  
TRAINGD, Epoch 200/300, MSE 1.45042e-05/1e-05, Gradient 0.00325667/1e-10  
TRAINGD, Epoch 211/300, MSE 9.64816e-06/1e-05, Gradient 0.00266775/1e-10  
TRAINGD, Performance goal met.
```

Görüleceği üzere 211. iterasyonda hedeflenen hata oranına erişilmiş ve ağın eğitimi durdurulmuştur. Ağın eğitilip eğitilmediğini test etmek için giriş verisi (p) ile ağı tekrar simüle edelim:

```
a = sim(net,p)  
a =  
-1.0010 -0.9989 1.0018 0.9985
```

Görüleceđi üzere ađ arzu edilen ıkıřları sađlamaktadır. Yani YSA verilen sistemi modellemiřtir. Sistemin giriři ve ıkıřı arasındaki bađıntıyı bulmuřtur. Bunu da uygun ađırlık deđerlerini seerek yapmıřtır. Bu ařamadan sonra eđitim verisinde kullanmadıđımız yeni veriler vererek ađı test edebiliriz.

Matlab'da kullanabileceđimiz pek ok eđitim algoritması bulunmaktadır. En temel algoritma yukarda bahsedilen geriye yayılım tabanlı toplu gradyent azaltım algoritmasıdır. Fakat Matlab'da varsayılan eđitim algoritması olarak Levenberg-Marquardt (trainlm) seilmiřtir. Bu algoritma ok hızlıdır ancak alıřabilmesi iin fazla hafızaya ihtiya duymaktadır. Bunların yanı sıra trainbfg ve trainrp algoritmaları da tercih edilebilir. (Demuth vd.)

4. UYGULAMA

Bu bölümde yapılan 2 farklı YSA uygulaması ve sonuçlarından bahsedilmiştir. 4.1’de yapay sinir ağlarını kullanarak mermer kesme işleminde spesifik enerjinin (SE) tahmin edilmesi uygulaması ve 4.2’de ise spesifik enerjinin dört farklı parametreye bağlı olarak tahmini uygulaması anlatılmıştır.

4.1. Mermer Kesme İşleminde Spesifik Enerjinin (SE) YSA ile Tahmini

Bu çalışma, deneysel verilerden elde edilen spesifik enerjiyi YSA metodunu kullanarak tahmin etmeyi amaçlamıştır. Bunun için, mermerin sertliği, kesme hızı ve testere dönüş hızı giriş verisi olarak kabul edilmiş, bu verilerden SE tahmin edilmiştir.

Tahmin yöntemlerinin uygulanmasında kullanılan tüm veriler daha önce yapılan bir deneysel çalışmadan alınmıştır (Çınar 2007). Denizli traverteni ve Bilecik Bej mermerlerinin 0.3, 0.4, 0.5, 0.6, 0.7 m/dk kesme hızları ve 1432, 1910, 2387, 2864 d/d testere dönüş hızları ile kesilmiş ve ölçülen aktif güçten yola çıkarak, birim hacim başına harcanan enerji miktarını ifade eden spesifik enerji (SE) hesaplanmıştır.

4.1.1. Ağın eğitimi için gerekli verilerin hazırlanması

Gerçekleştirilen çalışmada Kesme Derinliği (H_k), İlerleme Hızı (V_k) ve Testere Dönüş Hızı (V_d) parametrelerinin aşağıda verilen aralıklarında, Bilecik Bej mermeri ve Denizli traverteni numuneleri üzerinde toplam 40 adet deney gerçekleştirilmiştir.

Kesme Derinliği (H_k) : 50mm

İlerleme Hızı (V_k) : 0.3, 0.4, 0.5, 0.6 ve 0.7m/dk

Testere Dönüş Hızı (V_d) : 1432, 1910, 2387 ve 2864d/dk

Görüldüğü gibi makine parametrelerinden Kesme Derinliği (H_K) 50mm değerinde sabit tutulmaktadır. Kesme hızı (V_K) 0.3-0.7m/dk aralığında 5 değer almaktadır. Testere dönüş hızı (V_d) ise 1432-2864d/dk aralığında 4 değer almaktadır. Mermercilik üzerine yapılan akademik çalışmalarda testere dönüş hızı yerine testerenin çevresel hızı [V_c] kullanılmakta ve birimi [m/sn] belirlenmektedir. Ancak elektrik mühendisliği açısından testerenin çevresel hızı değil dakikada yaptığı devir sayısı önem arz etmektedir. Bu nedenle kesme deneylerinin literatür ile uyumlu olması açısından testere çevresel hız değerinin 30, 40, 50 ve 60m/sn değerlerine karşılık gelen 1432, 1910, 2387 ve 2864d/dk değerlerindeki Testere dönüş hızları kullanılmıştır.

Kesme derinliği, İlerleme hızı ve Testere dönüş hızı için seçilen parametrelere karşılık yapılacak deney sayısı 20 olmaktadır. Ancak iki tip kayaç için deneyler tekrar edildiğinde toplam deney sayısı 40'a çıkmaktadır. Varyasyon deneyleri aşağıda Çizelge 4.1'de verilen sıraya göre gerçekleştirilmiştir.

Çizelge 4.1. Varyasyon deneyleri işlem sırası

Deney No	Kesme Derinliği [mm]	Testere Dönüş Hızı [d/dk]	İlerleme Hızı [m/dk]
1	50	1432	0,3
2	50	1432	0,4
3	50	1432	0,5
4	50	1432	0,6
5	50	1432	0,7
6	50	1910	0,3
7	50	1910	0,4
8	50	1910	0,5
9	50	1910	0,6
10	50	1910	0,7
11	50	2387	0,3
12	50	2387	0,4
13	50	2387	0,5
14	50	2387	0,6
15	50	2387	0,7
16	50	2864	0,3
17	50	2864	0,4
18	50	2864	0,5
19	50	2864	0,6
20	50	2864	0,7

Çizelge 4.1’de sıralaması verilen varyasyon deneyleri Bilecik Bej mermeri ve Denizli traverteni kayaları için tekrarlanmıştır. Varyasyon deneylerinde söz konusu kayaların kesiminde iki farklı testere kullanılmıştır. Seçilen kayaç numunelerinin farklı fizikokimyasal özellikte olması dolayısıyla aynı tipte testere ile kesilmeye uygun olmaması her kayaç için özel testere kullanımını zorunlu kılmıştır. (Çınar 2007)

Çizelge 4.2. YSA’nın eğitimi için kullanılan veriler

Mermer Sertlik Derecesi	Kesme hızı	Testere dönüş hızı	SE
4	0.3	1432	2.5
4	0.4	1432	2.2
4	0.6	1432	1.95
4	0.7	1432	1.8
4	0.3	1910	2.6
4	0.4	1910	2.2
4	0.6	1910	1.95
4	0.7	1910	1.75
4	0.3	2387	2.85
4	0.4	2387	2.4
4	0.6	2387	2.1
4	0.7	2387	1.85
4	0.3	2864	3.1
4	0.4	2864	2.55
4	0.6	2864	2.1
4	0.7	2864	1.83
3	0.3	1432	1.7
3	0.4	1432	1.4
3	0.6	1432	1.25
3	0.7	1432	1.23
3	0.3	1910	1.85
3	0.4	1910	1.55
3	0.6	1910	1.26
3	0.7	1910	1.24
3	0.3	2387	2.2
3	0.4	2387	1.8
3	0.6	2387	1.5
3	0.7	2387	1.3
3	0.3	2864	2.3
3	0.4	2864	1.85
3	0.6	2864	1.6
3	0.7	2864	1.4

Alınan veriler Matlab'ta ağın eğitim için kullanılmak üzere matris formuna çevrilmiştir. Mevcut 40 adet verinin 32 adedi ağın eğitimi için kullanılmış ve 8 adet veri test işlemi için ayrılmıştır. 0.3, 0.4, 0.6 ve 0.7 m/dk kesme hızlarında yapılan deney sonuçları ağın eğitimi için kullanılmış, 0.5 m/dk kesme hızında yapılan deney sonuçları ise test işleminde kullanılmıştır. Mermerin sertlik derecesi, kesme hızı ve testere dönüş hızı değerleri ağın girişi olarak belirlenmiştir. Ağın çıkışı ise spesifik enerji faktörü(SE)'dür. Ağın eğitimi için kullanılan veriler, Çizelge 4.2'de görülmektedir.

4.1.2. Yapay sinir ağının eğitilmesi

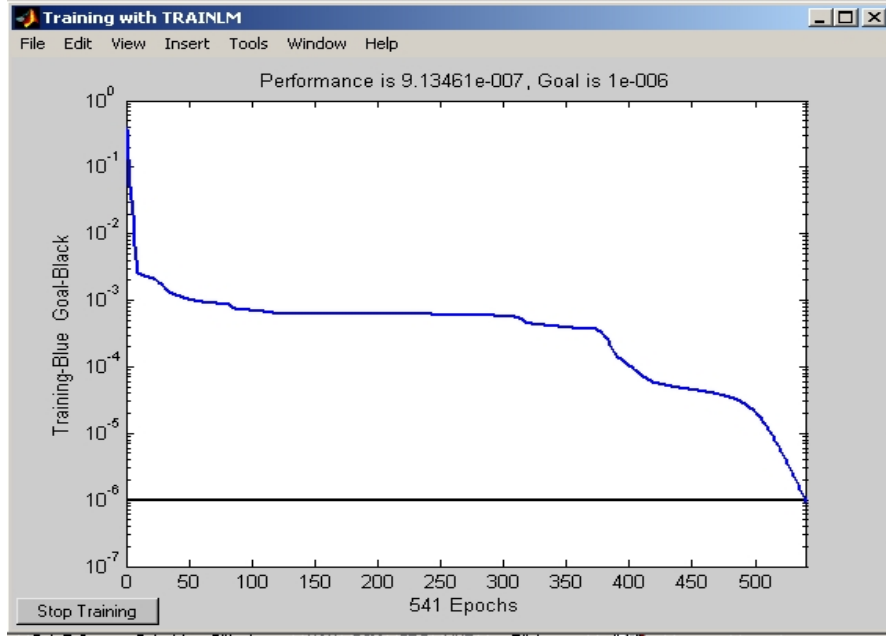
YSA için ileri beslemeli(feed forward) bir ağ yapısı oluşturulmuş ve eğitme algoritması olarak backpropagation(geri yayılım) algoritmasının bir çeşidi olan Levenberg-Marquardt algoritması seçilmiştir. Optimum sonucu elde etmek için yapılan denemelerde gizli katman sayısı, her gizli katman/katmanlardaki nöron sayısı ve her katmanın aktivasyon fonksiyonu deneme yanılma metodu kullanılarak değiştirilmiştir. Sonuç olarak, 3 nörondan oluşan 1. gizli katman, 8 nörondan oluşan 2. gizli katman ve 1 nörondan oluşan çıkış katmanı olmak üzere 3 katmanlı YSA ile en iyi sonuçların elde edildiği görülmüştür. 1. gizli katman için tan-sigmoid, 2. gizli katman için log-sigmoid ve çıkış katmanı için lineer transfer fonksiyonu seçilmiştir. Böyle bir ağ oluşturmak için kullanılan Matlab komutu aşağıda verilmiştir.

```
net=newff([3 4;0.3 0.7;1432 2864],[3,8,1],{'tansig','logsig','purelin'},'trainlm');
```

Burada “newff” komutundan sonra gelen ilk köşeli parantez içindeki değerler, giriş değerleri için belirlenen sınır değerlerdir. Burada “3 4”, birinci giriş değeri, mermer sertliği için sınır değerlerdir. Yani bizim ağımızın eğitimi için, mermer sertliği 3 ile 4 arasında değişen değerler kullanılmıştır. Aynı şekilde “0.3 0.7” ikinci giriş, yani kesme hızı için değer aralığı, “1432 2864”, üçüncü giriş yani testere dönüş hızı için değer aralığıdır.

[net,tr]=train(net,p,t);

“[net,tr]=train(net,p,t);” komutu ađın eđitimi bařlatır. Yapılan alıřmada ađın eđitimi sonucu oluřan hatanın iterasyona bađlı deđiřim grafiđi, Őekil 4.1’de gsterilmiřtir. Őekilden de grleceđi gibi 541 iterasyon sonucunda istenilen hata deđerine ulařılmıřtır.



Őekil 4.1. Hatanın iterasyona bađlı deđiřim grafiđi

4.1.3. Yapay sinir ađının test edilmesi

Eđitilen verilerin ne kadar dođru eđitildiđini grmek maksadıyla, aynı veriler tahmin iin ađa sunulmuř ve eđitilen verilerde, tahminlerin gerek deđerlere ok yakın olduđu grlmřtr. Bunun zerine eđitim ařamasında kullanılan verilere, daha nce ađa sunulmayan 8 veri daha eklenerek, 40 veri iin spesifik enerji faktrlerinin tahmininin yapılması istenmiřtir. Bunun iin yazılan Matlab kodu ařađıdaki gibidir.

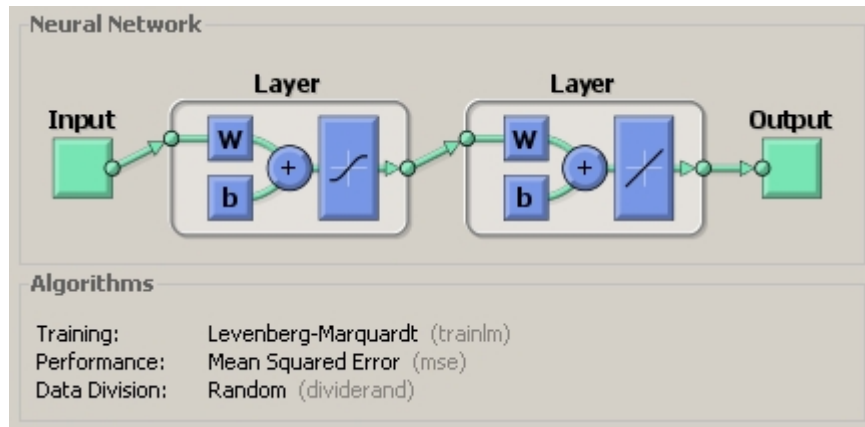
Çizelge 4.3. Deney sonucu elde edilen veriler ile tahmin sonuçlarını karşılaştıran tablo

a	t	% HATA
2,4999	2,5000	0,004000
2,1997	2,2000	0,013636
2,0636	2,0000	3,180000
1,9496	1,9500	0,020513
1,8008	1,8000	0,044444
2,6000	2,6000	0,000000
2,2001	2,2000	0,004545
2,0600	2,0700	0,483092
1,9509	1,9500	0,046154
1,7488	1,7500	0,068571
2,8499	2,8500	0,003509
2,4002	2,4000	0,008333
2,1920	2,1500	1,953488
2,0995	2,1000	0,023810
1,8502	1,8500	0,010811
3,1001	3,1000	0,003226
2,5497	2,5500	0,011765
2,2577	2,2500	0,342222
2,1002	2,1000	0,009524
1,8299	1,8300	0,005464
1,7002	1,7000	0,011765
1,4004	1,4000	0,028571
1,2489	1,2500	0,088000
1,2495	1,2400	0,766129
1,2303	1,2300	0,024390
1,8511	1,8500	0,059459
1,5469	1,5500	0,200000
1,3330	1,2600	5,793651
1,2614	1,2500	0,912000
1,2397	1,2400	0,024194
2,1987	2,2000	0,059091
1,8031	1,8000	0,172222
1,5562	1,5500	0,400000
1,4992	1,5000	0,053333
1,3001	1,3000	0,007692
2,3001	2,3000	0,004348
1,8495	1,8500	0,027027
1,5743	1,6500	4,587879
1,6001	1,6000	0,006250
1,4000	1,4000	0,000000
		0,486578

4.1.4. Spesifik Enerjinin 'nftool' ile tahmini

Mermer kesme işleminde spesifik enerjinin (SE) tahmini için kullanılacak olan YSA modeli, bir de, Matlab paket programının yeni sürümünde bulunan "nftool" aracı kullanılarak oluşturulmuştur. Aşağıda, bu şekilde uygulamanın nasıl yapıldığı anlatılmış ve elde edilen sonuçlar gösterilmiştir.

Uygulama için öncelikle kullanılacak olan giriş ve çıkış verilerinin sisteme girilmesi gerekir. Matlab workspace kullanılarak veriler girilebildiği gibi, daha önce hazırlanan dosyadan da bu veriler okutulabilir. Bu uygulamada giriş ve çıkış değerleri workspace kullanılarak girilmiştir. Giriş ve çıkış değerleri belirlendikten sonra bu verilerin yüzde kaçının eğitim, yüzde kaçının validation (onay) ve yüzde kaçının test için kullanılacağı belirlenir. Validation, ağın doğruluğunu göstermek için kullanılır. Yapılan uygulamada, verilerin %70'i eğitim için, %15'i validation (onay) için ve %15'i test işlemi için kullanılmıştır. Bu aşamadan sonra gizli katmanın kaç nörondan oluşacağı belirlenir. Bu uygulama için, çeşitli denemelerden sonra, 8 nörondan oluşan gizli katman ile en iyi sonuçlar elde edilmiştir. Çıkış katmanının kaç nörondan oluşacağı ise sistemin çıkışına göre otomatik olarak belirlenmektedir. Tüm bu işlemler tamamlandıktan sonra ağ eğitime başlanır. Bu uygulamada kullanılan ağa ait prensip şeması Şekil 4.3'te görülmektedir.

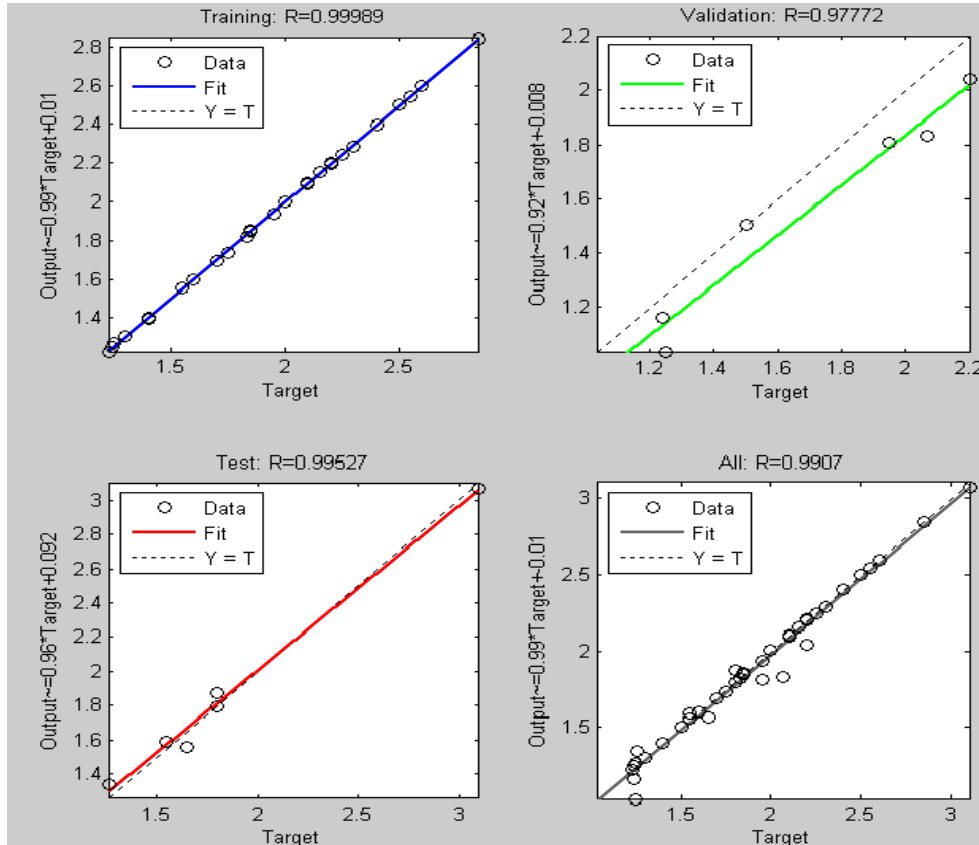


Şekil 4.3. Oluşturulan ağa ait prensip şema

Levenberg-Marquart geri yayılım fonksiyonu kullanılarak 14 iterasyonda ağın eğitimi tamamlanmış ve Şekil 4.4'te görülen sonuçlar elde edilmiştir. Eğitim safhasındaki ortalama karesel hata (MSE) $5.90537e^{-4}$, validation (onay) safhasındaki ortalama karesel hatası $5.39379e^{-3}$, test aşamasındaki ortalama karesel hatası ise $3.74251e^{-3}$ olarak elde edilmiştir.

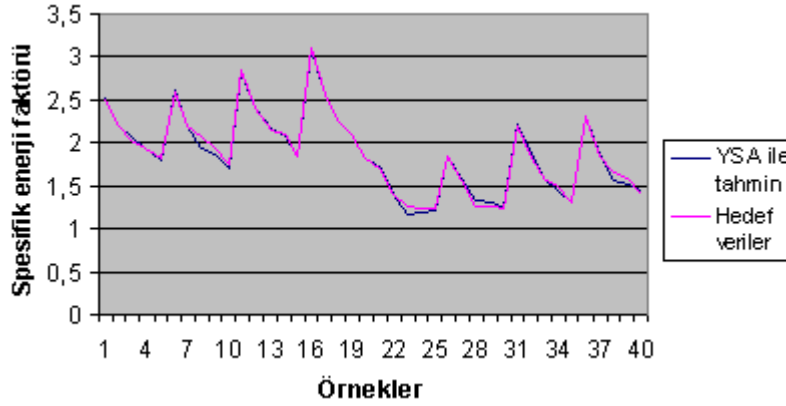
Results			
	Samples	MSE	R
Training:	28	$5.90537e^{-4}$	$9.98505e^{-1}$
Validation:	6	$5.39379e^{-3}$	$9.93222e^{-1}$
Testing:	6	$3.74251e^{-3}$	$9.95269e^{-1}$

Şekil 4.4. 14 iterasyona göre toplam MSE hata

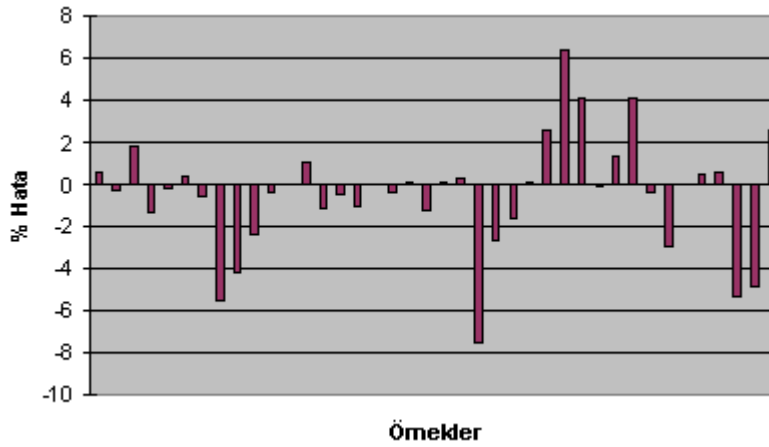


Şekil 4.5. Çıkış ve hedefin regresyon analizi

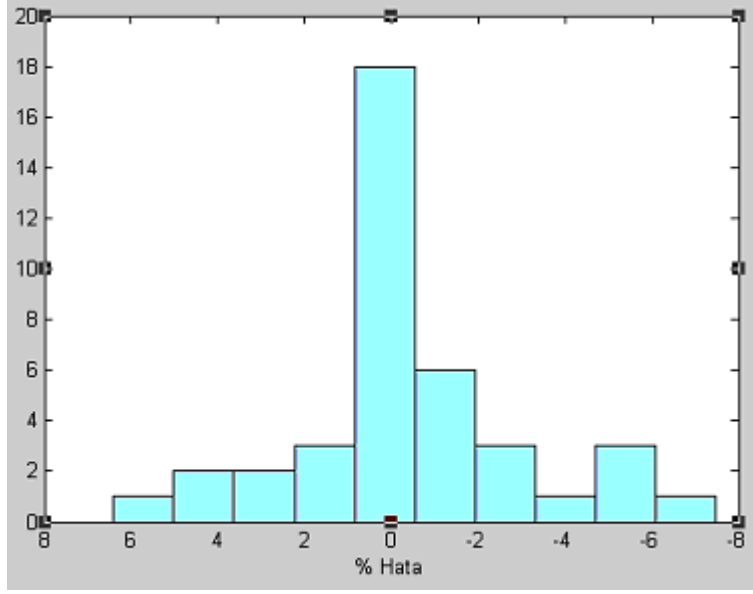
Şekil 4.5'te kesikli çizgi(Y=T eğrisi), ağın eğitiminde kullanılan verilerin regresyonudur. Düz çizgi ile gösterilen eğri (Fit) ise belirtilen aşamada, YSA'nın çıkışında elde edilen verilerin regresyonudur. Burada Fit eğrisi ile Y=T eğrisinin örtüştüğü durum, en iyi sonucu ifade eder. Yani bu durumda deney sonucu elde edilen verilerin regresyon eğrisi ile YSA'dan elde edilen verilerin regresyon eğrisi aynıdır. Şekil 4.6'da gerçek değer (hedef) ile tahmini değer (çıkış) grafik olarak karşılaştırması gösterilmektedir. Şekil 4.7 ise her bir değer tahmininde yüzde olarak ne kadar hata yapıldığı grafiksel olarak gösterilmiştir. Hatanın histogramı Şekil 4.8'de görüldüğü gibidir. Bu histogramdan da görüleceği gibi yapılan 40 tahminden 18 tanesinde yüzde sıfır hata değerine düşülebilmiştir.



Şekil 4.6. Gerçek ve tahmini değerlerin karşılaştırılması



Şekil 4.7. Her bir hata için çubuk grafiği



Şekil 4.8. Hatanın histogramı

4.2. Spesifik enerjinin dört farklı parametreye bağlı tahmini

Bu çalışmada da, deneysel verilerden elde edilen spesifik enerji faktörünü YSA metodunu kullanarak tahmin etmeyi amaçlanmıştır. Ancak bu sefer spesifik enerji, dört farklı giriş parametresine bağlı olarak tahmin edilmiştir. Mermerin sertliği, kesme hızı ve testere dönüş hızına ek olarak, bir de kullanılan motorun gücü hesaba katılmış, bu dört parametreye göre spesifik enerji tahmin edilmiştir. Ayrıca kesilen mermer sayısı da ikiden üçe çıkarılmıştır.

Tahmin yöntemlerinin uygulanmasında kullanılan tüm veriler daha önce yapılan deneysel çalışmalardan alınmıştır. Denizli traverteni, Bilecik Bej ve Uşak Yeşil mermerleri, farklı kesme hızları ve farklı testere dönüş hızları ile kesilmiş ve ölçülen aktif güçten yola çıkarak, birim hacim başına harcanan enerji miktarını ifade eden spesifik enerji (SE) hesaplanmıştır.

4.2.1. Ağın eğitimi için gerekli verilerin hazırlanması

Gerçekleştirilen çalışmada, 7,5 kW'lık asenkron motor kullanılarak, Kesme Derinliği (H_K), Kesme Hızı (V_K) ve Testere Dönüş Hızı (V_d) parametrelerinin aşağıda verilen aralıklarında, Bilecik Bej mermeri ve Denizli traverteni numuneleri üzerinde toplam 40 adet deney gerçekleştirilmiştir. Motor devir sayısı, invertör ile motor gerilimi frekansını değiştirerek ayarlanmıştır.

Kesme Derinliği (H_K) : 50mm

İlerleme Hızı (V_K) : 0.3, 0.4, 0.5, 0.6 ve 0.7m/dk

Testere Dönüş Hızı (V_d) : 1432, 1910, 2387 ve 2864d/dk

Ayrıca Uşak Yeşil mermeri üzerinde, 5,5 kW'lık asenkron motor kullanılarak, Kesme Derinliği (H_K), Kesme Hızı (V_K) ve Testere Dönüş Hızı (V_d) parametrelerinin aşağıda verilen aralıklarında, toplam 9 adet deney gerçekleştirilmiştir.

Kesme Derinliği (H_K) : 50mm

İlerleme Hızı (V_K) : 0.5, 0.6 ve 0.7m/dk

Testere Dönüş Hızı (V_d) : 2387, 2626 ve 2864d/dk

Görüldüğü gibi makine parametrelerinden Kesme Derinliği (H_K) 50mm değerinde sabit tutulmaktadır. Kesme hızı (V_K) 0.3-0.7m/dk aralığında 5 değer almaktadır. Testere dönüş hızı (V_d) ise 1432-2864d/dk aralığında 5 değer almaktadır. Motorun gücü ise 5,5 ve 7,5 kW olmak üzere iki farklı değer almaktadır. Yapay sinir ağının eğitiminde verilerin birbirine yakınlığının eğitimi kolaylaştıracağı düşünüldüğünden, mevcut veriler normalize edildikten sonra ağa sunulmuştur. Bu amaçla, kesme hızı parametreleri 10 ile çarpılmış ve testere dönüş hızı parametreleri ise 100'e bölünmüştür.

Alınan veriler Matlab'ta ağın eğitim için kullanılmak üzere matris formuna çevrilmiştir. Mevcut 49 adet verinin 39 adedi ağın eğitimi için kullanılmış ve 10 adet veri test işlemi için ayrılmıştır. Ağın eğitimi için kullanılan veriler, Çizelge 4.4'te görülmektedir.

Çizelge 4.4. YSA'nın eğitimi için kullanılan veriler

Motor Gücü(kW)	Mermer Sertlik Derecesi	Kesme hızı[m/dk]	Testere dönüş hızı[d/dk]	SE
7.5	4	3	14.32	2.500
7.5	4	4	14.32	2.200
7.5	4	6	14.32	1.950
7.5	4	7	14.32	1.800
7.5	4	4	19.10	2.200
7.5	4	5	19.10	2.070
7.5	4	6	19.10	1.950
7.5	4	7	19.10	1.750
7.5	4	3	23.87	2.850
7.5	4	4	23.87	2.400
7.5	4	5	23.87	2.150
7.5	4	7	23.87	1.850
7.5	4	3	28.64	3.100
7.5	4	4	28.64	2.550
7.5	4	5	28.64	2.250
7.5	4	6	28.64	2.100
7.5	3	3	14.32	1.700
7.5	3	4	14.32	1.400
7.5	3	5	14.32	1.250
7.5	3	6	14.32	1.250
7.5	3	7	14.32	1.230
7.5	3	4	19.10	1.550
7.5	3	5	19.10	1.260
7.5	3	6	19.10	1.260
7.5	3	7	19.10	1.240
7.5	3	3	23.87	2.200
7.5	3	4	23.87	1.800
7.5	3	6	23.87	1.500
7.5	3	7	23.87	1.300
7.5	3	3	28.64	2.300
7.5	3	5	28.64	1.650
7.5	3	6	28.64	1.600
7.5	3	7	28.64	1.400
5.5	4	5	23.87	2.702
5.5	4	5	26.26	2.760
5.5	4	6	26.26	2.548
5.5	4	6	28.64	2.711
5.5	4	7	23.87	2.365
5.5	4	7	26.26	2.465

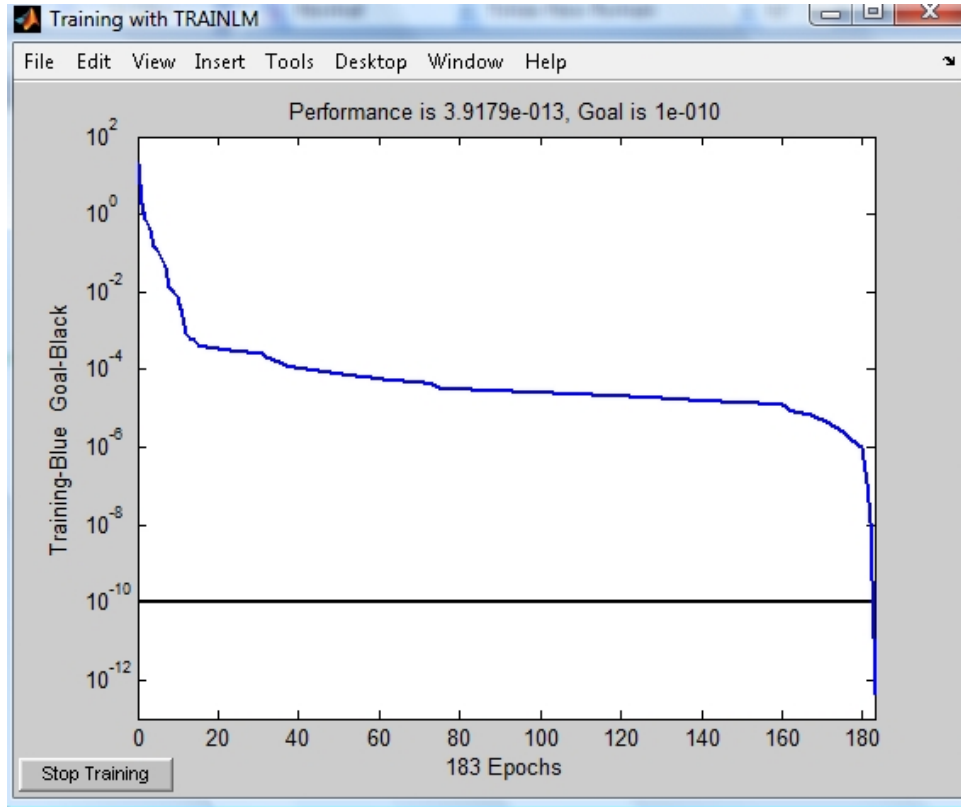
4.2.2. Yapay sinir ağının eğitilmesi

YSA için ileri beslemeli(feed forward) bir ağ yapısı oluşturulmuş ve eğitime algoritması olarak backpropagation(geri yayılım) algoritmasının bir çeşidi olan Levenberg-Marquardt algoritması seçilmiştir. Optimum sonucu elde etmek için yapılan denemelerde gizli katman sayısı ve her gizli katman/katmanlardaki nöron sayısı, her katmanın aktivasyon fonksiyonu deneme yanılma metodu kullanılarak değiştirilmiştir. Sonuç olarak, 4 nörondan oluşan birinci gizli katman, 8 nörondan oluşan ikinci gizli katman ve 1 nörondan oluşan çıkış katmanı olmak üzere 3 katmanlı YSA ile en iyi sonuçların elde edildiği görülmüştür. Birinci gizli katman için tan-sigmoid, ikinci gizli katman için log-sigmoid ve çıkış katmanı için lineer transfer fonksiyonu seçilmiştir. Böyle bir ağ oluşturmak için kullanılan Matlab komutu aşağıda verilmiştir.

```
net=newff([5.5 7.5;3 4;3 7;14.32 28.64],[4,8,1],{'tansig','logsig','purelin'},'trainlm');
```

Burada “newff” komutundan sonra gelen ilk köşeli parantez içindeki değerler, giriş değerleri için belirlenen değer aralıklarıdır. Burada “5.5 7.5” birinci giriş değer aralığı, motor gücü için verilen değer aralığıdır. “3 4”, ikinci giriş değeri, mermer sertliği için değer aralığıdır. Yani bizim ağımızın eğitimi için, mermer sertliği 3 ile 4 arasında değişen değerler kullanılmıştır. Aynı şekilde “3 7” üçüncü giriş, yani kesme hızı için değer aralığı, “14.32 28.64”, dördüncü giriş yani testere dönüş hızı için değer aralığıdır.

“newff” komutundan sonra gelen ikinci köşeli parantez ise gizli katmanlar ve çıkışın kaç nörondan oluştuğunu belirtir. “[4,8,1]”, 4 nörondan oluşan 1. gizli katman, 8 nörondan oluşan 2. gizli katman ve 1 nörondan oluşan çıkış katmanı ile bir ağ oluşturacağımızı belirtir. “{'tansig','logsig','purelin'}”, sırasıyla 1. gizli katman, 2. gizli katman ve çıkış için transfer fonksiyonlarını belirtir. ‘trainlm’ ise öğrenme algoritması olarak, geriye yayılım algoritmasının bir türü olan, Levenberg-Marquardt algoritmasını seçtiğimizi gösterir.



Şekil 4.9. Hatanın iterasyona bağlı değişim grafiği

4.2.3. Yapay sinir ağının test edilmesi

Eğitilen verilerin ne kadar doğru eğitildiğini görmek amacıyla, aynı veriler tahmin için ağa sunulmuş ve eğitilen verilerde, tahminlerin gerçek değerlere çok yakın olduğu görülmüştür. Bunun üzerine eğitim aşamasında kullanılan verilere, daha önce ağa sunulmayan 10 veri daha eklenerek, 49 veri için spesifik enerji faktörlerinin tahmininin yapılması istenmiştir. Bunun için yazılan Matlab kodu aşağıdaki gibidir.

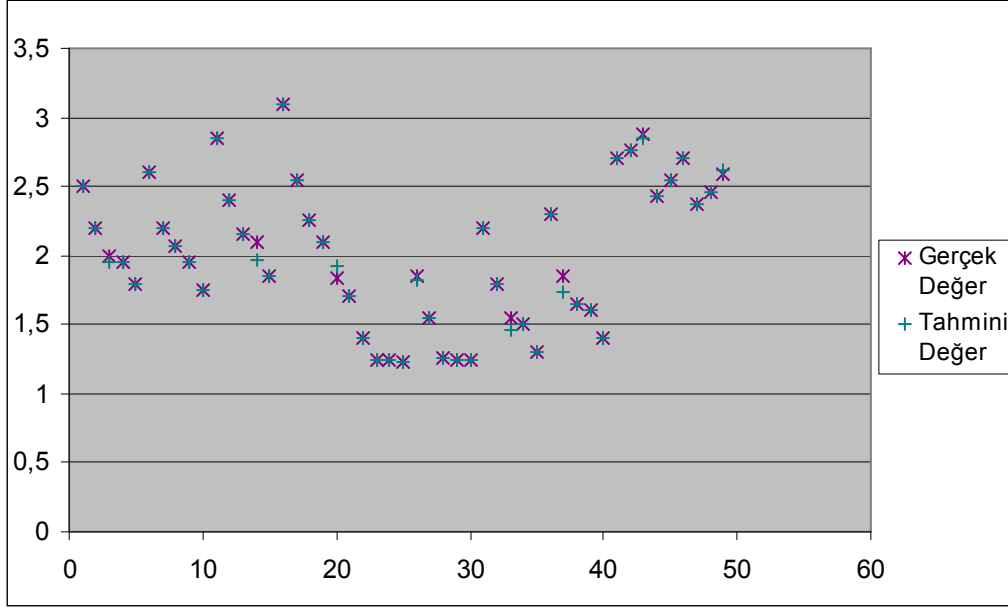
```
p=[...];  
a=sim(net,p)
```

Simülasyon sonucu elde edilen veriler, Çizelge 4.5'te verilmiştir.

Çizelge 4.5. Deney sonucu elde edilen veriler ile tahmin sonuçlarının karşılaştırılması

motor gücü (kW)	mermer sertliği	kesme hızı	testere dönüş hızı	SE	YSA ile yapılan tahmin	%Hata
7,5	4	3	14,32	2,5	2,5	0,00
7,5	4	4	14,32	2,2	2,2	0,00
7,5	4	5	14,32	2	1,95	2,50
7,5	4	6	14,32	1,95	1,95	0,00
7,5	4	7	14,32	1,8	1,8	0,00
7,5	4	3	19,10	2,6	2,61	0,26
7,5	4	4	19,10	2,2	2,2	0,00
7,5	4	5	19,10	2,07	2,07	0,00
7,5	4	6	19,10	1,95	1,95	0,00
7,5	4	7	19,10	1,75	1,75	0,00
7,5	4	3	23,87	2,85	2,85	0,00
7,5	4	4	23,87	2,4	2,4	0,00
7,5	4	5	23,87	2,15	2,15	0,00
7,5	4	6	23,87	2,1	1,97	6,40
7,5	4	7	23,87	1,85	1,85	0,00
7,5	4	3	28,64	3,1	3,1	0,00
7,5	4	4	28,64	2,55	2,55	0,00
7,5	4	5	28,64	2,25	2,25	0,00
7,5	4	6	28,64	2,10	2,10	0,00
7,5	4	7	28,64	1,83	1,93	5,43
7,5	3	3	14,32	1,70	1,70	0,00
7,5	3	4	14,32	1,40	1,40	0,00
7,5	3	5	14,32	1,25	1,25	0,00
7,5	3	6	14,32	1,25	1,25	0,00
7,5	3	7	14,32	1,23	1,23	0,00
7,5	3	3	19,10	1,85	1,82	1,65
7,5	3	4	19,10	1,55	1,55	0,00
7,5	3	5	19,10	1,26	1,26	0,00
7,5	3	6	19,10	1,25	1,25	0,00
7,5	3	7	19,10	1,24	1,24	0,00
7,5	3	3	23,87	2,20	2,20	0,00
7,5	3	4	23,87	1,80	1,80	0,00
7,5	3	5	23,87	1,55	1,47	5,46
7,5	3	6	23,87	1,50	1,50	0,00
7,5	3	7	23,87	1,30	1,30	0,00
7,5	3	3	28,64	2,30	2,30	0,00
7,5	3	4	28,64	1,85	1,74	6,16
7,5	3	5	28,64	1,65	1,65	0,00
7,5	3	6	28,64	1,60	1,60	0,00
7,5	3	7	28,64	1,40	1,40	0,00
5,5	4	5	23,87	2,70	2,70	0,00
5,5	4	5	26,26	2,76	2,76	0,00
5,5	4	5	28,64	2,88	2,85	1,09
5,5	4	6	23,87	2,43	2,44	0,40
5,5	4	6	26,26	2,55	2,55	0,00
5,5	4	6	28,64	2,71	2,71	0,00
5,5	4	7	23,87	2,37	2,37	0,00
5,5	4	7	26,26	2,47	2,47	0,00
5,5	4	7	28,64	2,59	2,61	0,90

“%HATA”, hatanın gerçek değere oranının yüzde olarak karşılığını göstermektedir. Tablodan görüldüğü gibi kabuledilebilir hata seviyesine düşülebilmektedir. Yapay sınır ağından alınan değerler, deney sonucu elde edilen değerlere istenilen ölçüde yaklaşmıştır. Tahmin edilen değerler ile deney sonucu elde edilen değerler arasındaki farkı gösteren grafik, Şekil 4.10’da gösterilmiştir.



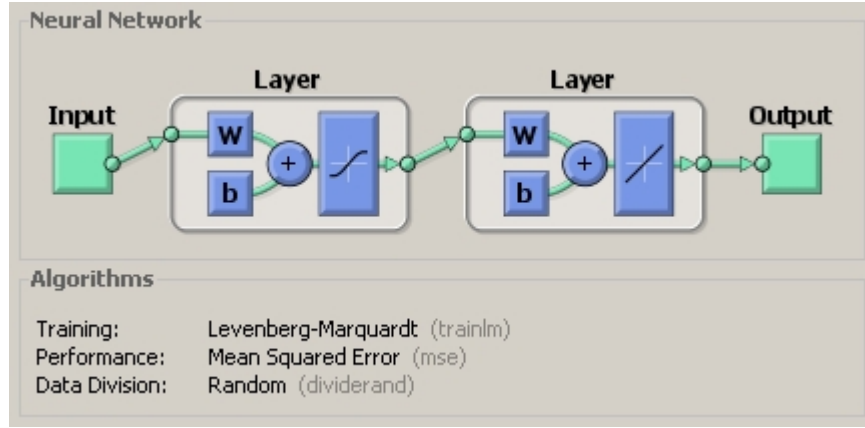
Şekil 4.10. Deneyler sonucu elde edilen değer ile tahmini değer arasındaki farkı gösteren grafik

4.2.4. Spesifik Enerjinin ‘nftool’ ile tahmini

Mermer kesme işleminde spesifik enerjinin(SE), dört farklı parametreye bağlı olarak tahmini için kullanılacak olan YSA modeli, bir de, Matlab paket programının 7.1 sürümünde bulunan “nftool” aracı kullanılarak oluşturulmuştur. Aşağıda, bu şekilde uygulamanın nasıl yapıldığı anlatılmış ve elde edilen sonuçlar gösterilmiştir.

Uygulama için öncelikle kullanılacak olan giriş ve çıkış verilerinin sisteme girilmesi gerekir. Matlab workspace kullanılarak veriler girilebileceği gibi, daha önce hazırlanan

dosyadan da bu veriler okutulabilir. Bu uygulamada giriş ve çıkış değerleri workspace kullanılarak girilmiştir. Giriş ve çıkış değerleri belirlendikten sonra bu verilerin yüzde kaçının eğitim için, yüzde kaçının validation (onay) için ve yüzde kaçının test için kullanılacağı belirlenir. Validation, ağın doğruluğunu göstermek için kullanılır. Yapılan uygulamada, verilerin %70'i eğitim için, %15'i validation (onay) için ve %15'i test işlemi için kullanılmıştır. Bu aşamadan sonra gizli katmanın kaç nörondan oluşacağı belirlenir. Bu uygulama için, çeşitli denemelerden sonra, 9 nörondan oluşan gizli katman ile en iyi sonuçlar elde edilmiştir. Çıkış katmanının kaç nörondan oluşacağı ise sistemin çıkışına göre otomatik olarak belirlenmektedir. Tüm bu işlemler tamamlandıktan sonra ağ eğitimi başlanır. Bu uygulamada kullanılan ağa ait prensip şeması Şekil 4.11'de görülmektedir.



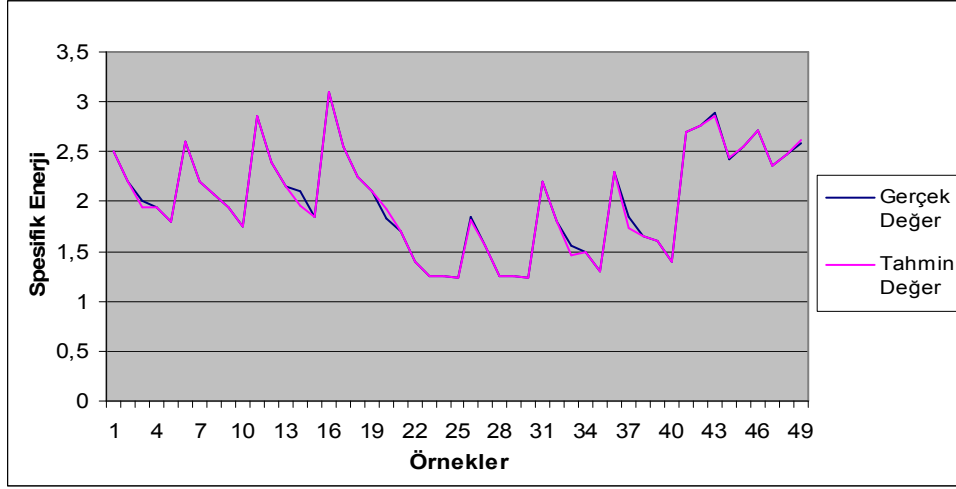
Şekil 4.11. Oluşturulan ağa ait prensip şema

Levenberg-Marquart geri yayılım fonksiyonu kullanılarak 166 iterasyonda ağın eğitimi tamamlanmış ve Şekil 4.12'de görülen sonuçlar elde edilmiştir. Eğitim safhasının ortalama karesel hatası (MSE) $1.32789e^{-4}$, validation (onay) safhasının ortalama karesel hatası $2.78264e^{-3}$, test aşamasının ortalama karesel hatası ise $2.19604e^{-3}$ olarak elde edilmiştir.

Şekil 4.13'te gerçek değer (hedef) ile tahmini değer (çıkış) grafik olarak karşılaştırması gösterilmektedir.

Results			
	Samples	MSE	R
Training:	35	1.32789e-4	9.99696e-1
Validation:	7	2.78264e-3	9.97660e-1
Testing:	7	2.19604e-3	9.97829e-1

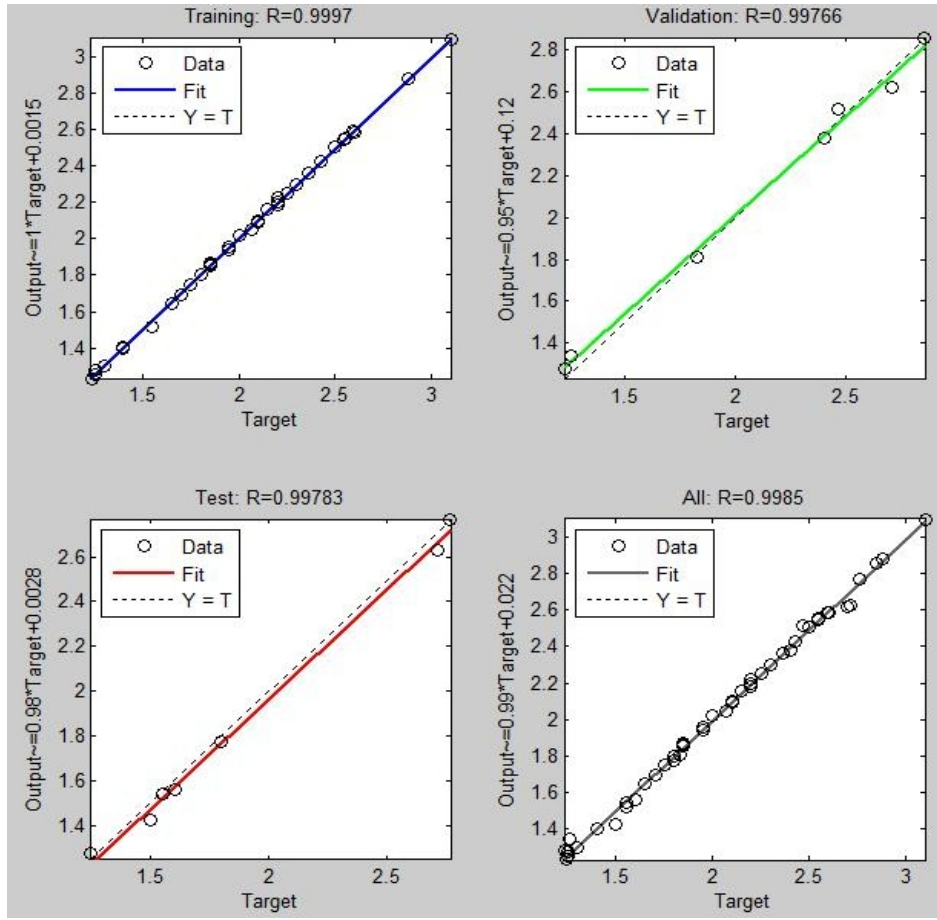
Şekil 4.12. 166 iterasyona göre toplam MSE hata



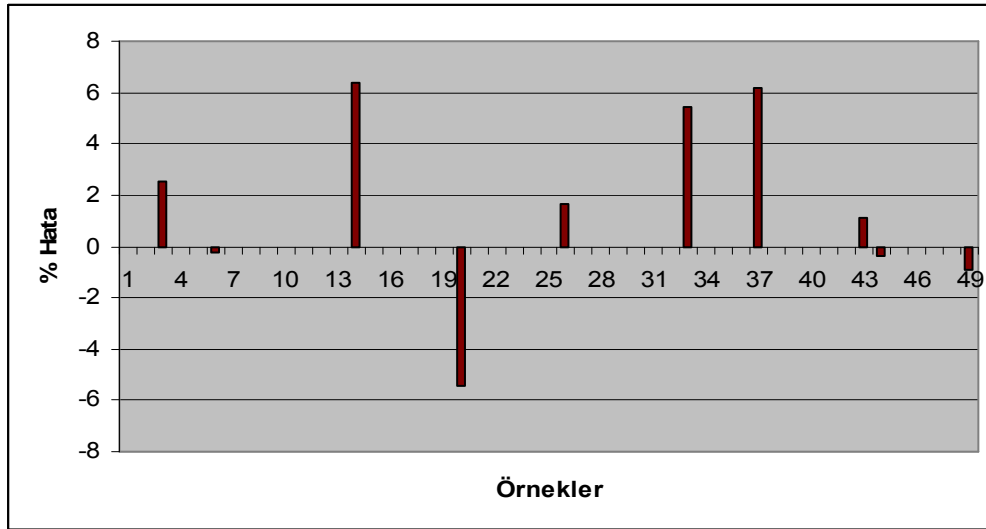
Şekil 4.13. Gerçek ve tahmini değerlerin karşılaştırılması

Şekil 4.14'te kesikli çizgi($Y=T$ eğrisi), ağız eğitiminde kullanılan verilerin regresyonudur. Düz çizgi ile gösterilen eğri (Fit) ise belirtilen aşamada, YSA'nın çıkışında elde edilen verilerin regresyonudur. Burada Fit eğrisi ile $Y=T$ eğrisinin örtüştüğü durum, en iyi sonucu ifade eder. Yani bu durumda deney sonucu elde edilen verilerin regresyon eğrisi ile YSA'dan elde edilen verilerin regresyon eğrisi aynıdır.

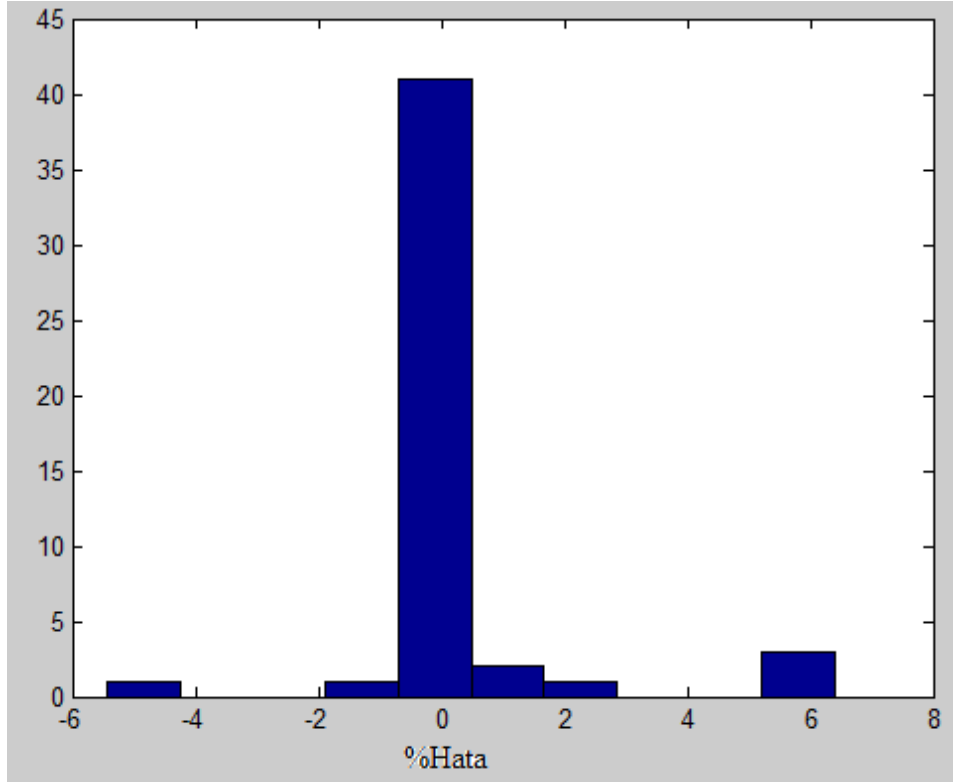
Şekil 4.15, her bir değer tahmininde yüzde olarak ne kadar hata yapıldığı grafiksel olarak gösterilmiştir. Hatanın histogramı Şekil 4.16'da görüldüğü gibidir. Bu histogramdan da görüleceği gibi yapılan 49 tahminden 41 tanesinde yüzde sıfır hata değerine düşülebilmektedir.



Şekil 4.14. Çıkış ve hedefin regresyon analizi



Şekil 4.15. Her bir hata için çubuk grafiği



Şekil 4.16. Hatanın histogramı

5. SONUÇ VE DEĞERLENDİRME

Yapılan çalışmada, yapay sinir ağı ile iki farklı uygulama gerçekleştirilmiştir. Her iki çalışmada da yapay sinir ağı ile tahmin uygulaması yapılmıştır. Birinci çalışmada, mermerin sertliği, kesme hızı ve testere dönüş hızı verilerine göre mermer kesme işleminde enerji optimizasyonunda önemli bir parametre olan spesifik enerjinin tahmin edilmesi için YSA metodu kullanılmıştır. Yapılan tahmin işlemlerinin doğruluğu daha önce yapılan deney sonuçları ile test edilmiştir. Tahmin işlemi sonucunda % 0,4865'lik bağıl hata ortalaması elde edilmiştir. Bu oldukça iyi bir değerdir ve bu YSA kullanılarak optimum kesme parametrelerinin elde edilebileceğini göstermektedir. Kesme parametrelerinin optimum değerlerinin belirlenebilmesi ise tüketilen enerjinin de optimize edilebileceği anlamına gelmektedir. YSA kullanılarak optimum parametrelerin elde edilmesiyle, kesme deneylerine gerek kalmayacak, bu da hem zaman, hem hammadde, hem de enerji tasarrufu sağlayacaktır.

İkinci çalışmada ise, yine spesifik enerji değeri tahmin edilmiş ancak bu defa motorun gücü de hesaba katılmıştır. Yani dört farklı giriş değerine karşılık gelen spesifik enerji değeri tahmin edilmiştir. Bu uygulamada ayrıca, verilerin birbirlerine yakın değerlerde olmasının, yapay sinir ağının eğitimini kolaylaştıracağı düşünüldüğünden, bazı parametrelerde normalizasyon yapılmıştır. Bu sebeple, giriş değerlerinden kesme hızı parametreleri on ile çarpılmış ve testere dönüş hızı parametreleri yüze bölünmüştür. Bu işlemin ardından verilerin daha kolay ve daha iyi eğitilebildiği görülmüştür.

Yapılan her iki çalışmada da görülmüştür ki, farklı ağ yapıları ve özellikler kullanılarak, ağ eğitme başarısı değiştirilebilmektedir. Ağın hata oranı optimum bir düzeye indirilmelidir. Zira eğitim hatasının çok küçük değerlere düşmesi, yani ağın çok fazla iyi eğitilmesinin de anormal hatalara neden olabileceği gözlenmiştir. Birinci uygulamada 10^{-6} ortalama karesel hatada en iyi sonuçlar elde edilirken, 10^{-8} hata değerinde, bazı tahminlerde çok büyük hatalar ortaya çıkmıştır. Aynı şekilde ikinci uygulamada, 10^{-10} ortalama karesel hata değerinde optimum sonuçlar elde edilmiştir. Yapılan çalışmalar sırasında, ağ 10^{-15} ortalama

karesel hata deęerine kadar eęitilebilmiř, fakat bu durumda bazı tahminlerde anormal sonular gzlenmiřtir.

Her iki alıřmada da ęrenme algoritması olarak, geri yayılım algoritmasının bir tr olan Levenber-Marquardt algoritması kullanılmıřtır. Yapılan literatr alıřmasında bu algoritmanın YSA'yı emsallerine gre daha hızlı eęittięi grlmřtr. Yapılan alıřma ile bunun doęruluęu anlařılmıřtır.

Yapılan alıřmalarda, iki farklı motor ve  farklı mermer ile yapılan kesimlerde hesaplanan spesifik enerji deęerleri tahmin edilmiřtir. Toplam 49 deney verisinin 10 tanesinin test iin ayrıldıęı dřnldęnde, sadece 39 adet deney verisi ile aę eęitilmiřtir. Bu sayının artması yapay sinir aęının daha iyi eęitilmesini saęlayacaktır. Daha fazla sayıda motor ve mermer kullanılarak daha iyi sonular elde edilebilir. Ayrıca spesifik enerji deęerine etki edebilecek motor devri, testere zellikleri gibi bazı faktrleri de hesaba katmak, aęı daha iyi eęitebilmek ve daha doęru tahminler yapabilmek aısından faydalı olacaktır.

6. KAYNAKLAR

Bağcıvandemir, M., 1997, “Mermer Oyma Makinası Tasarımı ve Kesme Hareketlerinin Invertörle Kontrolü”, Yüksek Lisans Tezi, Afyon Kocatepe Üniversitesi, Fen Bilimleri Enstitüsü, Afyon

Bolat, B., Kalenderli, Ö., “Levenberg-Marquart Algoritması Kullanılan ile Elektrot Biçim Optimizasyonu”, International XII. Turkish Symposium on Artificial Intelligence and Neural Networks – TAINN 2003

Buyuksagis, I. S. and Goktan, R. M., 2005, “Investigation of marble machining performance using an instrumented block-cutter”, Journal of Materials Processing Technology, Nov., Vol.169, pp.258-262.

Büyüksagiş, I. S., 1998, “ Dairesel Testereli Blok Kesme Makinalarında Mermerlerin Kesilebilirlik Analizleri ”, Doktora tezi, Osmangazi Üniversitesi, Fen Bilimleri Enstitüsü, Eskişehir.

Chen W., 2000, “Cutting forces and surface finish when machining medium hardness steel using CBN tools”, International Journal of Machine Tools & Manufacture, Now., Vol.40, pp. 455-466.

Çınar S.M. , 2007, “ Mermer Kesme Makinelerinde Enerji Tüketimi Optimizasyonu”, Yüksek Lisans Tezi, Afyon Kocatepe Üniversitesi, Fen Bilimleri Enstitüsü, Afyon

Demuth, H., Beale, M., Hagan, M., "Neural Network Toolbox, for use with MATLAB", "User Guide Version 5", Mathworks, 2006.

Dursun M. , 2009, “ Mermer Kesme Makinelerinde Invertör Parametreleri Deđiřtirilerek Elektrik Enerji Optimizasyonu”, Yüksek Lisans Tezi, Afyon Kocatepe Üniversitesi, Fen Bilimleri Enstitüsü, Afyon

Elmas, Ç., 2003, “Yapay Sinir Ağları (Kuram,Mimari, Uygulama)”, Seçkin Yayınları

Haykin, S., “Neural Networks: a Comprehensive Foundation”, Upper Saddle River, N.J., Prentice Hall, 1999.

Hopfield, J.J., 1982, “Neural Networks and Physical Systems with Emergent Collective Computational Abilities”, In Proc. of National Academy of Sciences, 2554-2558

Kalogirou, S.A., (1999), “Applications of artificial neural networks in energy systems A review, Energy Conversion & Management”, 40, 1073-1087.

Öztemel, E., 1993, “Yapay Sinir Ağları”, Papatya Yayıncılık

Polini, W. and Turchetta S., 2004, “Force and specific energy in stone cutting by diamond mill”, International Journal of Machine Tools and Manufacture, Sept., Vol.44, pp.1189-1196.

Rummelhart, D.E., Hinton, G.E., Williams, R.J., 1986, “Learning Internal Representation by Error Propagation in Parallel Distributed Processing”, Mit Press, Cambridge MA.

Sađırođlu, S., Beřdok, E., Eler, M., 2003, “Mühendislikte Yapay Zeka Uygulamaları, Yapay Sinir Ağları”, Ufuk Yayınları

Tatar, A., 2007, “İletim Sistemlerinde Yapay Sinir Ağları ile Mesafe Koruması Uygulamaları”, Yüksek Lisans Tezi, Kahramanmaraş Sütçü İmam Üniversitesi, Fen Bilimleri Enstitüsü, Kahramanmaraş

Tobias, S. A., 1965, “Machine Tool Vibration”, Blackie & Son Ltd., London

Tuzcuoğlu, H., 2003, “Yapay Zeka Teknikleri, Depremde Kullanılması ve Küme Kuramları”, Dokuz Eylül Üniversitesi Fen ve Mühendislik Dergisi, cilt 5, sayı 1, sh:73-88, İzmir

Üstün, S. V., 1997, “Yapay Sinir Ağları Kullanılarak Türkçe'deki Sesli Harflerin Tanınması”, Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul

Xu, X. P., Li, Y. Malkin, S., 2001, “Forces and Energy in Circular Sawing and Grinding of Granite”, Journal of Manufacturing Science and Engineering, Feb., Vol.123, pp.13-22.

6.1. İnternet Kaynakları

	Erişim Tarihi
1 http://www.actmakine.com	17.07.2006
2 http://www.mks.com.tr/blokkesme.php	08.04.2008

7. ÖZGEÇMİŞ

Adı Soyadı	Emre AKARSLAN
Doğum Yeri	Bolvadin
Doğum Tarihi	22.09.1980
Medeni Hali	Bekar
Yabancı Dili	İngilizce

Eğitim Durumu (Kurum ve Yıl)

Lise	Bolvadin Yabancı Dil Ağırlıklı Lise
Lisans	Selçuk Üniversitesi Elektrik-Elektronik Mühendisliği

Çalıştığı Kurum/Kurumlar ve Yıl aralığı

Anacom Bilgisayar, Konya, 2002-2003
AKU Yapı İşleri Teknik Daire Başkanlığı, 2006