

**T.C.
SAKARYA ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ**

**SÜREKLİ TAVLAMA HATLARINDA ENERJİ
GİDERİNİN KALİTE VE BOYUT DEĞERLERİNE GÖRE
OPTİMİZE EDİLMESİ VE GEÇİŞLERDE
OPERATÖR DAVRANIŞLARININ MODELLENMESİ**

DOKTORA TEZİ

Ömer Faruk SEZER

**Enstitü Anabilim Dalı : İşletme
Enstitü Bilim Dalı : Üretim Yönetimi ve Pazarlama**

Tez Danışmanı: Prof. Dr. Erman COŞKUN

TEMMUZ – 2017

T.C.
SAKARYA ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ

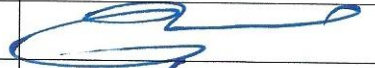
**SÜREKLİ TAVLAMA HATLARINDA ENERJİ
GİDERİNİN KALİTE VE BOYUT DEĞERLERİNE GÖRE
OPTİMİZE EDİLMESİ VE GEÇİŞLERDE OPERATÖR
DAVRANIŞLARININ MODELLENMESİ**

DOKTORA TEZİ

Ömer Faruk SEZER

Enstitü Anabilim Dalı : İşletme
Enstitü Bilim Dalı : Üretim Yönetimi ve Pazarlama

“Bu tez 19.1.2017 tarihinde aşağıdaki jüri tarafından Oybirliği / Oyçokluğu ile kabul edilmiştir.”

JÜRİ ÜYESİ	KANAATI	İMZA
Prof. Dr. Ermen Çoşkun	Bazırılı	
Doç. Dr. Yılmaz Gökyay	Bazırılı	
Yrd. Doç. Dr. Ender GÜRGEN	Bazırılı	
Yrd. Doç. Dr. Nevin Akbikeli	Bazırılı	
Yrd. Doç. Dr. H. İbrahim ÇEBECİ	Bazırılı	



SAKARYA
ÜNİVERSİTESİ

SOSYAL BİLİMLER ENSTİTÜSÜ
YÜKSEK LİSANS/DOKTORA İNTİHAL YAZILIM RAPORU BEYAN BELGESİ

Tez Başlığı:

Sürekli Tavlama Hatlarında enerji giderinin kalite ve boyut değerlerine göre optimize edilmesi ve geçişlerde operatör davranışlarının modellenmesi

Yukarıda başlığı belirtilen tez çalışmamın toplam **381** sayfalık kısmına ilişkin *Sakarya Üniversitesi Lisansüstü Yönetmeliği Madde 28* uyarınca aşağıda belirtilen filtrelemeler uygulanarak alınmış olan ve **15 / 06 / 2017** tarihinde Sosyal Bilimler Enstitüsü tarafından şahsıma iletilen *Turnitin* intihal tespit programı raporuna göre tezimin benzerlik oranı **% 10**'dur.

Uygulanan filtrelemeler:

- 1) Kaynakça hariç
- 2) Alıntılar dahil
- 3) 5 kelimeden daha az örtüşme içeren metin kısımları hariç

Bu bilgiler doğrultusunda tez çalışmamın herhangi bir intihal içermediğini; aksinin tespit edileceği muhtemel durumda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve yukarıda vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Gereğini saygılarımla arz ederim.

Öğrenci

Ömer Faruk Sezer

19.7.2017

ÖĞRENCİ BİLGİLERİ

Adı – Soyadı
Öğrenci Numarası
Ana Bilim Dalı
Programı
Statüsü

: Ömer Faruk Sezer
: 1160D04006
: İşletme
: Üretim Yönetimi ve Pazarlama
: ☐ Y. Lisans ☒ Doktora ☐ Bütünleşik Doktora

19.7.2017

Danışman
Prof. Dr. Erman Coşkun

ÖNSÖZ

Bu tez çalışması, kendi bilgi ve çalışmalarımın yanı sıra bugüne kadar beni eğiten değerli hocalarımla ve çalışma arkadaşlarımla yardımları ve destekleri sayesinde olmuştur. Bu nedenle kendilerine çok teşekkür ediyorum. Sağ kalanlara uzun ömür, ahirete göçenlere Allah'tan rahmet diliyorum.

Sadece bu tez çalışmasının hazırlanması sırasında değil, lisansüstü eğitimimden başlayarak bilgi ve tecrübelerini benimle paylaşan, destek olan ve bana yol gösteren, hem tezime hem de hayatıma dair yeni açılımlar kazandıran danışman hocam Prof. Dr. Erman Coşkun'a şükranlarımı sunuyorum.

Ayrıca tez izleme sürecinde ve savunması sırasında değerli fikirlerini paylaşarak çalışmanın daha nitelikli olmasına yardımcı olan tez izleme komitesi üyelerim Yrd. Doç. Dr. Halil İbrahim Cebeci ve Yrd. Doç. Dr. Nevra Akbilek hocalarıma ve tez savunma komitesi üyelerim Doç. Dr. Yılmaz Gökşen ve Yrd. Doç. Dr. Ender Gürgen hocalarıma katkılardan dolayı çok teşekkür ederim.

Bugünlere gelmemde büyük emekleri olan, maddi ve manevi desteklerini hiçbir zaman esirgemeyen, baştan rahmetli annem Akgül Sezer ve rahmetli babam Gültekin Sezer olmak üzere ablam Sezan Anıl'a, kardeşim Haluk Sezer'e ve tüm aile üyelerime teşekkür ediyorum.

Son olarak, tez yazım sürecinin tüm zorluklarına benimle beraber göğüs geren, uzun çalışma saatlerimi anlayışla karşılayan ve nihayetinde bu tezin meydana gelmesinde önemli bir pay sahibi olan sevgili eşim Gül Banu Sezer ve güzel kızım Meltem Sezer'e çok teşekkür ediyorum. İyi ki varsınız.

Bu tezi, bugüne kadar benim yanımda olan ve hayatıma renk katan tüm dostlarıma ithaf ediyorum.

İÇİNDEKİLER

KISALTMALAR.....	iii
TABLO LİSTESİ.....	v
ŞEKİL LİSTESİ	vii
ÖZET	x
SUMMARY	xi
GİRİŞ	1
BÖLÜM 1: KAVRAMSAL ALTYAPI	13
1.1 Çok Hedefli Karar Verme Teknikleri ve Doğrusal Olmayan Programlama	13
1.1.1 Karar Verme	13
1.1.2 Matematiksel Modelleme ve Yönetim Bilimi	14
1.1.2.1 Matematiksel Modelleme Türleri	15
1.1.2.1.1 Ragsdale Modeli	16
1.1.2.1.2 Winston Modeli	17
1.1.3 Problem Çözme Süreci	18
1.1.4 Doğrusal ve Doğrusal Olmayan Programlama	19
1.1.5 Doğrusal Olmayan Programlamanın Çözüm Yöntemleri.....	21
1.1.6 Çok Hedefli Programlama	30
1.1.6.1 Öncelikli Hedef Programlama	35
1.1.6.2 Ağırlıklı Hedef Programlama	36
1.1.6.3 Minmaks Hedef Programlama.....	37
1.2 Yapay Sinir Ağları.....	38
1.2.1 Yapay Sinir Ağlarının Genel Yapısı.....	41
1.2.1.1 Biyolojik Sinir Hücreleri	42
1.2.1.2 Yapay Sinir Ağları Genel Yapısı.....	44
1.2.1.3 Yapay Sinir Ağlarının Elemanları	46
1.2.1.4 Yapay Sinir Ağlarının Yapısı	50
1.2.1.5 Yapay Sinir Ağlarının Türleri.....	51
1.2.2 Yapay Sinir Ağlarının Avantajları ve dezavantajları.....	52
1.2.3 Yapay Sinir Ağlarının Uygulama Alanları	55
1.2.4 İleri Beslemeli Yapay Sinir Ağları	56

1.2.5	Yapay Sinir Ağları Temel Öğrenme Algoritmaları	62
1.2.6	Yapay Sinir Ağlarının Eğitilmesi	66
1.3	Çelik Genel Yapısı, Alaşımlama ve Tavlama.....	69
1.3.1	Çelik Genel Yapısı ve Alaşımlama.....	70
1.3.2	Tavlama	78
BÖLÜM 2: SÜREKLİ TAVLAMA HATLARI TANITIMI		83
2.1	Soğuk Haddelme	83
2.2	Sürekli Tavlama Hattı Tanıtımı	85
2.2.1	Sürekli Tavlama Hattı Genel	85
2.2.2	Sürekli Tavlama Hattı Fırın Bölümü	857
2.3	Sürekli Tavlama Hattı Modeli	90
2.3.1	Sürekli Tavlama Hattı Modeli Genel	90
2.3.2	Sürekli Tavlama Hattı Sıcaklık Kontrolü	91
BÖLÜM 3: HEDEF PROGRAMLAMA VE YAPAY SİNİR AĞI UYGULAMALARI		95
3.1	Problem Tanımı	95
3.2	Metodoloji.....	97
3.2.1	Giriş Konveyörü Optimizasyon Uygulaması.....	97
3.2.2	Yapay Sinir Ağının Oluşturulması	104
3.3	Veri Toplama	112
3.4	Uygulama ve Sonuçlar.....	122
3.5	Analiz.....	138
SONUÇ VE ÖNERİLER		145
KAYNAKÇA.....		151
EKLER		162
ÖZGEÇMİŞ		222

KISALTMALAR

STH	: Sürekli Tavlama Hattı
NKK	: Nippon-Kokan Kabushiki Kaisha
YSA	: Yapay Sinir Ağı
TÜM	: Toplam Üretim Maliyetleri
EM	: Enerji Maliyetleri
DP	: Doğrusal Programlama
DOP	: Doğrusal Olmayan Programlama
NLP	: Lineer Olmayan Programlama
ÇKA	: Çok Katmanlı Algılayıcı
GBYSA	: Geri Beslemeli Yapay Sinir Ağı
BPN	: Back Propagation Networks
SOM	: Self-Organizing Map
PHS	: Preheating Section
RTH	: Radiant Tube Heating
RTS	: Radiant Tube Soaking
GJC	: Gas Jet Cooling
RQ	: Roll Quench
OAS	: Over Aging Section
FWQ	: Final Water Quench
HMK	: Hacim Merkezli Kübik
YMK	: Yüzey Merkezli Kübik
HNX	: Hydrogen Nitrogen Mixtures (Hidrojen-Azot Karışımı)
MES	: Manufacturing Execution Systems (Üretim Yönetim Sistemi)
WPD	: Kaynak Noktası Detektörü
PLG	: Darbe Sayıcı (Uzunluk ölçmede kullanılıyor.)
LCP	: Giriş ve Çıkış Şerit Depolama Taşıyıcısının Pozisyon Sensörü
PLC	: Programlanabilir Mantıksal Denetleyici
PIO	: PLC I/O
MSE	: Mean Squared Error (Karesel hata ortalaması)
MSEREG	: Ağ Performans Ölçme Fonksiyonu

ABD	: Amerika Birleşik Devletleri
DCS	: Dağıtılmış Kontrol Sistemi
ECS	: Elektriksel Kontrol Sistemi
PLG	: Pulse Generator
WPD	: Weld Point Dedector
TX	: Position Sensor of Looper Carriage
CSF	: Coil Charging Schedule File
GMD	: Gathered Material Coil Data
PCF	: Product Coil File
PCD	: Product Coil Data
PDT	: Periodical Data Collection Table

TABLO LİSTESİ

Tablo 1: Yönetim Bilimi Modelleme Tekniklerinin Karakteristikleri ve Kategorileri	16
Tablo 2: Optimizasyon Metotlarının Çözüm Mantığına Göre Sınıflandırılması	25
Tablo 3: Doğrusal Olmayan Optimizasyon için Çözüm Algoritması	30
Tablo 4: Yapay Zekâ Çalışmaları Gelişim Evreleri - 1	40
Tablo 5: Yapay Zekâ Çalışmaları Gelişim Evreleri - 2	41
Tablo 6: Biyolojik Sinir Hücresi Yapısının Yapay Sinir Ağları Yapısı İle Karşılaştırılması	45
Tablo 7: Toplama Fonksiyon Örnekleri	48
Tablo 8: Aktivasyon Fonksiyon Örnekleri - 1	49
Tablo 9: Aktivasyon Fonksiyon Örnekleri - 2	50
Tablo 10: Alaşım Elementlerinin Çeliğe Etkisi	73
Tablo 11: Kaliteye Göre Şerit Tavlama Sıcaklıkları	99
Tablo 12: Radyan Tüp Isıtma Bölümü Isıtma Kapasiteleri	100
Tablo 13: Matlab Newff Fonksiyonu Temel Kullanım Parametreleri	106
Tablo 14: YSA Geri Yayılım Ağı Eğitim Algoritmaları	107
Tablo 15: YSA Geri Yayılım Ağı Eğitim Algoritmalarının Kısaltılmış İsimleri	108
Tablo 16: Kıyaslama için Kullanılan Fonksiyonlar	108
Tablo 17: Sin Veri Seti için Kıyaslama Sonuçları	108
Tablo 18: Parity(Eşlik) Veri Seti için Kıyaslama Sonuçları	109
Tablo 19: Motor Veri Seti için Kıyaslama Sonuçları	109
Tablo 20: Kanseri Veri Seti için Kıyaslama Sonuçları	109
Tablo 21: Kolesterol Veri Seti için Kıyaslama Sonuçları	109
Tablo 22: Diyabet Veri Seti için Kıyaslama Sonuçları	110
Tablo 23: Hat Şerit Takip Sistemi Bölgeleri	115
Tablo 24: Ürün Veri Toplama İsim Listesi	118
Tablo 25: 1. Deneme İçin En Uygun Sıcaklık Grafiği	123

Tablo 26: 2. Deneme İçin En Uygun Sıcaklık Grafiği.....	124
Tablo 27: 3. Deneme İçin En Uygun Sıcaklık Grafiği.....	125
Tablo 28: 4. Deneme İçin En Uygun Sıcaklık Grafiği.....	126
Tablo 29: 5. Deneme İçin En Uygun Sıcaklık Grafiği.....	127
Tablo 30: 6. Deneme İçin En Uygun Sıcaklık Grafiği.....	128
Tablo 31: 7. Deneme İçin En Uygun Sıcaklık Grafiği.....	129
Tablo 32: 8. Deneme İçin En Uygun Sıcaklık Grafiği.....	130
Tablo 33: 9. Deneme İçin En Uygun Sıcaklık Grafiği.....	131
Tablo 34: 10. Deneme İçin En Uygun Sıcaklık Grafiği.....	132
Tablo 35: 11. Deneme İçin En Uygun Sıcaklık Grafiği.....	133
Tablo 36: YSA Denemeler Tablosu	138
Tablo 37: 453-tansig-trainscg Gerçek Fırın Sıcaklıkları.....	142
Tablo 38: 453-tansig-trainscg Hesaplanan Fırın Sıcaklıkları.....	143
Tablo 39: 453-tansig-trainscg Hesaplanan ile Gerçek Fırın Sıcaklıkları Arasındaki Fark	144

ŞEKİL LİSTESİ

Şekil 1: Bazı Sanayi Kollarında Tüm Üretim Maliyetleri İçinde Enerji Maliyeti Oranı (%).....	2
Şekil 2: Demir ve Çelik Üretim Süreci Akış Şeması.....	3
Şekil 3: Sürekli Tavlama Hattı Akış Şeması	3
Şekil 4: Problem Çözme Adımları.....	18
Şekil 5: Bir Fonksiyonun Yerel ve Küresel Minimum Noktaları.....	22
Şekil 6: Hessian Matrisi.....	27
Şekil 7: Biyolojik Sinir Hücresi.....	42
Şekil 8: Genel Yapay Sinir Ağı Yapısı.....	45
Şekil 9: YSA Yapısı	51
Şekil 10: Tek Katmanlı Yapay Sinir Ağı Modeli	57
Şekil 11: İki Sınıfa Ayıran Doğrunun Geometrik Gösterimi.....	58
Şekil 12: Çok Katmanlı Yapay Sinir Ağı	59
Şekil 13: Geri Yayılım Algoritması.....	60
Şekil 14: Geri Yayılım İşleminin Akış Diyagramı	62
Şekil 15: Tipik Hopfield Ağı	64
Şekil 16: Delta Öğrenme Kuralı Şematik Gösterimi	65
Şekil 17: Allotropi Yapı	74
Şekil 18: Faz Dönüşüm Eğrisi (Ötektoit)	75
Şekil 19: Fe-C Denge Diyagramı	77
Şekil 20: Çelik Yapısındaki Deformasyon	78
Şekil 21: Çelik İçindeki Moleküler Yapının Isı ile Değişimi.....	79
Şekil 22: Çelikler İçin Yeniden Kristalleştirme ve Gerilim Giderme Isıl İşlem Sıcaklık Bölgeleri.....	81
Şekil 23: Soğuk Haddelme Hattı.....	85
Şekil 24: Tipik Bobin Görüntüsü.....	85

Şekil 25: Sürekli Tavlama Hattı Üretim Akış Şeması	86
Şekil 26: STH Giriş Bölümü Konveyör Görünümü	87
Şekil 27: STH Merdaneli Soğutma Isı Transfer Modeli.....	89
Şekil 28: Şerit Tavlama Döngüsü	91
Şekil 29: Şerit Sıcaklık Kontrolünün Diğer Sistemlerle olan İlişkisi	92
Şekil 30: Tavlama Fırının Sıcaklık Kontrolü Yapısı	93
Şekil 31: Program Seçim Akış Diyagramı	95
Şekil 32: Bobin Takip Sistemi Genel Şeması.....	114
Şekil 33: Hat Bobin İzleme Bölgeleri Ve Sensör Konumları.....	115
Şekil 34: Elektriksel Olay Temelli Veri Toplama Fonksiyonu	116
Şekil 35: Fırın Periyodik Veri Toplama Fonksiyonu	117
Şekil 36: 1. Deneme Kalınlık, Genişlik, Hıza Bağlı En Uygun Hedef Sıcaklıkları ...	123
Şekil 37: 2. Deneme Kalınlık, Genişlik, Hıza Bağlı En Uygun Hedef Sıcaklıkları ...	124
Şekil 38: 3. Deneme Kalınlık, Genişlik, Hıza Bağlı En Uygun Hedef Sıcaklıkları ...	125
Şekil 39: 4. Deneme Kalınlık, Genişlik, Hıza Bağlı En Uygun Hedef Sıcaklıkları ...	126
Şekil 40: 5. Deneme Kalınlık, Genişlik, Hıza Bağlı En Uygun Hedef Sıcaklıkları ...	127
Şekil 41: 6. Deneme Kalınlık, Genişlik, Hıza Bağlı En Uygun Hedef Sıcaklıkları ...	128
Şekil 42: 7. Deneme Kalınlık, Genişlik, Hıza Bağlı En Uygun Hedef Sıcaklıkları ...	129
Şekil 43: 8. Deneme Kalınlık, Genişlik, Hıza Bağlı En Uygun Hedef Sıcaklıkları ...	130
Şekil 44: 9. Deneme Kalınlık, Genişlik, Hıza Bağlı En Uygun Hedef Sıcaklıkları ...	131
Şekil 45: 10. Deneme Kalınlık, Genişlik, Hıza Bağlı En Uygun Hedef Sıcaklıkları .	132
Şekil 46: 11. Deneme Kalınlık, Genişlik, Hıza Bağlı En Uygun Hedef Sıcaklıkları .	133
Şekil 47: Veri Tabanı Kayıt Sayısı	134
Şekil 48: Düzenlemelerden Sonra Kalan Veri Satırı Sayısı	137
Şekil 49: 453-tansig-trainscg YSA için Nntraintool Görünümü	139
Şekil 50: 453-tansig-trainscg YSA için En İyi Performans Değeri.....	140

Şekil 51: 453-tansig-trainscg YSA için Eğitim Durumu.....	140
Şekil 52: 453-tansig-trainscg YSA için Regresyon Değerleri.....	141
Şekil 53: Akademik Yayınların Gelişimi	149
Şekil 54: Organizasyonların Zeki Sistemlere olan İlgileri	149



Tezin Başlığı: Sürekli Tavlama Hatlarında Enerji Giderinin Kalite ve Boyut Değerlerine Göre Optimize Edilmesi ve Geçişlerde Operatör Davranışlarının Modellenmesi

Tezin Yazarı: Ömer Faruk Sezer

Danışman: Prof. Dr. Erman Coşkun

Kabul Tarihi: 19 Temmuz 2017

Sayfa Sayısı: xi (ön kısım) + 161 (tez) + 61(ek)

Anabilimdalı: İşletme

Bilimdalı: Üretim Yönetimi ve Pazarlama

Soğuk haddelemenin en son işlemi olan tavlama, soğuk haddelenmiş ürünlerin kalitesini arttırmak için çok önemlidir. Çelik, yüksek basınçta ince bir şerit haline haddelendiği zaman sertleşir ve en son ürünler olan otomobil parçaları, bilgisayar kasaları, yapı elemanları ve benzerleri gibi ürün haline getirilmeye çalışıldığı zaman başarılı olunamaz. Bu nedenle sürekli tavlama hattı fırın bölümünde; şerit bünyesinde soğuk haddeleme sırasında oluşan iç yapı bozukluğu ve iç gerilme birikiminin yeniden kristalleşme sıcaklığına kadar tavlanoşu giderilmesi amaçlanmaktadır.

Tavlama esnasında malzeme kalitesini etkileyen faktörler arasında, şerit sıcaklığı anahtar konumdadır. Kaliteli ürün üretmek için her fırının çıkışındaki şerit sıcaklığını istenilen değerlerde tutturmak çok önemlidir. Her şeridin tavlama sıcaklığı tavlama kalitesine bağılı olarak değişir. Ayrıca, şerit kalınlığı, şerit genişliği, şerit hızı ve fırın sıcaklığı koşulları diğere tavlama da etkili faktörlerdir. Bu faktörlere bağılı olarak siparişin işlenmesi sırasında uygun sıcaklıkların bulunması rekabet koşulları altındaki piyasada uygun enerji maliyetlerin sağlanması için önemlidir.

Bu çalışmanın ilk kısmında hatta girmesi kesinleşen on bobin için çok hedefli karar verme tekniğı kullanılarak fırın kapasite sınırlılıkları altında en uygun tavlama sıcaklıklarının bulunması amaçlanmıştır. Gerçekleştirmek için Microsoft Silver Foundation yazılımı kullanılmıştır.

İkinci kısımda ise kalite ve boyut geçişleri sırasındaki ısıtma miktarları fırın kapasite sınırlarını aştığı için işlem matematiksel modeller ile karşılanamamaktadır. Bu durumlarda operatörlerin yaptığı uygulamalar YSA ile modellenmiştir. Hatta işlenen bobinler için tavlama kalitesi veya şerit boyutlarının değişimleri esnasında fırın içinde bobin uzunluklarının son %50'si kaldığında operatör uygulamalarının kayıt edildiğı verilerden yararlanılarak yapay sinir ağı oluşturulmuş ve eğitilmiştir. Hat verileri 3 ay boyunca toplanmış, değerlendirilmiş, hatalı veriler ayıklanmış ve Matlab yazılımına sunulmuştur.

Literatürde tavlama hakkında birçok çalışma bulunmasına rağmen bu geçişleri sağlayacak model üzerinde çalışma konusuna bizim yaptığımız literatür taramalarında rastlanamamıştır. Bu genellikle fırın otomasyonu yapan firmaların teknik bilgileri olduğu için dışarıdan bu konuda bilgi edinilmesi zordur. Ayrıca bu hatların sayıca çok az olması bu konuda çok sayıda akademik çalışma yapılmasını engellemektedir.

Yaptığımız bu çalışma ile geleneksel matematiksel modeller yerine konulabilecek hem maliyet olarak ucuz hem de uygulanması kolay, her firmanın kendi teknik bilgisini yansıtabileceğı bir yöntem önerilmiştir.

Anahtar Kelimeler: Sürekli Tavlama Hatları, Yapay Sinir Ağları, Çok Hedefli Karar Verme Teknikleri, Optimizasyon Algoritmaları

Title of the Thesis: Optimizing Energy Consumption of Continuous Annealing Line Regarding Quality and Size Values and Modelling Operator Behaviour During Quality and Size Transition	
Author: Ömer Faruk Sezer	Supervisor: Professor Erman COŞKUN
Date: 19.July.2017	Nu. of pages: xi (pre text) + 161 (main body)+ 61 (App.)
Department: Business Admin.	Subfield: Production Management and Marketing
Annealing, which is the last operation of cold rolling, is very important process to increase the quality of cold rolled products. The steel hardens when rolled into a thin strip under the high pressure and cannot be successful when it is attempted to make the latest products such as car parts, computer cases, building elements and the like. For this reason, in the furnace section of the continuous annealing line it is intended to anneal the strip to the recrystallization temperature in order to remove the internal structure defect and internal stress of the strip occurring during cold rolling. Among the factors affecting the quality of the material during annealing, the strip temperature is the key factor. It is very important to keep the strip temperature at the desired value at the outlet of each furnace to produce quality products. The annealing temperature of each strip varies depending on the annealing quality. In addition, the strip thickness, strip width, strip speed and furnace temperature conditions are other annealing factors. Depending on these factors, the availability of suitable temperatures during the processing of the order is important to ensure adequate energy costs in the competitive market. In the first part of this study, it was aimed to find optimum annealing temperatures under furnace limitations by using multi-purpose decision making technique for ten coils. Microsoft Silver Foundation software was used to accomplish this. In the second part of this study, the process cannot be met by mathematical models since the heating quantities during the quality and size transitions exceed the furnace capacity limits. The operators' applications are modeled by YSA in these cases. Operator behaviors for annealing furnaces was recorded for the last 50 % of the coil lengths in the furnace during annealing quality or coil dimensions changes. Utilizing these data, an artificial neural network will be created, trained and offered to use with Matlab software. Although there are many studies about tempering in the literature, we did not find it in the literature reviews. It is often difficult to obtain information from outside because it is the technical information of the companies. Moreover, the number of these companies are very few in the world and it prevents many academic studies from being conducted on this issue. With this work, we have proposed a method that traditional mathematical models can be replaced with firms' own technical knowledge will reflect. It is both cheap in cost and easy to implement.	
Keywords: Continuous Annealing Line, Artificial Neural Networks, Multiple Criteria Decision Making Techniques, Optimization Algorithms	

GİRİŞ

Tezin Önemi

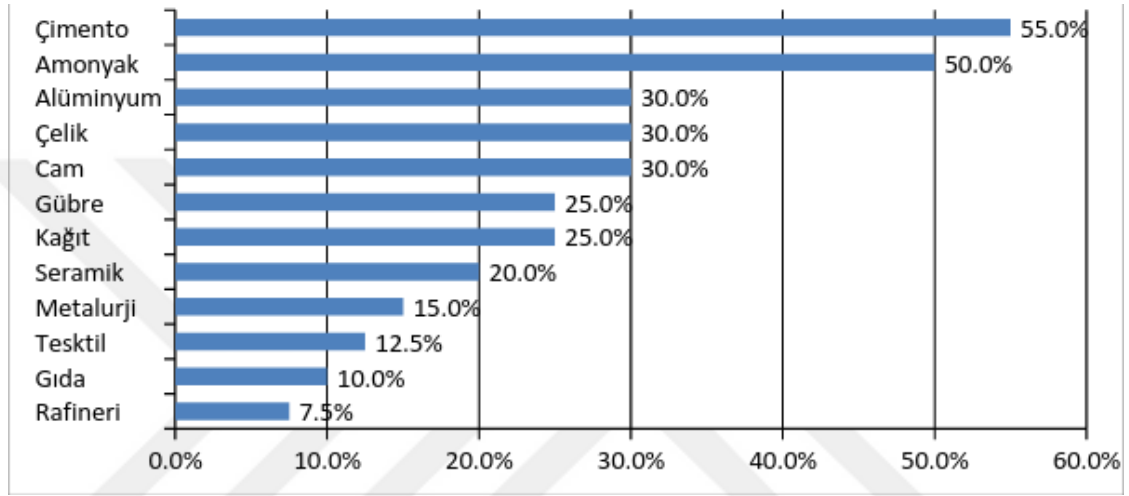
İşletmelerin rekabet koşulları altında varlıklarını ve karlılıklarını sürdürebilmeleri her alanda verimliliklerini arttırmalarına, maliyetlerini en az düzeye indirmelerine, operasyonel etkinliğini en üst düzeye taşımalarına ve operasyonel verimliliklerini maksimum seviyeye çekmelerine bağlıdır. İşletme maliyetleri arasında önemli bir yer tutan enerji maliyetleri ise; maliyetlerini en az düzeye indirme hipotezine en iyi örnektir. Artan enerji maliyetleri işletme maliyetlerini arttırmış bunun sonucu olarak işletmeler enerji maliyetlerini makine bazında takip etmeye başlamışlardır. Bu yöntem sonucu olarak da enerji kullanımının takibinin önemi bir kat daha artmıştır. Bunun yanı sıra artan küresel rekabet ve zorlaşan çevresel standartlar işletmelerde enerji kullanımında maksimum verimliliği gerektirmektedir.

Sanayi enerji tüketimi ülkemizde nihai enerji tüketimlerinde en yüksek paya sahip sektördür. Ülkeler arası ekonomik rekabetin artması, koşulların zorlaşması ve olumsuz çevre yaptırımlarının gün geçtikçe daha da ağırlaşması enerji tüketiminin etkin kullanımını ön plana çıkarmaktadır. Gelişen sanayiler enerjiyi etkin takip ederek ve enerji tüketimini azaltarak ayakta kalabilmektedirler. Üretim yaparken enerji kaynaklarının verimli kullanılmaması işletmelerin rekabet koşullarında başarısızlığa uğramasını kaçınılmaz kılmaktadır.

Sanayide enerji verimliliği çalışmaları maliyeti doğrudan etkilediğinden karlılığa olumlu yansıması ve yapılan yatırımların kısa sürede kendini amorti etmesi rekabet koşullarında ciddi avantaj sağlayabilmektedir (Elektrikport, 2013). Bunun yanında yapılacak yatırımlar diğer sektörlerle göre daha maliyetli yatırımlardır. Bu nedenle çok sayıda ülkede enerji verimliliği çalışmaları öncelikli hale gelmiştir. Bu öncelik özellikle sanayi sektöründe daha önemlidir. Enerji yoğun sanayilerde yapılan çalışmalarla enerjinin geri kazanımı için yatırımlar yapılmakta olup enerji maliyetleri aşağı çekilmeye çalışılmaktadır.

Enerji Yönetimi ürün kalitesinden ve miktarından ödün vermeden güvenlik ve çevresel tüm koşullardan fedakârlık etmeksizin enerjinin daha etkin ve verimli kullanılması doğ-

rultusunda yapılan çalışmalar bütünüdür (Elektrikport, 2013). İşletmelerde enerji verimliliğinde başarı, bilinçli bir enerji yönetim sistemi ile olur. Bu nedenle şirketler bütün yönetim kademelerinin takip edebileceği kurumsal sistemler kurmaya özen göstermişler ve bu doğrultuda yatırım yapmışlardır (TSKB,2011). Bu yatırımlar sayesinde ufak harcamalar ile enerji geri dönüşüm projeleri hayata geçirilmeye başlanmıştır. Tez çalışması yapılan demir çelik sektöründe Şekil 1’de görüleceği gibi toplam üretim maliyetleri içinde enerji maliyeti %30’u bulmaktadır.

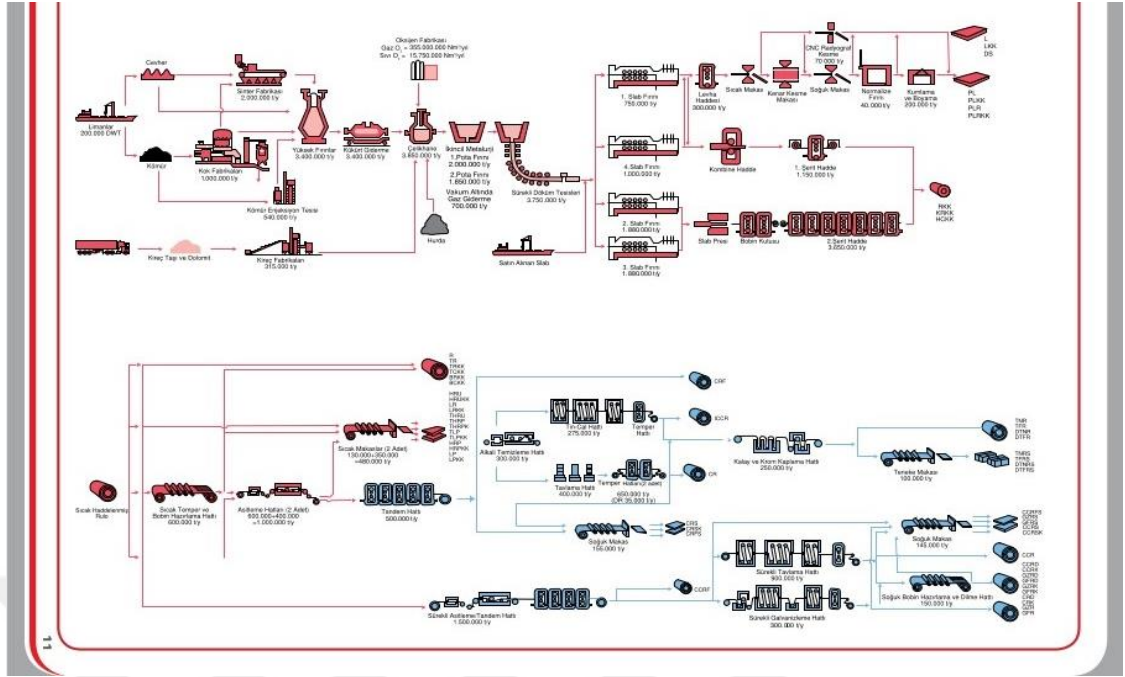


Şekil 1. Bazı Sanayi Kollarında Tüm Üretim Maliyetleri İçinde Enerji Maliyeti Oranı (%)

Kaynak: Ulusal Verimlilik Kongresi, Sanayide Verimlilik, Enerji Yönetimi ve Tasarruf, 2015

Tezin Kapsamı ve Amacı

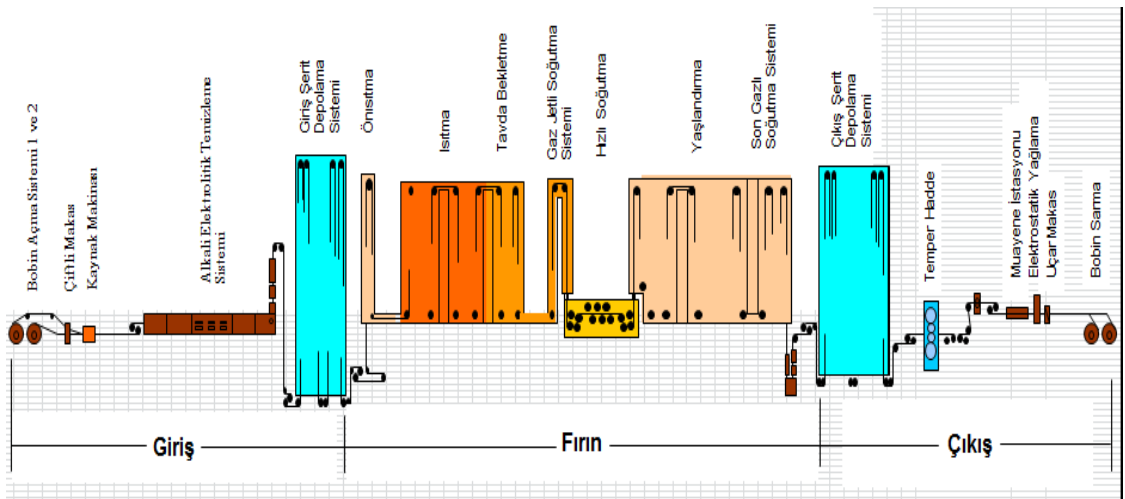
Soğuk haddeleme sürecinden sonraki süreç ve aynı zamanda soğuk haddelenmiş ürünler için en son süreç olan sürekli tavlama süreci soğuk haddelenmiş ürünlerin kalitesini arttırmak için önemli bir rol oynar. Nippon Steel Kimitsu İşletmesinin 1972’de ilk sürekli tavlama hattını (STH) kurmasından sonra, yüksek üretim kapasitesi, düşük maliyet, yüksek kalite ve çeşitlilik avantajları nedeniyle dünyada kullanımı yaygınlaşmıştır. Sürekli tavlama sürecinin çelik üretimindeki yeri Şekil 2’de gösterilmiştir.



Şekil 2. Demir ve Çelik Üretim Süreci Akış Şeması

Kaynak: Erdemir (STH), 2017

Fırın bölümünde; şerit bünyesinde soğuk haddeleme sırasında oluşan içyapı bozukluğu ve iç gerilme birikiminin yeniden kristalleşme sıcaklığına kadar tavlanıp giderilmesi amaçlanmaktadır (Sezer ve Coşkun, 2016). Tavlama hatlarında yassı çelik ürüne (şerit) istenilen fiziksel özellikleri kazandırmak için çeşitli ısı işlemleri uygulanır. Şekil 3’de ise tipik bir STH verilmiştir.



Şekil 3. Sürekli Tavlama Hattı Akış Şeması

Kaynak: Erdemir (STH), 2017

STH’larda enerji tasarrufu yapılacak iki önemli uygulama vardır. Bunlar hatta girecek olan ürünlerin en uygun sırada ve değerlerle ısıtma ve soğutma işleminin yapılması ve ürünlerin olabilecek en hızlı ve kaliteli olarak üretilmesidir. Eğer şartlar en uygun sırada ürünleri hattan geçirmeye müsaade etmiyorsa bu durumda hatta girecek ürünlere en uygun sıcaklıklarda ısıtma ve soğutma işlemleri yapmak enerji verimliliği açısından bir kat daha önem kazanmaktadır.

Tavlama ısıtma işlemleri sırasında uygulanan dinamik şerit sıcaklığı kontrol modelinin ana amacı maksimum hat hızında çalışmayı sağlamak ve şerit sıcaklığını kalitesine uygun sıcaklık değerlerinde tutmaktır. Bunu yaparken fırın sıcaklığı ve hat hızı gibi parametreleri kullanır. Şerit sıcaklığını istenilen değerlere getirmek amacıyla termodinamik ısı denklemini, fırın ısıtma kapasitesine ve hat hızına göre fırın sıcaklık değerlerini hesaplar. Dinamik şerit sıcaklık kontrol sistemi set noktası değerlerinin hesaplanması veya çıkış şerit sıcaklık tahmini için ısı transferi teorisine dayanan matematiksel modeller kullanır. Bu sistemde kullanılmak üzere geliştirilmiş matematiksel modeller Ek-4’de listelenmiştir.

Üzerinde dinamik şerit sıcaklığı kontrol modellerinin çalıştığı proses bilgisayarları hattı iki şekilde kontrol eder. Eğer şerit sıcaklıkları değiştirilmeyecek yani boyut ve kalite olarak yakın ürünler ise otomasyon sistemine sadece set değeri gönderir ve otomasyon sistemi bu değere göre kendi fırın sıcaklıklarını ayarlar. Fakat boyut ve kalite olarak bir sonraki ürün farklı ise bu defa tüm kontrolü kendi üzerine alır. Eğer sonraki ürün için fırın sıcaklığının artırılması gerekiyor ise önceden fırın sıcaklığını arttırmaya başlar. Fırın sıcaklığını öyle ayarlar ki bir sonraki ürün geldiğinde şerit sıcaklığı şimdiki ürünün üst limiti ile sonraki ürünün alt limiti olur. Bu işlemi yaparken hattın hızını da yavaşlatması otomasyon kontrolü için ayrı bir parametredir. Eğer iki ürün arasında sıcaklık farkı fırın ısıtma kapasitesinin üstünde ise bu defa ara ürün kullanılması gerekir. Bu ara ürün, sipariş edilen ürünler arasında var ise bu ürün işlenir fakat siparişler arasında yok ise bu ürün yerine ara ürün olarak stoklarında sadece bu iş beklettikleri ürün kullanılır.

Bu modeller kesin değerler içerdiği için belirli sıcaklıklar arasında bulunması gereken değerler geleneksel otomasyon sistemleri ile gerçekleştirilmeye çalışılır. Bu geleneksel

matematik modellerin ve otomasyon sistemlerinin eksik yanlarından biridir. Yeni durumlara göre kendini ayarlama şansları yoktur. Kurulduklarındaki gibi çalışırlar. Kendilerini geliştirme şansları yoktur.

Bir üretim hattı için istenen en önemli özelliklerden biri de siparişleri topladıktan sonra bunları üretim hatlarında üretim seviyesini arttıracak ve maliyetleri azaltacak en uygun sırada üretmektir. Bu özellikler sağlanamadığı takdirde üretim maliyetlerinde artış meydana gelmekte; bu da firma için rekabet edebilirlik düzeyini olumsuz etkilemektedir.

Firma için rekabet edilebilirliği arttırmak ve optimizasyonu sağlamak üzere üretim hatları optimizasyonu için alınan birçok teknolojik donanım ve yazılım, üretim planlama tarafından gönderilen siparişlere göre istenen sıralama özelliklerini sağlayamadığından kullanılamamakta, bunun yerine bir veya daha fazla personel, uygun bulunmayan optimizasyon sistemine çare olarak otomasyon kontrolü yapmaktadır. Bu durumda yazılıma yapılan yatırım boşa gitmiş olmakta aynı zamanda da bütün üretim süreç sorumluluğu operatörün eline bırakılmaktadır.

Prosesi kontrol eden bilgisayarların çalıştırdığı matematik modellerin hattın hızını düşürdüğü ve ara ürün kullanılmasının zorunluğu nedeniyle operatörler bilgisayar kontrolünün yerine hattı manuel kontrol ederek çalıştırmaktadırlar.

Operatörlerin farklı uygulama tarzları üretim hatlarında farklı sonuçların alınmasına neden olmakta ve üretim hatlarının optimum şekilde çalışması sağlanamamaktadır. Maliyet açısından bakıldığında; 2016 yılında 936,3 ton malzeme tav fırın sıcaklık hatalı kusurundan dispoze olmuştur. Dispoze oranı % 0,10 dur. Bu çalışma sonucunda getirilecek olan standartlaşmanın yararlarından biri de bu dispozelerin olmamasının sağlanmasıdır.

Yukarıda bahsedilen ara ürünler birden fazla kullanıldıkları için fırın içinde kopmaları mümkündür. Zaman zaman kullanıldıkları için yıpranmışlardır ve kopmaya müsait hale gelmişlerdir. Bu kopmalar sonucunda kopan parçanın çıkarılması ve hattın devreye alınması 24 saatlik süreyi bulmaktadır. Bu da üretim kaybına neden olur.

Ayrıca malzeme boyutları veya tavlama kalitesi değişimlerinde kurucu firmanın da önerdiği gibi araya gereken sıcaklık artırımını sağlayacak süre kadar bir ara ürün işlenmesi gerekmektedir. Bu malzeme de hurda ürün olduğu için bu malzemenin işlenmesi kadar gereken süre de boşa gitmektedir.

Bu nedenlerle bu geçişler esnasında bu kaybı telafi edecek önlemler alınmaktadır. Getirisi daha az olan malzemenin fırın içindeki kuyruk bölümünden başlayarak fırın sıcaklığı artırılmaya başlanmakta, bu malzemenin son kuyruk kısmı feda edilmektedir. Bu işletmenin getirdiği pratik bir yöntemdir. Bu pratiği uygulayacak geleneksel otomasyon kontrol sistemi yoktur.

Bu tez çalışmasında sürekli tavlama hatlarındaki enerji sarfiyatını azaltmak amacıyla hatta girecek olan soğuk haddelenmiş ürünler için şerit tavlama sıcaklıkları set değerlerini optimize edecek bir yöntem önerilmiştir. Sıcaklık optimizasyonu enerji tasarrufu sağlanması için bobinlerin en uygun sıcaklıklarda tavlama amacıyla uygulanmıştır. Öncelikle, ısı transferi, kapasite, ısı balansı, kapasitenin tav çevrimine göre değişimi, tavlama sınırlılıkları, radyan tüp ısıtma bölüm kapasiteleri değerleri dikkate alınarak sistem modellemesi gerçekleştirilmiş ve daha sonra sistemin amaç ve kısıtları belirlenerek hedef programlama işlemi için gerekli olan öncelik durumlarının belirlenmesi sağlanmıştır (Sezer ve Coşkun, 2016). Bu çalışmada çok hedefli programlama kullanılarak sistemin enerji tasarrufu getirecek en uygun sıcaklık ve hızda çalışması için optimizasyon denklemi oluşturulmuş ve programı yazılmıştır.

Operatörlerin yukarıda anlatılan geçişler sırasındaki kontrollerinin yerine konulabilecek ve en uygun operatör davranışlarının seçilerek hat kontrolünde standart sağlayacak yapay sinir ağı (YSA) modellemesi ise ikinci uygulama olarak geliştirilmiştir. Fırın içinde tavlanan bobinlerin bobin geçişleri sırasında kalan son % 50 uzunlukları sırasında operatörlerin davranışlarının modellenmesine çalışılmıştır.

Tezin Yöntemi

Bilgisayarlar, programlanmış sabit kuralları kullanarak birbirini tekrar eden monoton işlemleri doğru ve güvenilir şekilde hızla yapabilmektedir. Ancak, insanların tersine bil-

gisayarların yeni durumları anlama kabiliyetleri sınırlıdır ve bu durumlara uyum sağlaması zordur. Ancak yapay zekâ sistemleri, bilgisayarların bu karmaşık sorunların üstesinden gelebilmesi için gerekli yeteneklere sahip sistemlerdir.

Yapay sinir ağları yapay zekâ biliminin altında araştırmacıların çok yoğun ilgi gösterdikleri bir araştırma alanıdır ve bilgisayarların öğrenmesine yönelik çalışmaları kapsamaktadır. Teknolojinin gelişmesiyle birlikte önceleri sadece karmaşık hesaplamaları gerçekleştirmek üzere geliştirilen bilgisayarların zaman içerisinde büyük sayıdaki veriyi işleyerek özetleyebilen ve mevcut veri ile olaylar hakkında sonuçlar çıkarabilen nitelikler kazandığı görülmektedir (Öztemel (2012)). Günümüzde ise bilgisayarlar hem olaylar hakkında karar verebilmekte hem de olaylar arasındaki ilişkileri öğrenebilmektedir. Matematiksel olarak modellemesi yapılamayan ve çözülmesi mümkün olmayan problemler sezgisel ve deterministik yöntemler yolu ile bilgisayarlar tarafından çözülebilmektedir. İnsan karar verme sürecine oldukça benzer bir karar verme sürecine kavuşmakta ve daha karmaşık fakat kullanışlı sistemler ortaya çıkmaktadır.

Zeki sistemlerin en temel özellikleri olaylara ve problemlere çözümler üretirken veya çalışırken bilgiye dayalı olarak karar verebilme özelliklerinin olması ve eldeki bilgiler ile olayları öğrenerek, sonraki olaylar hakkında kararlar verebilmeleridir. Yapay zekâ uğraşlarındaki bu gelişmeler yapay zeka sistemlerinde de çok çeşitli sistemlerin ortaya çıkmasına neden olmaktadır. Bilgisayar donanım teknolojisi günden güne gelişmekte daha hızlı çalışabilen, daha çok bilgiyi saklayabilen, daha karmaşık sistemleri ve fonksiyonları yerine getiren bilgisayarlar ve bilgisayar sistemleri oluşturulmaktadır. Diğer taraftan ise yazılım teknolojisi gelişmektedir. İşleme yetenekleri, öğrenme, karar verebilme, problem çözme, muhakeme yapabilme yöntemleri ve bu yöntemlere dayalı yazılım sistemleri geliştirilmektedir (Barışkan, 2012). Bu gelişmeler akıllı sistemlerin günümüzde yaygın olarak tanınmasına ve kullanıma sunulmasına sebep olmaktadır.

Artık bilgisayarlar eskiden olduğu gibi sadece bilgi paylaşımı ve hesaplama yapan otomasyon sistemleri olarak görülmemektedir. İnsan karar verme sürecine oldukça benzer bir karar verme sürecine kavuşmakta ve daha karmaşık fakat kullanışlı sistemler ortaya çıkmaktadır. Çalışmalar laboratuvar ortamlarında sınırlı kalmamaktadır. Her geçen gün daha fazla ticari sistemler için çalışmalar yapılmakta ve yeni projeler ortaya çıkarılmaktadır.

Problemin çözümünü veren bir formülün olmadığı veya tam olarak kullanılamadığı durumlarda bilgisayarlara problemleri çözdürmek önemli bir uğraş alanıdır. Yapay zekâ çözümü olmayan problemlerle uğraşan bir bilim alanıdır. Bilgisayarların, bunu yapabilmesi için problemle alakalı her türlü bilgi ile donatılmış olması gerekmektedir. En önemli sorunlar ise bilgilerin toplanması, analiz edilmesi ve uygun şekilde sunulmasıdır. Bilgisayarın sahip olduğu bilgi ne kadar doğru ve geçerli ise sonuçlarda o kadar doğru ve geçerli olacaktır. Bilginin elde edilmesinde değişik yöntemler kullanılmakla birlikte üç ana grupta toplamak mümkündür. Bunlar;

- Anketler, uzmanlar ile görüşmeler, mülakatlar, literatür taramaları vb. yollar ile ilgili olayı değişik açılardan inceleyerek bilgileri toplamak
- İşi uzmanları (çalışanları) ile birlikte yapmak
- İlgili olay için gerçekleşmiş örnekleri inceleyerek bilgileri elde etme (örneklerden öğrenme) (Öztemel (2012)).

Bunlardan özellikle üçüncüsü, bilinen örneklerle bakarak bilinmeyen ve sonradan oluşacak benzeri örnekler hakkında karar verebilecek bilgilerin bulunmasıdır. Değişik örneklerle tekrar tekrar bakarak önemli bilgilerin ortaya çıkartılması gerekmektedir. Her örnekten bir şeyler öğrenilerek zaman içerisinde ilgili olay hakkında karar verebilecek bilgi düzeyine ulaşılmaktadır. Bu olaya “*makine öğrenmesi*” denmektedir (Öztemel (2012)).

İlk çalışma için optimizasyon denklemi termodinamik ısı teorisine dayanılarak gerçekleştirilmiştir. En uygun çözümü bulmak için bu çalışmada Microsoft Solver Foundation yazılımının hedef programlama ile ilgili fonksiyonları kullanılmıştır. Microsoft Visual Studio .Net C# kullanılarak optimizasyon programı yazılımı ve algoritma olarak HybridLocalSearchDirective (hybrid local search solver) kullanılmıştır. Bu algoritmaya göre hatta girecek 10 adet bobin için en uygun tavlama sıcaklıklarının ne olması gerektiği bulunmuştur. İlgili kaynak kodu Ek-5’de verilmiştir. Microsoft HybridLocalSearchDirective metodu **stokastik tepe tırmanma** algoritması kullanılmaktadır.

İkinci çalışma için üç seçenek üzerinde durulmuştur. Bunlar;

- İstatistiksel veri analizi yöntemi kullanılması (Regresyon)
- Uzman sistem olarak tasarlanması
- Yapay Sinir Ağı olarak tasarlanması

İstatistiksel veri analizi yönteminin kullanılması ve uzman sistem olarak tasarlanması yeni hatta girecek ürünler için gelecek zamanlarda sisteme tanıtımlar yapılmasını gerektirdiğinden YSA öğrenmesi üzerinde durulmuştur. YSA'ların verilerden bakarak öğrenme yapabilmesi, öğrenmeye sürekli olarak devam edebilmesi, hata toleransına açık olması, donanım olarak gerçekleştirilebilmesi ve paket yazılımlarının olması gibi sebepler yüzünden seçilmesine karar verilmiştir.

Bu çalışma için 3 ay boyunca gerçek veriler süreç bilgisayarlarından alınarak veri tabanında saklanmaya başlanmıştır. Bu veriler daha sonra Microsoft Excel ortamına aktararak değerlendirilmiş ve işletme mühendisleri tarafından kalite ve üretim hızı açısından uygun olmayan veriler ayıklanmıştır. Bu tez çalışmasında geçişlerdeki veriler önemli olduğundan diğer veriler silinmiştir. Bu veriler hatların çalışma parametrelerinin sınır değerlerine göre normalize edilmiştir. Normalize edilen veriler Matlab ortamında oluşturulan YSA'na yüklenmiş ve en uygun hata değerleri verecek olan transfer fonksiyonları ve nöron sayıları sıra ile denenmeye başlanmıştır. Bunlar arasında logsig, tansig ve purelin transfer fonksiyonları sayılabilir. Ağ eğitim algoritmaları olarak trainbr (Bayesian regulation backpropagation), trainlm (Levenberg-Marquardt backpropagation) ve trainscg (Scaled conjugate gradient backpropagation) fonksiyonları verilebilir.

Tezin Bölümleri

Tez üç bölümden oluşmaktadır. Birinci bölümde teorik alt yapının verilmesine çalışılmıştır. Birinci bölümün ilk alt bölümünde çok hedefli karar vermenin tanıtımı yapılmış ve çok hedefli karar verme yöntemlerinden bahsedilmiştir. İkinci alt bölümde ise zeki sistemlerin ve yapay zekânın tanıtımı yapılmış ve yapay sinir ağları hakkında bilgi verilmiştir. YSA türlerinden olan ileri beslemeli yapay sinir ağının tanıtımı yapılmıştır. Üçüncü alt bölümde demir ve çelik metal yapıları hakkında bilgi verilmiştir. Çelik metali için tavlama işleminin önemi anlatılmıştır. İkinci bölümde STH'ı tanıtılmış ve kullanılan matematiksel model hakkında bilgi verilmiştir. Üçüncü bölümde tez çalışması-

nın iki uygulaması anlatılmış ve sonuçları hakkında bilgi verilmiştir. Daha sonra değerlendirme, sonuç ve öneriler kısmında bulgular değerlendirilmiş ve gelecek araştırmalar için önerilerde bulunulmuştur. Bu tezin uygulaması sırasında karşılaştığımız kısıtlar hakkında bilgi verilmiştir.

Tezin Literatüre Katkısı

Literatür incelemesi sonucunda Sürekli Tavlama Hatları ile ilgili yapılan çalışmalar incelenmiş ve bu konuda ısıtma modeli ve fazla üretim hattı bulunan yerlerde ise üretim yeri optimizasyonu ile ilgili çalışmalar yapıldığı görülmüştür. Bunlar arasından ısıtma modeli ile ilgili çalışmalar üzerinde durulmuş ve yapılan çalışmalar incelenmiştir.

STH ile ilgili çalışmalar şunlardır; Ayan, Singh, Prasad, Chowdhurib (2012), yaptığı çalışmada sinir ağı ve klasik matematiksel modeli birleştirerek sistemin daha etkili kontrolü için matematiksel modelde iyileştirmeler yapmıştır. Carvalho, Ong ve Guimaraes (2006), tavlama sürecini kontrol etmek için matematiksel modeli geliştirmişlerdir. Ming, Datai, Jiangang (2008), kullanılmayan kontrol model yerine yeni şerit kontrol modelini yapay zekâ metotlarını kullanarak geliştirmişlerdir. Christopher, Watana-pongse and Gaskey (1988), lineer kaudratik Gauss kontrol tekniklerini kullanarak araba sacı üretmek için kontrol tekniğini geliştirmişlerdir. Wang, Li, Hua ve Liu (2010), tavlama fırını için yeni süreç kontrol ve kendi-kendini uyarlama (self-adapting) modeli geliştirmişlerdir. Zhang, Shuai ve Tenga (2012), özyinelemeli Kernel temel bileşen analizi (Recursive Kernel principal component analysis - RKPCA) metodunu kullanarak güvenilir çalışma ortamı oluşturmak için süreci izleme, arızaları tespit etme ve giderme amacıyla çalışmalar yapmışlardır. Ghasem, Mahmoud, Vaghefi, Lotfiani (2005), radyan tüplerin ömürlerini uzatmak için çalışmışlar. Shaoyuan, Chena ve Huang (2006), GGAP-RBF sinir ağını kullanarak kalite kontrolün dinamik modellemesi üzere çalışmışlardır. Yanagishima, Shimoyama, Suzuki, Sunami, Haga, Ida ve Irie (1981), Chiba Çelik Fabrikasında Sürekli Tavlama Hattı'nın karakteristiklerini ve operasyonunu anlatmışlardır. Yoshitani and Hasegawa (1998), vektör tipi değişken unutma faktörü ile özyinelemeli parametre kestirimi (recursive parameter estimation with a vector-type variable forgetting factor (REVVF)) metodunu kullanarak kendi kendini ayarlayabilen bir model geliştirmişlerdir. Fei, Yong-qin ve Shu-ren (2012), ısı transfer katsayılarını kendi

kendine ayarlayabilen bir metot geliřtirmişlerdir. Marlow (1996), kalite geçiřleri sırasında tavlama fırını için matematiksel bir model tanımlamaktadır. Arumugam, Chandramohan ve Murthy (2011), yapay zekâ algoritmaları kullanarak birden fazla tavlama fırını olan işletmelerde fırın kullanım optimizasyonu için yeni model oluşturmuřtur. Cho, Cassandras ve Kwon (2004), ileri ayırıştırma algoritması (forward decomposition algorithm) kullanarak birden fazla tavlama fırını olan işletmelerde fırın kullanım optimizasyonu için yeni model oluşturmuřtur. Guo, Zhang, You, Chen ve Zhang (2009), parçacık sürü optimizasyonu algoritmaları (Particle Swarm Optimization) kullanarak fırın kontrol modeli geliřtirmişlerdir. Jeong, Yi, Kim ve Ha (1991), her bölge için gaz sıcaklığını ölçerek kontrol metodu geliřtirmişlerdir.

Hedef programlama ile ilgili olarak da birçok çalışma yapılmıştır. Calvete ve diğerkleri (2007), araçların rotalama problemleri için optimizasyon yöntemi önermiştir. Brauer ve diğerkleri (1990) toplam envanter ve dağıtım planlamasında optimizasyon yöntemi kullanmıştır. Leung ve diğerkleri, üretim kısıtları altında toplam üretim planlaması için, Cristobal yenilenebilir enerji santralleri için yer ve en uygun karması için yöntemler önermişlerdir. Jadidi ve diğerkleri (2014) tedarikçi seçimi ve sipariř düzenlemesi için, Pal ve diğerkleri (2013) ekonomik ve çevresel güç jeneratörü dağıtım modeli için çalışmalar yapmışlardır. Liao ve diğerkleri (2013) lojistik servis operasyonları, Mukherjee ve Bera (1995) proje seçim kararı, Badri (1999) küresel tesis yeri seçimi sorunu üzerinde çalışmışlardır. Zheng ve diğerkleri (1996) lineer olmayan hedef programlama sorunları için, Zamfirescu ve Zamfirescu (2013) performans esaslı bütçeleme karar modeli için, Sofyaloğlu ve Öztürk (2013) tedarik zincirinde dağıtım planlaması ve bütçe ayırıştırması için, Finea ve diğerkleri (2005) ürün, süreç ve tedarik zinciri tasarımı içeren üç boyutlu eşzamanlı mühendislik (3D-CE) sorunları çözmek için, Moghaddam (2013) üretim sistemlerinde parça değıřim ve önleyici bakım işleri için, Choundary ve Shankar (2014) envanter boyutuna, tedarikçi seçimine ve taşıyıcı seçimi için, Chang (2015) yenilenebilir enerji tesislerinin optimum konumu modeli için, Tamiz ve diğerkleri (2013) uluslararası yatırım fonları arasından portföy seçilmesi için, Karsak ve diğerkleri (2002) kalite fonksiyon konuşlandırılması içinde ürün planlaması için, Paksoy ve Chang (2010) gerilla pazarlamada açılan mağazalar için çok aşamalı envanter kontrollü tedarik zinciri modeli için, Ghahtarani ve Hajafi (2013) portföy seçimi için, Agha (2006) çeřitli kaynaklardan gelen suları karıştırarak uygun klor ve nitrat değıerlerini bulmak için, Zhao ve

diğerleri Soğuk Haddeleme işlemlerinde uygun üretim planını sağlamak için hedef programlama kullanmışlardır.

Yapılan literatür taraması sonucunda; hatta girecek olan ürünler için sıcaklık optimizasyonu yapan ve aynı zamanda kalite ve boyut geçişleri sırasında operatör davranışlarını modelleyen bir çalışmaya rastlanamamıştır. Rastlanamama nedeni olarak bu hatların çok az sayıda olması ve tavlama fırını üreticilerinin kendi teknik bilgilerini dış çevrelere aktarmaması olabilir. Matematiksel modeller, üretici firmaların bilgi birikimleri olduğu için kurulum maliyetleri çok yüksektir. Üretim koşulları da matematiksel modeli çalıştırmaya izin vermiyor ise yatırımlar ve harcamalar boşa gitmektedir. Bunların yerine maliyeti çok az olan ve uygulaması kolay olan bir yöntem önerilmektedir.

BÖLÜM 1: KAVRAMSAL ALTYAPI

Bu bölüm altında çok hedefli karar verme teknikleri, doğrusal olmayan programlama, yapay sinir ağları hakkında bilgi verilecektir. Çeliğin genel yapısı anlatılacak ve tavlama işleminin çelik için önemi belirtilecektir.

1.1 Çok Hedefli Karar Verme Teknikleri ve Doğrusal Olmayan Programlama

Çok hedefli karar verme, karar vermenin bir ileri adımudur. Bu karar verme türünde elimizde ulaşmamız gereken birden çok karar bulunur. Bunların önem sırasına göre veya önem ağırlığına göre karşılamamız gerekir. Hepsini karşılamamız mümkün değildir. O zaman çok hedefli karar verme ve tekniğinden önce karar verme kavramının anlatılması gerekir.

1.1.1 Karar Verme

Karar verme, zihinsel işlemlerin sonucunda, çeşitli alternatifler arasından kişiye göre en tatmin edici birisinin seçilmesi sürecidir. Her karar verme sürecinin sonucu bir karar olmalıdır. Sonuç, yapılması gereken bir iş veya düşünce olabilir. Fakat her karar doğru olmayabilir. Her kararın en uygun alternatif olması için duygulara değil mantığa dayanması, tüm mevcut kaynakların en uygun şekilde kullanılıyor olması, tüm olası seçeneklerin araştırılmış, incelenmiş ve değerlendirilmiş olması ve bu alternatiflerin tüm olasılıklarını değerlendirebilecek iyi bir modele oturtulması gerekmektedir. Tüm bunların yanında doğru bir karar için yine de bir şansa ihtiyaç vardır.

İyi bir karar iyi bir modelleme ile mümkündür. Koçak (2015)'a göre model, bir sistemin değişen şartlar altındaki davranışlarını incelemek, kontrol etmek ve geleceği hakkında tahminlerde bulunmak amacıyla elemanları arasındaki bağıntıları kelimeler veya matematik terimlerle belirten ifadeler topluluğudur.

Ragsdale'e (2008) göre problem çözmede ve karar verme sürecinde modelleme yeni bir gelişme değildir; modellemeyi bilgisayarların keşfine bağlamak da doğru değildir. Hayatımızda birçok kez karar verme sorunu yaşarken modelleme yapmak zorunda kalmışızdır. Örneğin evimizi döşerken elimizdeki bütçeye göre, zevklerimize göre ve evin durumuna göre birçok değişik alternatif belirleyip en uygun çözümü aramaya başlayıp bazen bütçemizden bazen zevklerimizden ödün vererek karar vermişizdir. Ragsdale

(2008), bu durumu zihinsel modelleme olarak nitelendirmiştir ve modellemeyi dört kategoriye ayırmıştır.

- a. Zihinsel Modelleme
- b. Görsel Modelleme
- c. Fiziksel Modelleme
- d. Matematiksel Modelleme

Zihinsel modelleme en basit modelleme çeşididir ve karmaşık problemler için yeterli gelmeyebilir. Evin odasının mobilya yerleşimi zihinsel bir modelledir. Bu işlemi zihnimizde sorunsuzca yapmak kolaydır. Görsel modelleme, ev veya bina krokisi gibi zihinsel olarak çözülemeyecek kadar karmaşık olan durumlarda kullanılan problemi geniş olarak görmemizi sağlayan bir modelleme çeşididir. Bize farklı alternatif çözümleri grafiksel olarak bütünüyle gösterir. Fiziksel model ise zihinsel veya fiziksel olarak çözülemeyecek problemlerin modellenmesinde kullanılır. Yeni otomobil geliştirme çalışmalarında fiziksel model olmadan sürtünme katsayısı gibi parametreleri test etmek mümkün değildir. Bir diğer modelleme çeşidi matematiksel modelledir. Bu modelleme tekniği ile tüm sistemin matematiksel modeli oluşturulur ve çeşitli giriş verilerine göre çıkış değerlerinin ne olacağı anlaşılmaya çalışılır.

1.1.2 Matematiksel Modelleme ve Yönetim Bilimi

Winston (2003), matematiksel modellemeyi daha iyi karar verebilmek veya gerçek olayları daha iyi anlayabilmek için gerçek olayların matematiksel olarak ifade edilmesi olarak tanımlamıştır. Ragsdale'in (2008) Tablo-1'de ki kategorilere ayırdığı yönetim bilimi araştırmaları ise bir veya daha çok matematiksel modeli kapsayan bir karar verme yaklaşımıdır. Literatürde yöneylem araştırmaları olarak da bilinmektedir. Yönetim bilimi kısıtlı kaynaklar altında sistemi işletecek en uygun yapıyı bulmak amacıyla geliştirilmiş bilimsel bir karar verme yaklaşımıdır.

Değişen teknolojilerin, sınırlı kaynakların, artan rekabetin, karmaşık hale gelen sistemlerin doğurduğu problemlerin klasik yöntemlerle (matematiksel veya matematiksel olmayan, analitik veya sayısal) çözümünün güçleşmesi optimizasyon ve bunun yanında yönetim bilimi kavramının doğmasına neden olan en önemli sebeplerdendir (Çayiroğlu, 2015).

İlk yönetim bilimi arařtırmaları İkinci Dünya Savařı yıllarına dayanmaktadır. İngiltere askeri alanda düşmanlarına karşı üstünlük sağlamak amacıyla birçok disiplinler arası grubu bir araya getirerek çalışma başlatmıştır. Matematikçiler, istatistikçiler, mühendisler, fizikçiler, elektronikçiler, psikologlar, askeri personel ile birlikte önemli operasyonel askeri eylemler üzerinde çalışmak için görevlendirilmişlerdir. Bu çalışma daha sonra yöneylem arařtırmaları olarak adlandırılmıştır. Bunlardan İngiliz ordusuna yaptığı en önemli katkı olarak erken radar uyarı sistemi verilebilir. İngiliz ordusu bu çalışma sayesinde Britanya Savařı'nda zafer elde etmiştir. Bunun yanında bir konvoy için en uygun savaş gemisi sayısı, anti denizaltı savaş organizasyonu, bombalama doğruluęu gibi çalışmalar yapmışlardır. Bu grubun en önemli özellięi disiplinler arası düşüncelerin sadece deęerlendirilmesi deęil, bunun yanında ilk defa bu düşüncelerin bir araya getirilmesi olmuştur. Yöneylem arařtırmalarının İngiltere'de başarılı olması ve A.B.D.'deki Pearl Harbor baskınının sonucu olarak A.B.D.'de de bu çeřit gruplar kurulmuştur.

Savaş bittikten sonra da bu çalışmalara hem A.B.D.'de hem de İngiltere'de askeri alanda ve savunma sanayi alanında devam edilmiştir. A.B.D.'de askeriyeenin finanse ettięi matematik tekniklerini geliřtirmek amacıyla akademik alanda yöneylem arařtırmaları yapılmıştır. İngiltere'de ise savaş sonrası ekonomik yapıyı iyileřtirmek ve sosyal alanda iş programlarını geliřtirmek amacıyla kullanılmıştır. Bundan sonra bu çalışmalar endüstriyel ve ticari alanlarda özel řirketlere kadar uzanmıştır. Yazılım ara yüzlerinin içine kadar girerek kişisel kullanımlara kadar genişlemiştir. Birçok firma kendi yazılımlarını geliřtirmişlerdir. Microsoft hem MS Office Excel içine Solver eklentisini koyarak hem de MS Visual Studio için Microsoft Solver Foundation eklentisini geliřtirerek Yönetim Bilimine olumlu katkıları olmuştur. Microsoft yanında birçok firma da kendi yazılımlarını geliřtirmişlerdir. Gurobi Optimization firması kendi optimizasyon yazılımlarını geliřtirmişlerdir. Lindo Systems Inc. , lineer, lineer olmayan ve tamsayı programlama modelleme yazılımları üzerine çalışmıştır.

1.1.2.1 Matematiksel Modelleme Türleri

Yukarıdaki gelişmelerden sonra birçok yönetim bilimi modelleme teknięi geliştirilmiştir. Bu konu üzerinde çalışan akademisyenler farklı yönlerden tekniklere bakmışlar ve sınıflandırmalar yapmışlardır.

1.1.2.1.1 Ragsdale Modeli

Bu konuda çalışan Ragsdale (2008), yönetim bilimi modelleme tekniklerini 3 kısma ayırır. Ragsdale (2008)'in sınıflandırması ve model özellikleri Tablo 1’de verilmiştir.

Tablo 1
Yönetim Bilimi Modelleme Tekniklerinin Karakteristikleri ve Kategorileri

Sınıf	Model Özellikleri		
	Fonksiyon Oluşumu ($f(*)$)	Bağımsız Değişkenlerin Değerleri	Yönetim Bilimi Teknikleri
Kuralcı Modeller (Prescriptive Models)	Bilinmiyor, iyi tanımlanmış	Bilinmiyor veya karar vericinin kontrolü altında	Doğrusal Programlama, Şebeke Programlama, Tamsayı Programlama, CPM, Hedef Programlama EOQ, Doğrusal Olmayan Programlama
Kestirimci Modeller (Predictive Models)	Bilinmiyor, iyi tanımlanmamış	Bilinmiyor veya karar vericinin kontrolü altında	Regresyon Analizi, Zaman Serisi Analizi, Diskriminant Analizi
Tanımlayıcı Modeller (Descriptive Models)	Bilinmiyor, iyi tanımlanmış	Bilinmiyor veya kesin değil	Simülasyon, Sıralama, PERT, Envanter Modelleri

Kaynak: RAGSDALE, C.T. (2008), “Spreadsheet Modeling & Decision Analysis”

Kuralcı modeller (Prescriptive Models)

Bağımlı ve bağımsız değişkenler arasındaki ilişki iyi bilinen veya çok iyi fonksiyonel ilişki kurulabilen modellerdir. Bütün ilişkiler karar vericinin kontrolü altındadır.

Kestirimci modeller (Predictive Models)

Bağımsız değişkenler herhangi bir değer aldığında bağımlı değişkenin olması gereken değeri tahmin eden veya yaklaşık bir değer hesaplanmasını sağlayan modeldir. Bu model tarzında problemi fonksiyonel modellemek mümkün olmadığından bağımsız değişkenlerin bilinen değerlerinden yaklaşık fonksiyon üretilir.

Tanımlayıcı modeller (Descriptive Models)

Bağımsız değişkenler ile bağımlı değişkenler arasında tam bilinen bir ilişki vardır. Normal yaşamda bu model çok kullanılır. Bu modelde ilişkiler karar vericinin kontrolü altında değildir. Örneğin; eğer siparişlere bağlı olarak bir bağımsız değişkenimiz var ise ve bu bağımsız değişkene göre üretim yeri belirleyecek isek o zaman sipariş bağımsız

değişkeninin birçok değerine göre değişik alternatifler belirlememiz gerekir. Bu alternatifler arasından en uygun olanı belirlemek için bu model kullanılır.

1.1.2.1.2 Winston Modeli

Bu konuda çalışan Winston vd. (2003) ise yönetim bilimi modelleme tekniklerini dört kısma ayırmıştır.

Statik ve Dinamik Modeller

Winston vd. (2003), statik modeli, karar değişkenleri etkilerinin sonraki dönemlerde görülmeyeceği karar modeli olarak tanımlamışlardır. Bu tezin konuları içinde bulunan giriş stok sahasındaki bobinlerin enerji tasarrufu dolayısıyla en uygun sırada üretilmesi probleminin çözümü için de statik model kullanılmıştır. Verilen karar sonraki dönemlerde verilecek kararları etkilememektedir. Anderson vd. (2009), dinamik modellemeyi çözülemeyecek kadar büyük problemleri çözümü daha kolay daha küçük parçalara ayırarak problem çözme yöntemi yaklaşımı olarak tanımlamışlardır. İki şehir arasındaki en kısa yolu probleminin çözümü de dinamik modeldedir. Bir önceki seçim sonraki seçimleri direkt olarak etkilemektedir.

Doğrusal ve Doğrusal Olmayan Modeller

Amaç fonksiyonundaki veya kısıt fonksiyonlarındaki karar değişkenleri sabit sayılarla çarpılıp toplanıyor ise bu model doğrusal modeldir. Eğer amaç ve karar değişkenleri birbirleri ile çarpılıp toplanıyor ise bu model doğrusal olmayan modeldir.

Tamsayı ve Tamsayı Olmayan Modeller

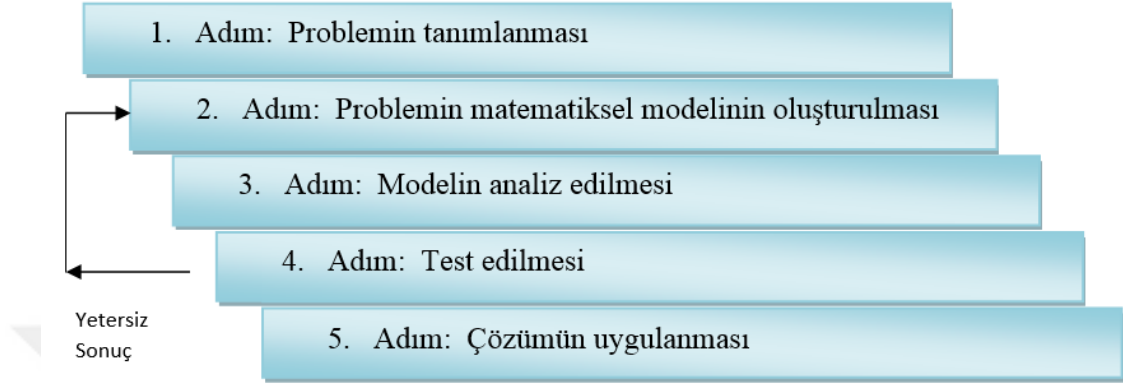
Eğer amaç ve kısıt fonksiyonlarındaki karar değişkenlerinden herhangi birinin tamsayı olma zorunluluğu var ise bu model tamsayı modeli, eğer bütün değişkenler tamsayı olmaya zorlanmıyor ise bu model tamsayı olmayan modeldir.

Deterministik ve Stokastik Modeller

Amaç ve kısıt fonksiyonları için ulaşılması gereken değerler rastlantısal durumlara bağlı değil ise bu model deterministik model aksi halde stokastik model denir.

1.1.3 Problem Çözme Süreci

Winston vd. (2003), bir organizasyon problemini çözmek için yöneylem araştırmacılarının Şekil 4’deki 5 adımını takip etmek zorunda olduğunu belirtmiştir.



Şekil 4. Problem Çözme Adımları

Kaynak: Winston,2003

1. Adım: Problemin tanımlanması:

İlk aşamada problemin farkına varılması lazımdır. Bu farkındalık raporlamanın sonucu olabilir ya da direkt olarak bir şikâyet üzerine ortaya çıkabilir. Problem ile ilgili karmaşıklığı çözebilmek için çok sayıda kişi ile görüşme yapmak ve problem hakkında çok bilgi toplamak gerekmektedir. Eğer problemin tanımını etkili bir şekilde yapamaz isek yeterli sonuçların alınamaması sonucu işimiz uzayacak, tekrar matematiksel modelin oluşturulması ile başlayan döngüye geri dönelecektir. Açıkça tanımlanamayan bir problem, çözümlenemeyen veya hiçbir zaman uygulanamayacak çözüm modellerini ortaya çıkarır.

2. Adım: Problemin matematiksel modelinin oluşturulması

Problemin iyi tanımlanması sonucu olarak problemin matematiksel modelini oluşturma işlemi başlar. Problemin tanımlanması ne kadar gerçeğe yakın olursa oluşturulacak modelde o kadar gerçeğe yakın olur. Tablo-1’den de görüleceği gibi birçok problem tekniği olmasına rağmen en iyi teknik modeli oluşturacak kişinin bildiği ve istediği tekniktir. O teknik uygun ise daha uygun teknik aramaya gerek yoktur.

3. Adım: Modelin analiz edilmesi

Modeli çeşitli karar değişkenleri ile test edip uygun alternatifler arasından seçim yapmamız gerekmektedir. Bu yol bize en uygun çözümü verecektir.

4. Adım: Test edilmesi

Modeli gerçek veriler ile test edilmesi ile uygunluğunun bir kez daha onaylanması anlamına gelmektedir. Bu süre zarfında gerçek hayatta daha değişik koşullar oluşmuş olabilir. Veya beklenmedik durumlar oluşabilir. Test sonucu yetersiz çıkar ise 2. adıma geri dönülmesi gerekmektedir.

5. Adım: Çözümün uygulanması

Çözümün uygulanması, modelden elde edilen çıktıların uygulanması aşamasıdır. Bu aşamada testler sırasında gözden kaçırılan hususlar var ise yanlış kararlar alınmasına neden olunabilir. Bu aşamada parametrelerin eksiksiz olduğundan, modelin tam kurulduğundan ve kısıtlamaların tam tanımlandığından emin olmak gerekir. Ayanoğlu (2006) , “Yanlış bir modelden elde edilen sonuçların uygulamaya konulması çok tehlikeli olaylara sebep olur.” demiştir. Problemi algılayıp modeli oluşturup test ettiğiniz çözüm sonucunda insan uygulaması var ise değişime karşı dirençten dolayı insanlara bu çözümü uygulatmak da zor olabilir.

1.1.4 Doğrusal ve Doğrusal Olmayan Programlama

Belirli sınırlamaları sağlayacak şekilde bilinmeyen parametre değerlerinin bulunmasını içeren herhangi bir problem optimizasyon problemi olarak adlandırılır (Karaboğa, 2011). Bir optimizasyon işleminin 1. adımı olan problemin tanımlanması aşamasında karar değişkenleri olan hedef fonksiyonunun ve problem ile ilgili kısıtların (sınırlılıkların) tanımlanması gerekmektedir. Bu amaç fonksiyonu minimize edilecek bir maliyet fonksiyonu veya maksimize edilecek bir kâr fonksiyonu olabilir. Maliyet fonksiyonu, daha iyi çözümü temsil eden parametre değerlerinin kullanılması durumunda daha düşük bir nümerik değer üretirken; kâr fonksiyonu ise daha yüksek bir nümerik değer üretmektedir. Bazı sınırlamalar eşitsizlikler bazıları da eşitlikler biçiminde olabilir. Bazı durumlarda problemin birden fazla maliyet fonksiyonu olabilir. Bu tür problemler ise çok hedefli karar verme veya çok hedefli optimizasyon problemi olarak adlandırılır. Bu tür problemlerin çözümü amacıyla problemin modellenmesi için birçok teknik olmasına

rağmen burada çok hedefli ve doğrusal olmayan modelleme tekniği üzerinde durulacaktır.

Yönetim Bilim’inde en yaygın kullanım alanı bulan tekniklerden bir tanesi olan Doğrusal Programlama (DP), doğrusal karar modelleriyle ilgili kavram ve teknikler bütünüdür (Bayraktar ve Çebi, 2003). Doğrusal programlama, bütün model parametrelerinin kesin olarak bilindiğini varsayan deterministik bir tekniktir (Bayraktar ve Çebi, 2003).

Doğrusal programlama, “Yöneylem Araştırmalarında en yaygın kullanım alanı bulan tekniklerden bir tanesi olan doğrusal programlama doğrusal karar modelleri ile ilgili kavram ve teknikler bütünüdür. Bütün model parametrelerinin kesin olarak bilindiğini varsayan deterministik bir tekniktir. ” diye tarif edilmiştir (Bayraktar ve Çebi, 2003). Bir optimizasyon modeli eğer sürekli değişkenlere ve tek bir doğrusal amaç fonksiyonuna sahipse ve tüm kısıtlamaları doğrusal eşitlik veya eşitsizliklerden oluşuyorsa, doğrusal (lineer) program olarak adlandırılır. Başka bir deyişle, modelin tek hedefli fonksiyonu ve tüm kısıtlamaları, süreklilik gösteren karar değişkenlerinin ağırlıklı toplamlarından oluşmalıdır (Bayraktar ve Çebi, 2003).

Bir doğrusal programlama problemi üç bölümden oluşur:

- 1- Bir DP problemi, karar değişkenlerinin ($X_1, X_2, \dots, X_n$) doğrusal bir fonksiyonu olan amaç fonksiyonu içerir. Amaç fonksiyonu maksimizasyon ya da minimizasyon hedefli olabilir (Bayraktar ve Çebi, 2003).
- 2- Bir DP problemi, karar değişkenlerinin alacağı değerleri sınırlayan kısıtlar seti içerir. Her bir kısıt seti doğrusal eşitlik ya da eşitsizlik şeklinde ifade edilmelidir (Bayraktar ve Çebi, 2003).
- 3- Bir DP problemi, karar değişkenlerinin negatif olmama gerekliliğini belirleyen bir kısıt içerir (Örnek: $X_j \geq 0, (j=1, \dots, n)$) (Bayraktar ve Çebi, 2003).

Gerçek hayatta karşılaşılan çoğu problem için geliştirilen karar modellerinin kısıtlarında ve amaç fonksiyonunda doğrusal ilişkileri gözlemek zordur. Karar modelinin kısıtlarından en az biri veya amaç fonksiyonunun doğrusal olmadığı durumlar için geliştirilen kavram ve teknikler “Doğrusal Olmayan Programlama (DOP) ” başlığı altında incelenmektedir (Aras,2012).

Doğrusal olmayan karar modelinin genel yapısı aşağıdaki gibidir:

X karar değişkenleri vektörü, $X = (X_1, X_2, \dots, X_n)$

$g_i(x)$: i. kısıtın ifadesi ($i = 1, 2, \dots, m$)

b_i : i. kısıtın sağ taraf sabiti ($i = 1, 2, \dots, m$),

$f(x)$: Amaç fonksiyonu ve en az bir $g_i(x)$ ve/veya $f(x)$ doğrusal olmayan vektör fonksiyonları olmak üzere; $f(x)$ fonksiyonunu optimal X vektörünün bulunması ile modelin çözümü gerçekleştirilmiş olur.

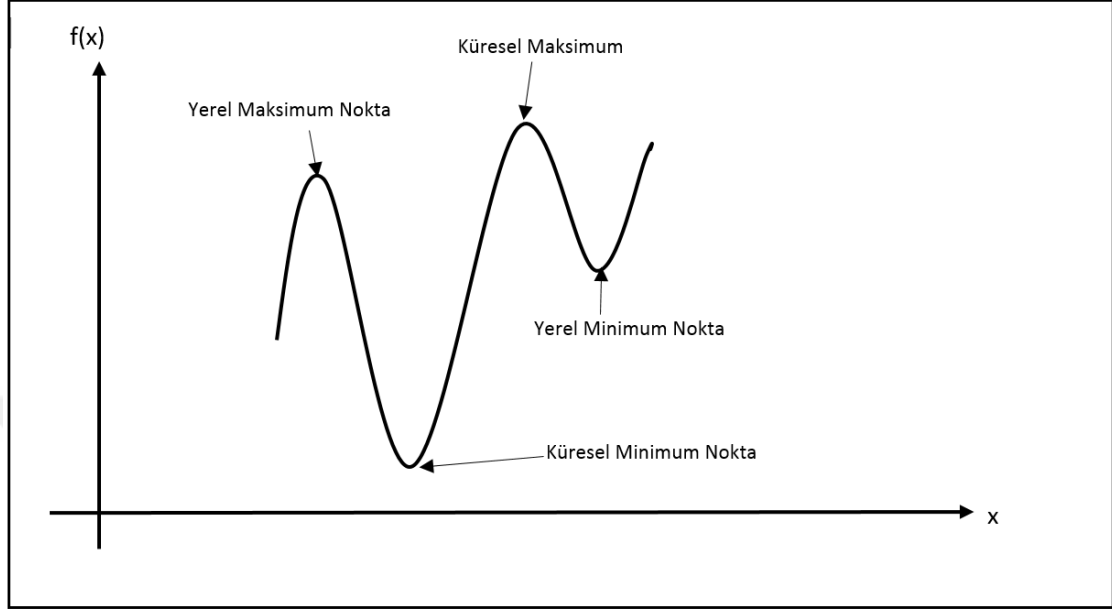
Eğer amaç fonksiyonu dış bükey (konkav (maksimizasyon problemi)) fonksiyon veya iç bükey (konveks (minimizasyon problemi)) fonksiyon ve sınırlılıklar seti konveks fonksiyon ise konveks problem olarak adlandırılır ve konveks optimizasyon genel kuralları bu durumlarda uygulanır. Bu nedenle maksimizasyon probleminde amaç fonksiyon içinde küresel maksimumu yakalamak, minimizasyon probleminde ise küresel minimumu yakalamak olacaktır. Bazı çözüm tekniklerinin yerel minimumları veya yerel maksimumları yakalaması sonucu problemin çözümü tam olarak elde edilememektedir.

1.1.5 Doğrusal Olmayan Programlamanın Çözüm Yöntemleri

Matematiksel modelleme tekniği öncelikle doğrusal ve az sayıda değişkenlerin kullanıldığı doğrusal programlamanın kullanıldığı modeller ile başlamıştır (Yılmaz, 2008) . Bir süre sonra doğrusallık varsayımının her problem için geçerli olmadığı anlaşılmış, bu durumda doğrusal olmayan modelleme yöntemi bulunmuştur. Ancak doğrusal olmayan modellerin kendine özgü çözümleri uygulamada birçok sorunu beraberinde getirmiştir. Bu sorunlar küresel optimum nokta yerine yerel optimum noktalarda takılı kalmaktır. Yerel minimumlar veya maksimumlar yerine küresel minimumlar veya maksimumları

bulmak matematikte önemli bir konudur. Zamanla geliştirilen bazı yöntemler doğrusal olmayan modellerin hızla gelişmesine neden olmuştur.

Şekil 5’de küresel ve yerel minimum noktalar gösterilmiştir.



Şekil 5. Bir Fonksiyonun Yerel Ve Küresel Minimum Noktaları

Kaynak: Akçayol (2015), “Zeki Optimizasyon Teknikleri

Gerçek hayatta karşılaşılan birçok problem için geliştirilen karar modellerinin kısıtları vardır ve amaç fonksiyonları her zaman doğrusal bir ilişki kurulamazlar. Bu nedenlerle 1950’li yıllardan sonra geliştirilmeye başlayan ve temelleri 18’inci ve 19’uncu yüzyıllara dayanan yeni analitik ve sayısal yöntemler 1960’lı yıllardan sonra sayısal bilgisayarlarında desteği ile hızla çoğalmıştır.

Klasik optimizasyon teorisi Cauchy, Lagrange ve Newton tarafından geliştirilmiştir. Newton ve Leibnitz ‘in analiz çalışmaları optimizasyonun diferansiyel hesap metotlarının geliştirilmesine katkıda bulunmuştur. Fonksiyonların maksimum ve minimumları için olan Euler-Lagrange denklemlerinin türetilmesi ile ortaya çıkan değişkenler kalkülüsünün (matematiksel analiz) yaratıcılarındandır. Kısıtsız optimizasyon problemlerini çözmek için en dik iniş (Steepest Descent) metodunun ilk uygulaması da Cauchy tarafından yapılmıştır.

Doğrusal Olmayan Programlama konusundaki ilk önemli çalışmalar 1951 yılında Kuhn ve Tucker tarafından optimal çözüm için gerek ve yeter şartlar teorisi başlığı adı altında sunulmuştur (Hillier ve Lieberman,1990).

Gerek şart: Optimum noktada (minimum veya maksimum nokta) şartları sağlaması gereken durumlar olarak adlandırılır (Hillier ve Lieberman,1990). Diğer bir tanımla eğer herhangi bir nokta gerek şartları sağlamıyorsa optimum nokta olamaz (Hillier ve Lieberman,1990). Bununla birlikte gerek şartları sağlayan nokta optimum olmayabilir veya tek bir nokta olmayabilir. Gerek şartları sağlayan noktalar aday nokta (candidate points) olarak adlandırılır (Hillier ve Lieberman,1990). Dolayısıyla optimum nokta ile optimum olmayan noktaları ayırmak için başka şartlara ihtiyaç duyulur ve bu şartlar yeter şart olarak adlandırılır (Hillier ve Lieberman,1990).

Yeter şart: Eğer aday optimum noktalarda yeter şartları sağlıyorsa bu nokta optimum noktadır ve daha ileri testler yapmaya gerek yoktur (Hillier ve Lieberman,1990). Ancak bu şartların sağlanamadığı veya kullanılmadığı durumlarda aday noktalarından herhangi birisinin optimum olmadığı söylenemeyebilir (Hillier ve Lieberman,1990)..

Aşağıdaki optimizasyon problemi için Kuhn-Tucker şartlarını araştıralım (Hillier ve Lieberman,1990):

$$\min f(x)$$

Kısıtlar:

$$h_i(x) = 0 \quad i = 1, 2, \dots, p$$

$$g_j(x) \leq 0 \quad j = 1, 2, \dots, m$$

$$x = [x_1, x_2, \dots, x_n]^T$$

Bu problem için Lagrange fonksiyonu aşağıdaki gibi yazılır.

$$L(x, u, v, s) = f(x) + \sum_{i=1}^p v_i h_i(x) + \sum_{i=1}^m u_i (g_i(x) + s)$$

v, u : Lagrange çarpanı

s : artık değişken

Bu Lagrange fonksiyonu için Kuhn-Tucker şartları şöyledir.

$$\frac{\partial L}{\partial x_j} = \frac{\partial f}{\partial x_j} + \sum_{i=1}^p v_i^* \frac{\partial h_i}{\partial x_j} + \sum_{i=1}^m u_i^* \frac{\partial g_i}{\partial x_j} = 0 \quad j = 1, 2, \dots, n$$

$$h_i(x^*) = 0 ; \quad i = 1, 2, \dots, p$$

$$g_i(x^*) + s_i^2 = 0 ; \quad i = 1, 2, \dots, m$$

$$u_i^* s_i = 0 ; \quad i = 1, 2, \dots, m$$

$$u_i^* \geq 0 ; \quad i = 1, 2, \dots, m$$

Bu şartlar aynı zamanda 1. derece gerek şartlar (first - order conditions) olarak da adlandırılır. Bu denklemlerin x^* düzenli nokta denilen özel bir noktalarda değerlendirilmesi gerekmektedir (Hillier ve Lieberman,1990).

Kuhn - Tucker gerek şartlarının kullanım amaçları şunlardır (Hillier ve Lieberman,1990).

- Bir noktanın muhtemel optimum olup olmadığını kontrol etme
- Aday minimum noktaların tespiti

Doğrusal olmayan optimizasyon metotları, deterministik metotlar ve istatistiksel metotlar olmak üzere iki ana gruba ayrılabilir (Hillier ve Lieberman,1990). Deterministik optimizasyon metotları, lokal maksimum veya minimuma yakınsayan algoritmalarlardır. Türevsel hesaplamalar veya türevsel yaklaşımlar deterministik metotlara örnek verilebilir. İstatistiksel metotlar ise küresel minimum veya maksimumu bulmada bazı stratejileri ve rasgele sayıları kullanırlar (Hillier ve Lieberman,1990). Son yıllarda kişisel bilgisayarların hızlarındaki artış bu algoritmaların uygulama sahasında sıkça görülmesine neden olmuştur.

Optimizasyon metotlarının çözüm mantığına göre sınıflandırılması Tablo 2’de verilmiştir.

Tablo 2
Optimizasyon Metotlarının Çözüm Mantığına Göre Sınıflandırılması

Direkt Metotlar	Steepest descent, conjugate gradient, gradient projection, feasible directions, Newton, penalty, simplex
Sezgisel (heuristik) Metotlar	Rasgele metot, Simulated Annealing, Genetik algoritma
Yaklaşık Metotlar	Sequential linear programming, Sequential quadratic programming, response surface metot

Kaynak: Öztemel, 2012

Dinamik sistemlerin kararlı denge noktalarının bulunması yaklaşımı ile doğrusal olmayan optimizasyon (doğrusal olmayan programlama) problemlerinin çözümü konusundaki teknikler de direkt yöntemlerden sayılmaktadır. Söz konusu çalışmalarda, optimizasyonu yapılacak doğrusal olmayan fonksiyonun gradyanı kullanılarak birinci mertebeden adi diferansiyel denklem yardımıyla bir dinamik sistem tanımlanmakta ve bu dinamik sistemin denge noktaları, doğrusal olmayan fonksiyonun yerel optimumları olarak bulunmaktadır. Diğer taraftan ikinci mertebeden adi diferansiyel denklem üzerine kurulmuş dinamik sistem yaklaşımı üzerine de araştırmalar yapılmıştır (Hacıoğlu (2011)).

Tek Değişkenli Fonksiyonlar için Optimizasyon

Tek değişkenli fonksiyonlarda dikkat edilecek husus elde edilen minimum değerın yerel minimum mu yoksa küresel minimum mu olduğunun tespit edilmesidir (Hacıoğlu (2011)). Bir değişkenli bir $f(x)$ fonksiyonun h 'ın küçük pozitif ve negatif değerinde aşağıdaki ifadeyi veriyorsa bu fonksiyon $x = x^*$ 'da lokal minimumdur (Karaboğa, 2011).

$$f(x^*) \leq f(x^* + h)$$

Benzer olarak x^* noktasında eğer aşağıdaki ifade sağlanıyorsa bu değerde $f(x)$ fonksiyonu maksimumdur (Karaboğa, 2011).

$$f(x^*) \geq f(x^* + h)$$

Küresel minimum veya maksimum değer için en uygun değer şartlarının sağlanması gerekmektedir ki bir değişkenli bir fonksiyon için aşağıda verilmiştir.

Optimumluk şartları, bir $f(x)$ 'nın aday noktalarını belirlemede kullanılır. Optimumluk şartlarını elde etmek için öncelikle aşağıda verilen kabul yapılır (Hacıoğlu (2011)): “ x

minimum noktadır ve bunun civarındaki bir noktada fonksiyonun değeri ve türevi dikkate alınacaktır.” x^* noktası fonksiyonun yerel minimum noktası olsun. x ise x^* noktasına yakın herhangi bir nokta olarak dikkate alalım. Dolayısıyla artım miktarı d aşağıdaki gibi tanımlanır (Karaboğa, 2011):

$$d = x - x^*$$

ve bu noktalara karşılık $f(x)$ fonksiyonun farkı aşağıdaki gibi verilir (Karaboğa, 2011):

$$\Delta f(x) = f(x) - f(x^*)$$

x^* noktası $f(x)$ fonksiyonun yerel minimum olduğu nokta olduğundan küçük bir ilerlemede (x değerine varıldığında) $f(x)$ 'in değeri değişmez veya mutlaka artar. Dolayısıyla $\Delta f(x)$ negatif olmayan bir değer alır. Dolayısıyla,

$$\Delta f(x) = f(x) - f(x^*) \geq 0$$

olmalıdır. Buradan $f(x)$ fonksiyonu Taylor serisi ile açıldıktan sonra sonuç olarak d 'nin her değerinde şartı sağlayan değeri:

$$f'(x^*) = 0$$

olacaktır (Karaboğa, 2011).

Bu şarta birinci-derece optimumluk şartı (first order optimality condition) veya birinci-derece gerek şart (first-order necessary condition) olarak adlandırılır (Hacıoğlu (2011); zira fonksiyonun sadece birinci türevini içerir. Bu şartları sağlayan noktalar yerel minimum veya maksimum veya hiçbirisi olmayabilir (büküm noktası olabilir) (Hacıoğlu (2011). Bu noktalar istasyon (stationary) noktaları olarak adlandırılır. Aday noktaları belirledikten sonra bu noktalardan hangisinin fonksiyonu minimum veya maksimum yaptığını belirlemek için yeter şartlar dikkate alınır. Buna göre;

$$f''(x^*) \geq 0 \text{ bu nokta minimum,}$$

$$f''(x^*) \leq 0 \text{ bu nokta maksimum,}$$

$$f''(x^*) = 0 \text{ daha ileri türveleri araştırılmalıdır.}$$

Bu şartlar yeter şart olarak adlandırılır (Hacıoğlu (2011).

Cok Değişkenli Fonksiyonlar için Optimizasyon

$f(x)$ maliyet fonksiyonu, $x^* = (x_1^*, x_2^*, \dots, x_n^*)$ çözümü için Denklem (1) ile verilen

şartı uygun bölge (S) içindeki tüm muhtemel çözümler (x) için sağlıyor ise f(x) fonksiyonu x^* 'de küresel minimuma sahiptir denir.

$$f(x^*) \leq f(x) \quad (1)$$

Eğer Denklem (1) uygun bölge içerisinde, x^* 'in dar bir komşuluğu (N) içindeki tüm muhtemel çözümler için sağlanıyorsa f(x) fonksiyonu x^* 'de yerel minimuma sahiptir denir. x^* 'in komşuluğu (N), küçük bir değere sahip olan δ parametresine bağlı olarak

$$N = \{ x \mid x \in S \text{ ve } \|x - x^*\| < \delta \} \quad (2)$$

şeklinde tanımlanabilir. Bu, geometrik olarak x^* noktasını ihtiva eden küçük uygun bir bölgeyi tanımlamaktadır.

$f(x)$ fonksiyonunun x^* noktasındaki x_1 'e göre kısmi türevi $\partial f(x) / \partial x_1 \mid_{x=x^*}$ şeklinde tanımlanır. Şayet tüm kısmi türevler bir sütun vektörü formunda düzenlenirse, vektör gradyent vektörü olarak adlandırılır ve ∇f veya grad f ile gösterilir. Geometrik olarak gradyent vektörü, x^* noktasına teğet olan $f(x) = \text{sabit}$ düzlemine diktir ve $f(x)$ fonksiyonunun en hızlı artış gösterdiği yönü belirtmektedir. Gradyent vektörünün bu özelliği, klasik gradyente dayalı optimizasyon algoritmalarının temelini teşkil etmektedir.

Gradyent vektörün ikinci kez türevini almak suretiyle klasik optimizasyonda önemli bir yeri olan Hessian matris H (Şekil 6) elde edilir. Dolayısıyla bu matris, f(x) fonksiyonun ikinci dereceden kısmi türevlerinden oluşmaktadır. Hessian matris simetrik özelliğe sahiptir. Optimallik için yeterli şartların tayininde önemli rol oynar.

$$H(f) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

Şekil 6. Hessian Matrisi

Kaynak: Research hubs, Hessian Matrix, 2017

x^* , $f(x)$ fonksiyonunun yerel minimumu olan bir nokta ve x, x^* 'in yakın civarında bir nokta olsun. Bu durumda, çözümlerin x^* civarındaki değişimi d ve bundan dolayı amaç fonksiyonunda ortaya çıkan $f(x^*)$ civarındaki değişimi Δf sırasıyla,

$$d = x - x^* \text{ ve } \Delta f = f(x) - f(x^*)$$

olarak tanımlanabilir. x^* yerel minimum olduğu için yakın civarında, yani küçük d değerleri için aşağıdaki eşitsizliğin sağlanması gerekir:

$$\Delta f = f(x) - f(x^*) \geq 0$$

Bu eşitsizlik, yerel minimum için gerek ve yeter şartların türetilmesi amacıyla kullanılır. d oldukça küçük olduğundan Δf , x^* noktasında Taylor serine açılarak optimallik şartları türetilir. Tek değişkenli fonksiyon için birinci dereceden gerek şart

$$f'(x^*) = 0$$

dır. Ancak bu şartı hem maksimumlar hem de minimumlar, yani değişimin sıfır olduğu durağan noktalar sağlamaktadır. Minimumluk için yeter şart

$$f''(x^*) \geq 0$$

dır. Eğer, $f'(x^*) = 0$ ise, x^* 'in bir minimum nokta olmadığı sonucuna varılamaz. Bu durumda daha yüksek dereceden türevlerin hesaplanması gerekir. İkinci dereceden gerek şart ise

$$f''(x^*) > 0$$

ile tanımlıdır ve $f'(x^*) < 0$ şartını sağlayan herhangi bir nokta **yerel minimum** olamaz, $f'(x^*) = 0$ ise, bu noktanın **yerel minimum** olabilmesi için

$$f''(x^*) = 0$$

gerek şartı ile

$$f''(x^*) > 0$$

yeter şartını sağlaması gerekir. Bu türev sıfıra eşitse yine şüpheli durum söz konusudur ve bir üst dereceden türeve bakılmalıdır. Kısacası çift-derece türev sıfırdan küçükse aday nokta kesinlikle **yerel minimum** şartını sağlamıyor demektir ama sıfıra eşitse bir üstteki-dereceli türevin sıfıra eşit ve bir üst çift-derece türevin ise sıfırdan büyük olup olmadığı kontrol edilmelidir.

Yukarıda bahsedildiği gibi, doğrusal olmayan optimizasyon problemlerinde fonksiyonların bir kısmı veya bütünü doğrusal değildir. Doğrusal olmayan programlama (DOP) kavramı, doğrusal olmayan problemleri çözmek amacıyla geliştirilen nümerik metotlar için kullanılmaktadır. Bu tür problemler için nümerik metotlara ihtiyaç duyulmasının ana sebebi, bazı problemlere analitik metotlarının uygulanmasının zor olması hatta bazen mümkün olmamasıdır. Analitik metotlarda, gerekli şartlar yazılır ve bunlar aday **yerel minimum** noktalar için çözülür. Arora (2004), doğrusal olmayan programlamaya ihtiyaç duyulmasının sebeplerini aşağıdaki gibi sıralamıştır:

- Tasarım değişkenlerinin ve sınırlamaların sayısının çok fazla olması durumunda oldukça fazla sayıda denklem oluşabilir ve dolayısıyla gerekli şartların çözülmesi çok zor olabilir. Bu durum özellikle sınırlamalı optimizasyon problemlerinde ortaya çıkmaktadır.
- Tasarım probleminin fonksiyonları yüksek seviyede doğrusal olmama içerebilir. Bu yüzden problemin boyutu küçük olsa bile gerekli şartlar da oldukça doğrusal olmayan yapıda bulunabilir.
- Çoğu mühendislik uygulamalarında fonksiyonlar (maliyet ve sınırlama) tasarım değişkenleri açısından açık olamayabilir yani bağıntılar net şekilde kurulamayabilir. Bu yüzden mühendislik problemlerinin çözümü ve mühendislik optimum tasarımları için sayısal yaklaşımlar geliştirilmiştir. Bu tür yaklaşımlarda bir başlangıç çözümü alınır ve bu çözüm iteratif olarak optimallik şartları sağlanıncaya kadar geliştirilir (Tablo 3).

Tablo 3
Doğrusal Olmayan Optimizasyon için Çözüm Algoritması

1.Adım	x^* noktasında başlangıç çözümü al.
2.Adım	Araştırma uzayında bu nokta için araştırma yönünü belirle. (Amaç fonksiyonu, gradyentinin ve varsa sınırlama fonksiyonlarının hesaplanması gerektirir.)
3.Adım	Araştırmanın yakınsamasını kontrol et. Belirlenen değerin altında ise döngüyü bitir.
4.Adım	Adım büyüklüğünü hesapla.
5.Adım	Yeni çözüm noktasını hesapla. ($x^{(t+1)} = x^t + \alpha^{(t)} * d^{(t)}$)
6.Adım	Döngü sayacını arttır. Ve 2. adıma dön.

Kaynak: KARABOĞA D. (2011), “Yapay Zeka Optimizasyon Algoritmaları”, Nobel Kitap Dağıtım, İstanbul

Araştırma yönü gradyan bilgisine bağlı olarak belirlendikten sonra adım büyüklüğünün hesaplanması gerekir. Adım büyüklüğünün belirlenmesi için kullanılan metotlar en genel şekli ile iki gruba ayrılır: Analitik ve Nümerik metotlar. ,

Yaygın olarak kullanılan nümerik metotlar (Karaboğa,2011):

- Eşit aralıklı araştırma (equal interval search)
- Altın Bölüm Arama (Golden Section Search)
- Polinom İnterpolasyonu

1.1.6 Çok Hedefli Programlama

Hedef programlama bir optimizasyon programıdır. Hedef programlama, belirli bir objektif fonksiyonu değil, daha çok ulaşmak istediğimiz hedefler içeren problemleri çözmeyi içerir. Çok hedefli optimizasyon, hedef programlama ile yakından ilişkilidir ve birden fazla hedef fonksiyon içeren problemler için geçerlidir (Ragsdale, 2008). Hedef programlama aslında geliştirilmiş bir doğrusal veya doğrusal olmayan programlamadır. Tek bir hedef yoktur. Birçok hedef vardır ve bu hedeflerin karar vericinin atadığı önem

sırasına göre karşılanması gerekmektedir. Öncelik düzeyi yüksek olan hedeften başlayarak hedef düzeyi düşük olan hedefe doğru hedeften sapmaların minimize edilmesine çalışılır (Sezer ve Coşkun, 2016). Başarılı bir hedef programlama karar vericinin ihtiyaçlarını karşılamalıdır. Bir başka yaklaşım ise çok sayıdaki amacın tek bir amaca dönüştürülmesidir. Problem çok hedefli programlamadan tek hedefli problem haline dönüştürülerek çözülür. Gerçek hedef problemleri genelde çok hedefli problemlerdir. Elimizdeki kısıtlı bütçeler ile birçok ihtiyacı karşılamaya çalışır fakat sonunda hepsini karşılayamayız. Bu durumda en önemli harcamalardan başlayarak daha az öncelikli harcamalara yöneliriz.

Hedef programlamada ana düşünce, orijinali çok hedefli olan problemi tek hedefli probleme dönüştürmektir. Hedef programlamada belirlenen hedeflerin hepsinin model formülasyonuna dâhil edilmesi gerekir. Çözüm sonucu belirlenen hedeflerden herhangi bir sapma olup olmadığı belirlenerek eğer belirlenen hedeflerden bir sapma olmuş ise eldeki kısıtlara uygun olarak bu sapmalar minimize edilmeye çalışılır. Böylece belirli kısıtlar altında hedefleri sağlayan en uygun çözüm bulunmaya çalışılır.

Hedef programlama ilk olarak 1955 yılında Charnes, Cooper ve Ferguson tarafından kullanılmıştır (Charnes, Cooper, Ferguson (1955)). Fakat bu adıyla ilk defa 1961 yılında Charnes ve Cooper'ın makalesinde görülmüştür (Charnes ve Cooper (1961)). İlk mühendislik uygulaması ise aya ilk insanı indiren Apollo uzay kapsülünün aya konuşturulmasıdır (Sezer ve Coşkun, 2016).

1965 yılında Ijiri tarafından “genelleştirilmiş ters alma tekniği” ile çözüm elde edilebilen daha kullanışlı bir hale getirilmiş, 1968 yılında da Contini hedef programlamayı belirsizlik durumlarına, Jaaskelainen ise toplu üretim planlamasına uyarlamıştır. Daha sonraları hedef programlama akademik, finans, işletme yönetimi, medya gibi birçok değişik dalda uygulama alanı bulmuştur. Bunların çoğu Sang Lee'nin “Goal Programming for Decision Analysis” isimli kitabı ve Lee ve Jaaskelainen'in geliştirdiği bilgisayar programları sayesinde gerçekleştirilmiştir (Wu ve Coppins (1981)).

Hedef programlama maksimizasyon ve minimizasyon problemlerinden farklı olarak belirlenen hedeflerden en az sapmalı çözümü bulmaya yöneliktir (Malczeswki (1999)). Hedef programlama modelinde optimizasyon düşüncesinden daha çok, karar vericinin

doyurucu bulduğu bir çözüm belirlenmeye çalışılır. Hedef programlama, çok sayıda hedef veya amaçların bulunduğu doğrusal programlama problemlerine uygulanan bir yöntemdir. Doğrudan amaçları optimize eden doğrusal programlamanın aksine, hedef programlama, hedef değerler ve gerçekleşmiş sonuçlar arasındaki sapmaları minimize ederek, çatışan amaçları yönetmek amacıyla kullanılır (Stephen ve diğerleri, 2003).

Doğrusal programlama problemlerinde amaç fonksiyonu birim açısından tek bir ölçekle belirtilir. Bu verimlilik, maliyet, ağırlık, hacim gibi tek bir ölçüm çeşidi olabilir. Bundan dolayı çok boyutlu bir ölçek kümesi ile ifade edilebilen çok hedefli bir doğrusal programlama problemini yazmak mümkün değildir. Bu nedenle de çok hedefli problemlerin çözümünde hedef programlama yöntemine ihtiyaç duyulmuştur. Bütün işletmelerin bir takım hedefleri vardır. Bu hedefler işletmeden işletmeye değişebilen; bazı işletmeler için haftalık minimum veya maksimum üretim miktarı ve üretim maliyeti gibi basit hedefler olabileceği gibi bazı işletmeler için de birbirinden bağımsız birden fazla hedefin birleşmesinden meydana gelebilecek karmaşık hedefler de olabilmektedir.

Çok hedefli karar problemini hedef problemi haline getirmek amacıyla sayısal öncelikler tanımlamamız gerekmektedir. Her bir hedefe öncelik sırası veya puanı atayarak problem hedef programlama optimizasyon problemi haline getirilir ve çözüme kavuşturulur. Bu çözümde amaç amaçlar ve önceliklere bakarak en uygun çözümü bulmaktır. Birebir çözüm bulma olasılığı çok düşüktür. Sonuçta hedeflerden sapmalar, yani bir hedefin beklenenin üzerinde gerçekleşmesi (+) pozitif sapma ve hedefin altında kalınması (-) negatif sapma değerleri toplamının minimize edilmesi bir tek amaç olarak ortaya konulur.

Hedef programlamanın temel ilkelerini şöyle özetleyebiliriz (Timor(2001));

1. Hedef programlama için her bir amaç bir hedef olarak kabul edilir.
2. Hedef programlamada hedeflerin gerçekleştirilmesinde öncelikler dikkate alınır. Önce birinci öncelik düzeyindeki hedefler daha sonra ikinci öncelik düzeyindeki hedefler gerçekleştirilir. Sıra atlamadan bütün hedefler tamamlanana kadar devam edilir.
3. d_i^- i. hedefin altında kalınması durumunu, d_i^+ hedefin aşılması durumunu gösterir. Bu hedeflerin ikisi aynı anda amaç fonksiyonuna sokulamaz.

4. Hedef düzeyleri dikkate alınarak hedeflerden toplam sapma minimize edilmeye çalışılır.
5. Hedef programlama modelinde amaç fonksiyonu, karar vericinin istekleri, sınırlı kaynaklar ve kontrol değişkenleri üzerine konulan kısıtlama koşulları göz önünde bulundurularak oluşturulur. Hedef programlama modelinde amaç fonksiyonları, karar değişkenlerinin matematiksel bir fonksiyonu olarak gösterilebilir.

Doğrusal olmayan hedef programlama modeli için bir takım tanımlamalar mevcuttur. Genel olarak kullanılan bu tanımlamalar aşağıda verilmiştir (Sarıay(2012)).

Karar Değişkeni: Bütün karar değişkenleri problemi tanımlar ve karara şekil verirler. Karar vericilerin belirlemek istediği değişkenlerdir. Hedef programlama modelinin amacı, karar verme sürecinde hedeflerin ve kısıtların en iyi şekilde tatmin olduğu noktayı yani, karar değişkeni değerlerinin mümkün olan bütün kombinasyonlarının bir araştırması olarak görülebilir.

Kriter: Karar vericilerin sistem tasarımını oluşturmada kullandıkları değerlerdir. Kriter değerlendirme yapmak için esasın ve etkinliğin ölçüsüdür. Kriterler problemin yapısı içerisinde amaçların bir formu olarak ortaya çıkar (Zeleny 1982: 17).

Amaç Fonksiyonu: Karar verici tercih ettiği niteliklerin hangi seviyede maksimizasyon veya minimizasyon yapacağını kendi istek ve ihtiyaçları doğrultusunda kararlaştırır. Ayrıca sistem performansını açıklayan ve sistemin gelişim yönünü gösteren başarı seviyeleridir.

Hedef: Karar vericinin ihtiyaçları ve arzu ettiği seviyeler olarak tanımlanabilir. Hedefler tam olarak gerçekleştirildiği gibi hedefe yakın değerler olarak da ortaya çıkabilirler.

Sapan Değişken: Başlangıçta verilen hedef ile çözüm sonucunda gerçekleşen değerler arasındaki farkları gösterir. Bu farklar pozitif sapma ve negatif sapma şeklinde tanımlanır. Hedeflerin eşitsizlik yönüne bağlı olarak istenmeyen sapma ya da istenen sapma kavramları da kullanılabilir.

Kısıtlar: Kısıtlar sistem kısıtları ve hedef kısıtlar olarak iki grupta incelenir. Sistem kısıtları tam olarak sağlanması gereken ve hiçbir sapmaya izin verilmeyen kısıtlayıcılarıdır. Hedef kısıtlar sistem kısıtlarına göre daha esnek bir yapıya sahip olup arzu edilen hedef değerlerini gösteren fonksiyonlardır (Öztürk 2004: 292).

Hedef Programlamanın matematiksel modeli aşağıdaki gibi ifade edilir (Charnes ve Cooper 1977).

Amaç Fonksiyonu:

$$\text{Min/Max } Z = \sum_{i=1}^m (d_i^+ + d_i^-)$$

Kısıtlar:

$$\sum_{j=1}^n a_{ij}x_j - d_i^+ + d_i^- = b_i$$

$$d_i^+ \cdot d_i^- = 0$$

$$d_i^+, d_i^-, x_j \geq 0 \quad i = 1, 2, \dots, m \quad j = 1, 2, \dots, n$$

Hedef Programlamada yer alan $d_i^+ \cdot d_i^- = 0$ kısıtı her bir hedefte yer alan sapan değişkenler aynı anda meydana gelemeyeceğinden sapan değişkenlerin en az bir tanesinin veya her ikisinin de sıfır olması gerektiğini göstermektedir. İstenmeyen sapan değişkenlerin belirlenmesinden sonra Hedef Programlama formülasyonu yapılır. Bu değişkenler içerisinde yalnızca bir tanesi karar verici tarafından minimize yapılmak istenir. Bu düşüncüyü aşağıdaki üç formda yapmak mümkündür (Romero 2001).

1. $f(x) \leq b_i$; Eğer i-inci hedef belirlenen başarı düzeyinden küçük veya eşit ise oluşacak pozitif sapan değişken d_i^+ için mümkün olan en küçük pozitif değerin alınması gerekir.
2. $f(x) \geq b_i$; Eğer i-inci hedef belirlenen başarı düzeyinden büyük veya eşit ise oluşacak negatif sapan değişken d_i^- için mümkün olan en küçük pozitif değerin alınması gerekir.
3. $f(x) = b_i$; i-inci hedef belirlenen başarı düzeyini tam olarak karşılıyor ise hem pozitif sapan değişken d_i^+ hem de negatif sapan değişken d_i^- 'nin toplamalarının aynı anda minimize yapılması gerekir.

Minimizasyon süreci, hedef programlamanın farklı metotlarıyla gerçekleştirilir. Temel olarak, literatürde 3 farklı hedef programlama metodu mevcuttur.

- **Öncelikli Hedef Programlama:** En yaygın kullanılanıdır. Bu metotta hedeflere farklı öncelikler verilerek (öncelik sırası atanarak) istenmeyen sapmaların minimizasyonu bu sıralamaya göre gerçekleştirilir.
- **Ağırlıklı Hedef Programlama:** İstenmeyen sapmaların ağırlıklı toplamaları kullanılarak birleşik amaç fonksiyonunun minimizasyonu gerçekleştirilir.
- **Chebyshev (minmax) Programlama:** Bu metotta belirlenen hedeflerden maksimum sapmaların minimizasyonu gerçekleştirilir.

1.1.6.1 Öncelikli Hedef Programlama

Hedef Programlamanın bu tipi Sözlüksel (Lexicographic) Hedef Programlama olarak da isimlendirilir. Bu yöntemde hedefler karar vericiden sağlanan bilgiler ışığında hiyerarşik olarak sıralanır. Bu sıralamaya bağlı olarak hedeflerin ilgili sapan değişkenleri minimize yapılır. İjiri 1965 yılında hedeflerin birbirlerine göre göreceli ağırlıklarını ve öncelik sıralamalarını içeren bir model önerisinde bulunmuştur. Öncelikli Hedef Programlamayla ilgili ilk formülasyonu Lee tarafından 1972 yılında gerçekleştirilmiştir (Schniderjans 1995: 6). Charnes ve Cooper (1977) ağırlıkları içermeyen, yalnızca hedefler arasında öncelik sıralamasının yapıldığı aşağıdaki modeli önermiştir;

Amaç Fonksiyonu:

$$\text{Min } Z = \sum_{i=1}^m P_i (d_i^+ + d_i^-)$$

Kısıtlar:

$$\sum_{i=1}^m \sum_{j=1}^n a_{ij}x_j - d_i^+ + d_i^- = b_i$$

$$d_i^+, d_i^-, x_j \geq 0 \quad i = 1, 2, \dots, m \quad j = 1, 2, \dots, n$$

$$P_1 > P_2 > \dots > P_m$$

Birinci Öncelikli hedeften başlayarak daha düşük öncelikli hedeflere doğru çözüm gerçekleştirilir.

1.1.6.2 Ağırlıklı Hedef Programlama

Ağırlıklı Hedef Programlama aynı zamanda Archimedian Hedef Programlama olarak bilinir. Bu yöntem ile sapmaların ağırlıklı toplamı minimize edilir. Charnes ve Cooper (1961) ağırlıklı modellerle ilgili çalışmalar gerçekleştirmiş ve 1977 yılında ağırlıklı hedef programlama modelini önermişlerdir. Matematiksel olarak ağırlıklı hedef programlama modeli aşağıdaki gibidir;

Amaç Fonksiyonu:

$$\text{Min } Z = \sum_{i=1}^m (w_i^+ d_i^+ + w_i^- d_i^-)$$

$$\sum_{i=1}^m \sum_{j=1}^n a_{ij} x_j - d_i^+ + d_i^- = b_i$$

$$d_i^+, d_i^-, x_j \geq 0 \quad i = 1, 2, \dots, m \quad j = 1, 2, \dots, n$$

Hedef Programlama probleminin çözümü için kullanılan algoritmalarından Ağırlıklılandırma Yönteminde tek bir amaç fonksiyonu, problemin hedeflerini temsil eden fonksiyonların ağırlıklandırılmış toplamı haline getirilir. Öncelik koruma yöntemi ise, önem derecelerine göre hedeflerin önceliklendirilmesi ile başlar ve yüksek öncelikli hedefin en uygun değerinin düşük öncelikli hedef tarafından kötüleştirilmesine izin verilmeyecek şekilde her seferinde bir hedefi optimum yapar. Bu yöntemler aynı çözümü vermedikleri için farklıdır ve iki yöntemin birbirlerine karşı herhangi bir üstünlüğü olduğu söylenemez (Taha 2000: 348).

1.1.6.3 Minmaks Hedef Programlama

Modelde hedefler ayrı ayrı gösterilir ve öncelik sıralaması yapılmaksızın geleneksel simpleks algoritması kullanılarak çözüm yapılır. Modelin amaç fonksiyonu sadece maksimum sapmanın minimizasyonunu belirleyen uzaklık parametresinden oluşur (Ignizio ve Cavalier 1994). Bu yöntem ilk kez Flavell (1976) tarafından önerilmiştir. Matematiksel olarak aşağıdaki gibidir:

Amaç Fonksiyonu:

Min D

$$(w_i^+ d_i^+ + w_i^- d_i^-) - D \leq 0$$

$$\sum_{i=1}^m \sum_{j=1}^n a_{ij} x_j - d_i^+ + d_i^- = b_i$$

$$d_i^+, d_i^-, x_j \geq 0 \quad i = 1, 2, \dots, m \quad j = 1, 2, \dots, n$$

1.2 Yapay Sinir Ağları

Bilgisayarlar ile bazı yazılım programlama dillerini kullanarak birbirini takip eden işlemler doğru ve güvenilir şekilde ve insan beynine göre daha yavaş bir biçimde fakat hızlıca yapabilmektedir. Bunun yanında, insanların tersine bilgisayarlar belli durumları anlayamaz ve yeni durumlara uyum sağlayamazlar yani olayları muhakeme edemezler. Yapay zekâ, bilgisayarların bu tür karmaşık sorunların üstesinden gelebilmesi için geliştirilmiştir (Öztemel, 2012).

Yapay zekâ, makine veya yazılımın gösterdiği zekâ olarak bilinmektedir. Bu aynı zamanda bir akademik çalışma alanıdır. Bu akademik çalışma alanı, bilgisayarların ve bilgisayar yazılımlarının insan zekâsına yakın oluşturulması için çalışmalar yapmaktadır. Yapay zekâ, insan zekâsına özgü olan, algılama, öğrenme, çoğul kavramları bağlama, düşünme, fikir yürütme, sorun çözme, iletişim kurma, çıkarım yapma ve karar verme gibi yüksek bilişsel (zekâ) fonksiyonları veya bağımsız davranışları sergilemesi beklenen yapay bir işletim sistemidir. Bu sistem aynı zamanda düşüncelerinden tepkiler üretebilmeli (tepkisel yapay zekâ) ve bu tepkileri fiziksel olarak dışa vurabilmelidir (Öztemel, 2012).

Yapay zekânın geçmişi modern bilgisayar bilimi kadar eskidir. Alan Mathison Turing, "Makinelere düşünmeye yeteneği kazandırılabilir mi?" sorusunu ortaya atarak makine zekâsını tartışmaya açmıştır. 1943 yılında II. Dünya Savaşı sırasında kriptoları analiz edebilecek kabileyette üretilen elektromekanik cihazlar ile birlikte bilgisayar bilimi ve yapay zekâ kavramları doğmuştur. Alan Turing, Nazilerin kriptolu mesaj gönderen Enigma makinesinin şifre algoritması ile uğraşıp çözmeye çalışan matematikçiler grubu içindeydi. İngiltere, Bletchley Park'ta şifre çözme amacı ile başlatılan çalışmalar, Turing'in çalışmaları sonucu oluşan bilgisayar ilk örnekleri Heath Robinson, Bombe Bilgisayarı ve Colossus Bilgisayarlarıdır. Bunlar Boole Cebirine dayanan veri işleme mantığı ile çalışan makinelerdir ve ilk defa "Makine Zekâsı" kavramının ortaya atılmasına sebep olmuştur.

Modern bilgisayarın gelişmesine neden olan bu makineler ise programlama yeteneklerini esasında insan zekâsının işlem yönteminden ilham almışlardı. Ancak sonraları, geliştirilen bu makineler günümüzde sunucu bilgisayarlar adını aldılar ve expert sistemler

olarak adlandıracağımız programlar ile günlük hayatımızın problemlerini çözmeye yönelik olarak yaygınlaştılar. 1970'li yıllarda büyük bilgisayar şirketleri sunucu bilgisayarları ucuzlatarak ve basitleştirerek kişisel bilgisayar modeli haline getirdiler ve evde ve ofiste kişisel olarak kullanılır hale getirdiler. Bilgisayarların kişisel hale gelmesi yapay zekâ çalışmalarını yavaşlatmış ve dar bir araştırma çevresi etrafına hapsedmiştir.

Yapay sinir ağları (YSA) yapay zekâ biliminin altında araştırmacıların çok yoğun ilgi gösterdikleri bir araştırma alanıdır. Bilgisayarların öğrenmesine yönelik çalışmaları kapsamaktadır. Teknolojinin gelişmesi izlendiğinde önceleri sadece elektronik data transferi yapmak ve karmaşık hesaplamaları gerçekleştirmek üzere geliştirilen bilgisayarların zaman içerisinde büyük orandaki dataları analiz eden, özetleyebilen ve mevcut bilgileri kullanarak olaylar hakkında yorumlar yapabilen nitelikler kazandığı görülmektedir.

Yapay zekâ konusundaki ilk çalışmalar McCulloch ve Pitts tarafından yapılmıştır. Bu iki araştırmacının keşfettiği YSA hücrelerini kullanan hesaplama modelinin dayanağı;

- 1- önermeler mantığı
- 2- fizyoloji
- 3- Turing'in hesaplama kuramıdır (Winston, 2003).

Herhangi bir hesaplanabilir fonksiyonun sinir hücrelerinden oluşan ağlarla hesaplanabileceğini ve mantıksal işlemlerin gerçekleştirilebileceğini gösterdiler. Bu ağ yapılarının uygun bir şekilde bir araya getirilirse öğrenme yeteneği kazanabileceğini de ileri sürdüler. Donald Olding Hebb, yapay sinir hücreleri arasındaki bağlantıların şiddetlerini değiştirmek için basit bir kural önerdi (İleri bölümlerde bu kural anlatılacaktır). Böylece kendi kendine öğrenebilen YSA gerçekleştirmek de mümkün hale geldi.

1950 yılına gelindiğinde Alan Turing ve Claude Elwood Shannon satranç programları yazıyorlardı. İlk YSA düşüncelerine dayanan bilgisayar SNARC Massachusetts Üniversitesinde Marvin Minsky ve Dean Edmonds tarafından 1951 yılında geliştirildi. Çalışmalarını Princeton Üniversitesi'nde sürdüren McCarthy, Minsky, Shannon ve Rochester'la birlikte 1956 yılında Dartmouth'ta iki aylık bir açık çalışma düzenledi. Bu toplantıda birçok çalışmanın temelleri atılmakla birlikte, toplantının en önemli özelliği McCarthy tarafından önerilen yapay zekâ adının konmasıdır. İlk kuram ispatlayan prog-

ramlardan Mantık kuramcısı (Logic Theorist) burada Newell ve Simon tarafından tanıtılmıştır (Winston, 2003). Rosenblatt, Hebb'in çalışmalarını takiben Perseptron modelini geliştirmiştir. Arkasından Widrow ve Hoff tarafından ADALINE (ADaptive LInear NEuron) geliştirildi (Winston, 2003).

1960'lı yılların sonunda yapay sinir ağı çalışmaları duraklama devrine girdi. Minsky ve Pappert yazdıkları “Algılayıcılar” kitabı ile yapay sinir ağlarına dayalı tek katmanlı algılayıcıların doğrusal olmayan problemlere karşı doğru çalışmadığını ve bu nedenle de bilimsel bir değerinin olmadığını ispatladılar. Bunun için de XOR problemini kanıt olarak kullandılar. 1969 yılına kadar yapılan önemli çalışmaların kronolojik olarak listesi Tablo 4 'de verilmiştir (Winston, 2003).

Tablo 4
Yapay Zekâ Çalışmaları Gelişim Evreleri - 1

1890	İnsan beyninin yapısı ve fonksiyonları ile ilgili ilk yayının yapılması
	İnsan beyninin bileşenlerinin belirli bir düzenek ile sinir hücrelerinden (nöronlar) oluştuğu fikri benimsenmiştir.
	Yapay sinir hücrelerine dayalı hesaplama teorisinin ortaya atılması ve eşik değerli mantıksal devrelerin geliştirilmesi
	Hebb öğrenme prosedürünü bilgisayarlar tarafından gerçekleştirilecek şekilde geliştirmiştir.
1957	Perseptron geliştirilmiştir.
1959	Widrow ve Hoff ADALINE öğrenme algoritmasını geliştirmiştir.
1965	İlk makine öğrenmesi kitabı yayınlanmıştır.
1967 - 1969	Bazı gelişmiş öğrenme algoritmaları geliştirilmiştir. (Grosberg vs.)
1969	Tek katmanlı algılayıcıların yetersizlikleri ispatlanmıştır.

Kaynak: WINSTON, W. (2003), “Operations Research Applications and Algorithms”

Çalışmaların 1969 yılında kesilmesine ve ödenek bulunamamasına karşın bazı bilim adamları çalışmalarına devam ettiler. 1970'li yıllara gelindiğinde çalışmalar meyvelerini vermeye başladı ve yapılan çalışmalar sonucunda Tablo 5 'deki sonuçlar elde edildi.

Tablo 5
Yapay Zekâ Çalışmaları Gelişim Evreleri - 2

1970	Fukushima tarafından Neocognitron modeli tanıtılmıştır.
1972	Korelasyon matris belleği geliştirilmiştir.
1974	Geriye yayılım modeli geliştirilmiştir.
1978	ART modeli geliştirilmiştir.
1982	Kohonen öğrenmesi ve SOM modelinin geliştirilmiştir.
1982	Hopfield ağları geliştirilmiştir.
1982	Çok katmanlı algılayıcıları geliştirilmiştir.
1984	Boltzmann Makinesi geliştirilmiştir.
1985	Çok katmanlı algılayıcıları geliştirilmiştir. (genelleştirilmiş delta öğrenme kuralı ile)
1988-	RBF - PNN modeli geliştirilmiştir.
1991-	GRNN modeli geliştirilmiştir.

Kaynak: WINSTON, W. (2003), “Operations Research Applications and Algorithms”

1991’den günümüze sayısız çalışma yapılmış ve uygulamalar geliştirilmiştir.

Yapay zekâ bilimindeki gelişmeler bu sistemlerde çeşitlendirmeye neden olmaktadır. Bir taraftan donanım teknolojisi gelişmekte ve daha hızlı çalışabilen, daha çok bilgiyi saklayabilen, daha karmaşık sistemleri ve fonksiyonları yerine getiren bilgisayarlar ve bilgisayar sistemleri oluşturulmakta iken diğer taraftan yazılımı teknolojisi gelişmekte ve işleme yetenekleri, öğrenme, karar verebilme, problem çözme, muhakeme yapabilme yöntemleri ve bu yöntemlere dayalı yazılım sistemleri geliştirilmektedir. Bu gelişmeler ise zeki sistemlerin her geçen gün daha yaygın olarak bilinmesine ve gerçek hayatta insanların kullanımına alınmasına neden olmaktadır (Öztemel (2012)).

Artık bilgisayarlar insan karar verme sürecine oldukça benzer bir karar verme sürecine kavuşmakta ve daha karmaşık fakat kullanışlı sistemler ortaya çıkmaktadır. Çalışmalar laboratuvarlardan çıkmakta ve her geçen gün daha fazla ticari sistemler üretilmektedir.

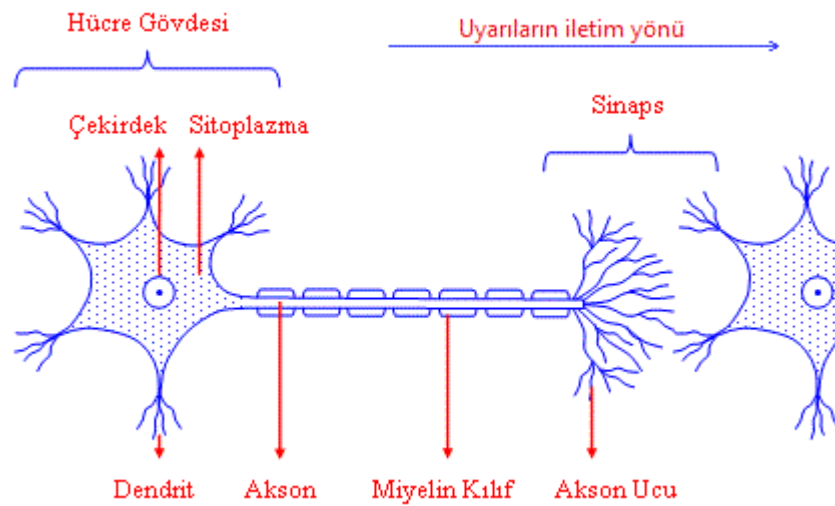
1.2.1 Yapay Sinir Ağlarının Genel Yapısı

Yapay sinir ağları, insan beyninin özelliklerinden olan öğrenme yolu ile yeni bilgiler türetebilme, oluşturabilme ve keşfedebilme gibi yetenekleri bir yardım almadan kendi

kendine gerçekleştirmek amacı ile kurulan bilgisayar sistemleridir. Bu yetenekleri geleneksel programlama yöntemleri ile gerçekleştirmek oldukça zordur veya mümkün değildir. O nedenle, yapay sinir ağları bilim dalının, programlanması çok zor veya mümkün olmayan olaylar için geliştirilmiş kendi kendini geliştirebilen bilgi işleme ile ilgilenen bir bilgisayar bilim dalı olduğu söylenebilir (Öztemel (2012)).

1.2.1.1 Biyolojik Sinir Hücreleri

İnsan merkezi sinir sistemi, evrende insanın bildiği en karmaşık biyolojik yapılandırma. Milyarlarca sinir hücresi, bunların aralarındaki trilyonlarca bağlantı ve “yardımcı hücreler” olarak nitelenen glia hücreleri, sinir sisteminin ana yapısını oluşturur. Bu akıl almaz düzeydeki karmaşık yapı, bu günkü bilgilerimiz ışığında, tüm canlılık olaylarını ve davranışları düzenleyen bir ara-birim olarak görev yapar. Merkezi sinir sistemimiz 100 milyardan fazla nörondan yani sinir hücresinden oluşur. Her bir nöronda on bine yakın sinaps vardır. Bu, bir nöronun aynı anda on bin ayrı sinir hücresi ile bağlantı kurabilme yeteneğini olduğunu göstermektedir(Biyoloji ders notları,2017). Burada kısaca özetlenen beyindeki nöron ağı, tüm dünyayı saran telefon ağından birkaç yüz kat daha karmaşık bir yapıya sahiptir. Şekil-7, tipik bir biyolojik sinir hücresinin yapısını göstermektedir.



Şekil 7. Biyolojik Sinir Hücresi

Kaynak: Biyoloji ders notları, <http://www.biyolojidersnotlari.com/denetleme-ve-duzenleme-sistemi-sinir-sistemi-ve-endokrin-sistem.html>

Sinir sistemini merkezi ve çevresel (periferik) sinir sistemleri oluşturmaktadır. Çevresel sistem, vücudun her yanından alınan duyu (tat, dokunma, görme, işitme, vücudun pozisyonu, ağrı, ısı, titreşim gibi) bilgilerini merkeze taşıyan ve merkezden çıkan emirleri kas veya salgı bezi gibi ilgili yerlere götüren sinir kablolarından oluşur. Çevresel sinir sistemi verileri gerekli yerlere ileten taşıyıcısı olarak düşünebiliriz. Merkezi sinir sistemi ise, çevresel sinir sisteminden gelen verileri değerlendirip, ürettiği emirler aracılığıyla vücudumuzun yeni durumlara uyum sağlamasına yardımcı olur (Sinirsistemi.gen.tr,2016).

Sinir sisteminin ana işini yürüten hücreler, nöron (=sinir hücresi) denen özel hücrelerdir. Bu hücreler, istisnaları olmak üzere, bir gövde, ağaç gibi yan dallar (dendritler) ve bir de, bazen dallanabilen ve hücrenin “kararlarını” diğerlerine ileten, tek bir uzantıdan (akson) oluşurlar. Nöronlar, görevleri ve bulundukları yerlere göre çok değişik şekil ve kimyasal içerik farkları gösterirler. Hücrenin gövde kısmında bulunan çekirdek, hücrenin temel işlevlerini belirleyen ve DNA molekülü üzerinde kodlanmış halde bulunan genetik bilgiyi içerir. DNA üzerindeki bilgi, hücrenin bulunduğu ortama, ortamdaki değişimlere ve hücrenin iç çevresine bağlı olarak deşifre edilerek, hücre içi olayların meydana gelmesini sağlar(Sinirsistemi.gen.tr,2016). .

Sinir hücreleri birbirleri ile sıkı ilişki halindedirler. Bu ilişki, sinirsel hareketin temelini oluşturan bilgi alışverişini sağlar. Hücreler arası bilginin aktarıldığı noktaların adı SİNAPS’dir. Sinapslar, değişik tip ve özelliklerdedir. Sinapsların hepsi bilginin aktarılmasından sorumludur. Nöronlar kendi aralarında bağlantılar kurarak beyin fonksiyonlarını oluşturan ana elemanlardır (Sinirsistemi.gen.tr,2016). Elbette ki, bu elektriksel devre sistemi, herhangi bir insanın hatta bir sinir bilimcinin hayal edebileceği karmaşıklığın çok çok ötesinde bir karmaşıklığa sahiptir. Genel olarak tüm hücrelerde hücrenin içi, dış ortama göre -70 ila -90 mili volt kadarlık bir negatif yüke sahiptir (Koz,2017). Canlı tüm hücrelerin önemli bir özelliği olan bu "zar potansiyeli", sinir ve kas hücreleri gibi "uyarılabilir hücreler" dediğimiz hücrelerde amaca göre değiştirilerek haberleşme ve biyolojik işlevleri icra etme amaçlı olarak kullanılmaktadırlar. Genel olarak bir sinir hücresi, gövde ve dendrit (dendron=ağaç) dediğimiz gövde dalları aracılığıyla diğer hücrelerden gelen verileri “alır” (Sinirsistemi.gen.tr,2016). Bu veriler, hücre içindeki duruma ve tüm verilerin toplam etkisine göre, akson vasıtası ile diğer bir hücreye akta-

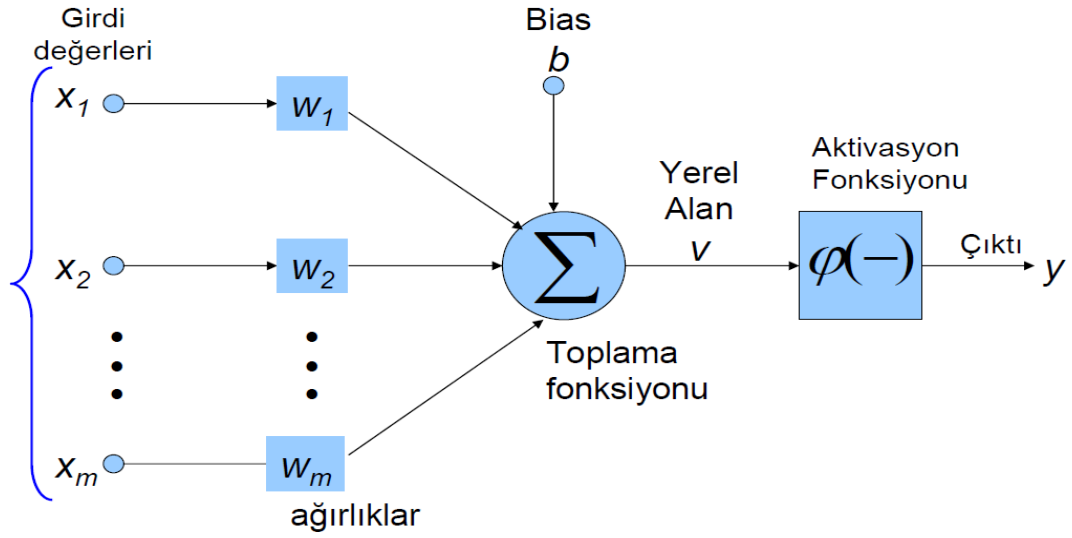
rılır (Sinirsistemi.gen.tr,2016). . Daha sonra, aksonla gönderilen bu bilgi, o aksonun dalları aracılığıyla bir veya (dallanmalar sayesinde) binlerce sinir hücresine (veya kas ve salgı bezi hücreleri gibi diğer hücrelere) ulaştırılır ve bu hücreler, yine aynı mekanizma ile bu uyarının gerektirdiği işi yaparlar.

Biyolojik sistemlerde öğrenme, nöronlar arasındaki sinaptik bağlantıların güncellenmesi ile olur. İnsanlar küçüklükten itibaren yaşayarak ve tecrübe ederek öğrenme sürecini gerçekleştirirler. Bu süreç içinde beyin sürekli bir gelişme göstermektedir. Yaşayıp tecrübe ettikçe bağlantılar güncellenir ve yeni olaylar karşısında yeni bağlantılar oluşur. Bu sayede öğrenme gerçekleşir (Yazgan,2015).

1.2.1.2 Yapay Sinir Ağları Genel Yapısı

YSA'lar insan sinir ağlarındaki öğrenme şeklini kıstas alarak geliştirilmiştir ve insan biyolojik sinir sisteminin çalışma şeklini benzerleyen programlardır. Yapay sinir ağları, olayların oluşunu incelemekte, incelemeler sonucunda olay hakkında tümevarım yapmakta, olayla ilgili bilgiler toplamakta ve sonra hiç görmediği olaylara karşı öğrendiği bilgileri kullanarak olaylar hakkında karar verebilmektedir (Öztemel,2012). YSA teknolojisi insan oğlunun doğayı araştırma ve taklit etme çabalarının sonucunda ortaya çıkan bir teknolojidir.

Simüle edilen sinir hücreleri (işlem elemanları) içerirler ve bu işlem elemanları çeşitli şekillerde birbirlerine bağlanarak ağı oluştururlar. YSA'ların öğrenme, hafızaya alma ve veriler arasındaki bağlantıyı bulma yetenekleri vardır. YSA'lar, normalde bir insanın düşünme ve gözlemlemeye yönelik doğal yeteneklerini gerektiren problemlere çözüm üretmektedir. Bir insanın düşünme ve gözlemleme yetenekleri vardır. Bazı problemlere çözümler bulabilmesinin sebebi ise insan beyninin bu yetenekleri kullanma ve öğrendiklerinden ders çıkarma kabiliyetidir. YSA'da öğrenme, eğitime yoluyla örnekler kullanılarak olur; önceden bilinen girdi/çıktı verilerinin işlenmesiyle yani eğitime algoritmasının bu verileri kullanarak bağlantı ağırlıklarını bir yakınsama sağlanana kadar ayarlamasıyla öğrenme sağlanmış olur (Öztemel,2012). Şekil 8'de genel bir yapay sinir ağı yapısı gösterilmiştir.



Şekil 8. Genel Yapay Sinir Ağı Yapısı

Kaynak: Psikolojik, <http://www.psikolojik.gen.tr/yapay-sinir-aglari.html>, 2017

YSA'lar, ağırlıklandırılmış şekilde birbirlerine bağlanmış birçok işlem elemanlarından oluşan matematiksel sistemlerdir. Bir nöron ağırlıklandırılmış olarak sinyalleri toplar ve aktivasyon fonksiyonuna uygular. Aktivasyon fonksiyonu ileride de anlatılacağı gibi bir denklemdir. Nöronlar, diğer nöronlardan veya girdilerden gelen sinyalleri alır; bunları ağırlıklandırarak toplama fonksiyonu ile biraraya getirir, eşik değerini geçen sinyalleri aktivasyon fonksiyonuna uygular ve sonunda sayısal bir sonuç verir (Öztemel, 2012). Genelde, işlem elemanları kabaca gerçek nöronlara karşılık gelirler ve bir ağ içinde birbirlerine bağlanırlar; bu yapı da sinir ağlarını oluşturmaktadır. Tablo 6'da biyolojik sinir hücresi yapısının yapay sinir ağları yapısı ile karşılaştırılması gösterilmiştir.

Tablo 6

Biyolojik Sinir Hücresi Yapısının Yapay Sinir Ağları Yapısı ile Karşılaştırılması

Biyolojik Sinir Ağı	Yapay Sinir Ağı	Görevi
Dendrit	Alıcı ve Toplayıcı	Çevreden gelen veriyi alır.
Akson	Verici	Gövdenin oluşturduğu veriyi iletir.
Hücre Gövdesi	Aktivasyon Fonksiyonu	Gelen veriyi toplayarak yorumlar ve çıktıyı oluşturur.
Sinaptik Ağırlıklar	Bağlantı Ağırlıkları	Öğrenilenleri hafızada saklar

Kaynak: Çelik, 2008

Sinirsel hesaplamanın merkezinde dağıtılmış, kendini ayarlayabilen (adaptif) ve doğrusal olmayan işlem kavramları vardır (Öztemel,2012). YSA'lar, bildiğimiz programlama türlerinden farklı şekilde çalışmaktadırlar. Geleneksel işlemcilerde, tek bir merkezi işlem elemanı her hareketi sırasıyla gerçekleştirir. YSA işlem elemanlarının her biri problemin farklı parçası ile ilgilenir ve problemin hangi parçası hakkında bilgisi olduğu bilinemez. Çok sayıda basit işlem elemanlarından oluşan YSA işlem elemanları problemin farklı bölümleri hakkında bilgiye sahiptir. İşlem elemanlarının çalışma şekli basit gibi görünür fakat YSA'nın gücü, işlem yükünü paylaşan işlem elemanlarının birbirleri arasındaki bağlantı yapısının ağırlıklarının ayarlanmasından gelmektedir. YSA'larda benzer özelliklere sahip işlem elemanları katmanlar halinde oluşturulurlar ve transfer fonksiyonları eş zamanlı olarak çalıştırılırlar. Tüm ağların işlem elemanları birden fazla veriyi toplayan ve aktivasyon fonksiyonuna uygulayarak çıktı üreten özelliklere sahiptirler. YSA'nın temeli olan aktivasyon fonksiyonu ağın mimarisini kuranlar tarafından şekillendirilir. Fonksiyonun temel yapısını ağırlıkların büyüklüğü ve işlem elemanlarının işlem şekli belirler. YSA'ların davranışları, yani girdi veriyi çıktı veriye nasıl ilişkilendirdikleri, ilk olarak nöronların transfer fonksiyonlarından, nasıl birbirlerine bağlandıklarından ve bu bağlantıların ağırlıklarından etkilenir.

1.2.1.3 Yapay Sinir Ağlarının Elemanları

YSA'nın temeli olan yapay sinir hücrelerinin beş temel bölümü vardır (Yüksek,2007):

- Dış dünyadan gelen girdiler
- İşlem elemanı ağırlıkları
- Toplama Fonksiyonu
- Aktivasyon Fonksiyonu
- Dış dünyaya gönderilen çıktılar

• Girdiler

Girdiler nöronlara gelen verilerdir. Yapay sinir ağının yapısına bağlı olarak girdiler yapay sinir hücresine bir diğer hücreden gelebileceği gibi doğrudan dış dünyadan da gelebilir. Bu girdilerden gelen veriler biyolojik sinir hücrelerinde olduğu gibi toplanmak üzere nöronlara gönderilir (Kakıcı, 2009).

- **Ağırlıklar**

Yapay sinir hücresine gelen bilgiler girdiler üzerinden aktivasyon fonksiyonuna ulaşmadan önce geldikleri bağlantıların ağırlığıyla çarpılır. Bu işlem ile girdilerin çıktı üzerindeki etkisi ayarlanabilmektedir. Bu ağırlıklar herhangi bir reel değer olabilir. Ağırlık katsayısı sıfır olan girdilerin çıktı üzerinde normal olarak bir etkisi olamaz.

- **Toplama Fonksiyonu**

Birleştirme fonksiyonu bir yapay sinir hücresine ağırlıklarla çarpılarak gelen girdileri toplayarak o hücrenin net girdisini hesaplayan bir fonksiyondur.

$$NET = \sum_{i=1}^N X_i * W_i$$

Bir problem için en uygun toplama fonksiyonu belirlenirken geliştirilmiş bir yöntem yoktur. Bu nedenle en iyi toplama fonksiyonunu bulmak için denemeler yapılır. Bazen bazı işlem hücrelerinin toplama fonksiyonları farklı seçilir. Bu konulara karar vermek tasarımcıya aittir (Çayıroğlu, 2015). Toplama fonksiyonları örnekleri Tablo 7’de verilmiştir.

Tablo 7
Toplama Fonksiyon Örnekleri

$Net = \sum_{i=1}^N X_i * W_i$	Ağırlık değerleri ile girdi değerleri çarpılır ve bulunan değerler toplanarak toplam değer bulunur.
$Net = \prod_{i=1}^N X_i * W_i$	Ağırlık değerleri ile girdi değerleri çarpılır ve bulunan değerler çarpılarak değer bulunur.
$Net = Max (X_i * W_i)$	Ağırlık değerleri ile girdi değerlerinin çarpımının en büyüğü alınır.
$Net = Min (X_i * W_i)$	Ağırlık değerleri ile girdi değerlerinin çarpımının en küçüğü alınır.
$Net = \sum_{i=1}^N Sgn(x_i * W_i)$	Girdilerle ağırlıklar çarpıldıktan sonra pozitif ile negatif olanların sayısı bulunur. Büyük olan sayı hücrenin net girdisi olarak kabul edilir.
$Net = Net(eski) + \sum_{i=1}^N X_i * W_i$	Hücreye gelen bilgiler ağırlıklı olarak toplanır. Daha önce hücreye gelen bilgilere yeni hesaplanan girdi değerleri eklenerek hücrenin net girdisi hesaplanır.

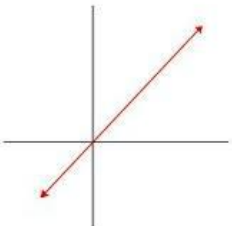
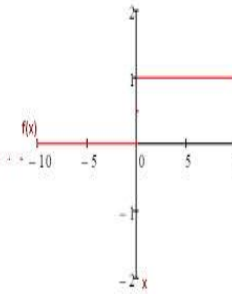
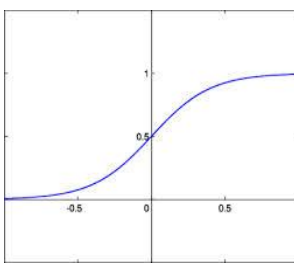
Kaynak: Şengöz N., Yapay Sinir Ağları, Süleyman Demirel Üniversitesi Bilgisayar Mühendisliği, <http://www.derinogrenme.com/2017/03/04/yapay-sinir-aglari/>, Erişim Tarihi: 10.1.2017

- **Aktivasyon Fonksiyonu**

Toplama fonksiyonundan çıkan NET toplam hücrenin çıktısını oluşturmak üzere aktivasyon fonksiyonuna iletilir. Aktivasyon fonksiyonu genellikle doğrusal olmayan bir fonksiyon seçilir. Yapay sinir ağlarının bir özelliği olan “doğrusal olmama” aktivasyon fonksiyonlarının doğrusal olmama özelliğinden gelmektedir. Aktivasyon fonksiyonu se-

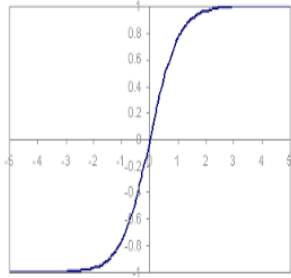
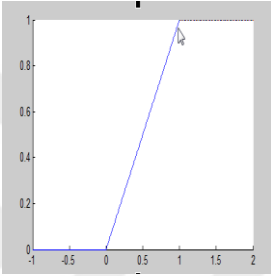
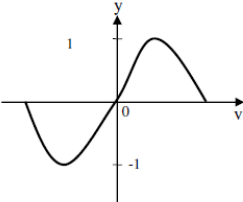
çilirken dikkat edilmesi gereken bir diğer nokta ise fonksiyonun türevinin kolay hesaplanabilir olmasıdır. Geri beslemeli ağlarda aktivasyon fonksiyonunun türevi de kullanıldığı için hesaplamanın yavaşlamaması için türevi kolay hesaplanır bir fonksiyon seçilir. Günümüzde en yaygın olarak kullanılan “Çok katmanlı algılayıcı” modelinde genel olarak aktivasyon fonksiyonu olarak “Sigmoid fonksiyonu” kullanılır(Çayıroğlu,2015). Aktivasyon fonksiyon örnekleri Tablo 8 ve Tablo 9’da verilmiştir.

Tablo 8
Aktivasyon Fonksiyon Örnekleri - 1

Doğrusal Aktivasyon Fonksiyonu		$F(\text{Net}) = A * \text{NET}$ (A = sabit sayı)	Toplama fonksiyonundan çıkan sonuç, belli bir katsayı ile çarpılarak hücrenin çıktısı olarak hesaplanır.
Adım Aktivasyon Fonksiyonu		$F(\text{Net}) = \begin{cases} 1 & ; \text{Net} > \text{Eşik Değer} \\ 0 & ; \text{Net} \leq \text{Eşik Değer} \end{cases}$	Gelen Net girdinin belirlenen bir eşik değerinin altında veya üstünde olmasına göre hücrenin çıktısı 1 veya 0 değerini alır.
Sigmoid Aktivasyon Fonksiyonu		$F(\text{Net}) = \frac{1}{1 + e^{-\text{Net}}}$	Bu fonksiyon girdi değerlerinin her biri için 0 ile 1 arasında bir değer üretir.

Kaynak: Çayıroğlu, İ. (2015), İleri Algoritma Analizi – 5 Yapay Sinir Ağları, <http://www.ibrahimcayiroglu.com/Dokumanlar/IleriAlgoritmaAnalizi/IleriAlgoritmaAnalizi-5.Hafta-YapaySinirAglari.pdf>, Karabük Üniversitesi

Tablo 9
Aktivasyon Fonksiyon Örnekleri - 2

Tanjant Hiperbolik Aktivasyon Fonksiyonu		$F(Net) = \frac{e^{Net} + e^{-Net}}{e^{Net} - e^{-Net}}$	Sigmoid fonksiyonunda çıkış değerleri 0 ile 1 arasında değişirken hiperbolik tanjant fonksiyonunun çıkış değerleri -1 ile 1 arasında değişmektedir.
Eşik Değer Aktivasyon Fonksiyonu		$F(Net) = \begin{cases} 0 & ; \quad Net \leq 0 \\ Net & ; \quad 0 < Net < 1 \\ 1 & ; \quad Net \geq 1 \end{cases}$	Gelen bilgilerin 0 'dan küçük-eşit olduğunda 0 çıktısı, 1 den büyük-eşit olduğunda 1 çıktısı, 0 ile 1 arasında olduğunda ise yine kendisini veren çıktılar üretilebilir.
Sinüs Aktivasyon Fonksiyonu		$F(Net) = \sin(Net)$	Olayların sinüs fonksiyonuna uygun dağılım gösterdiği durumlarda kullanılır.

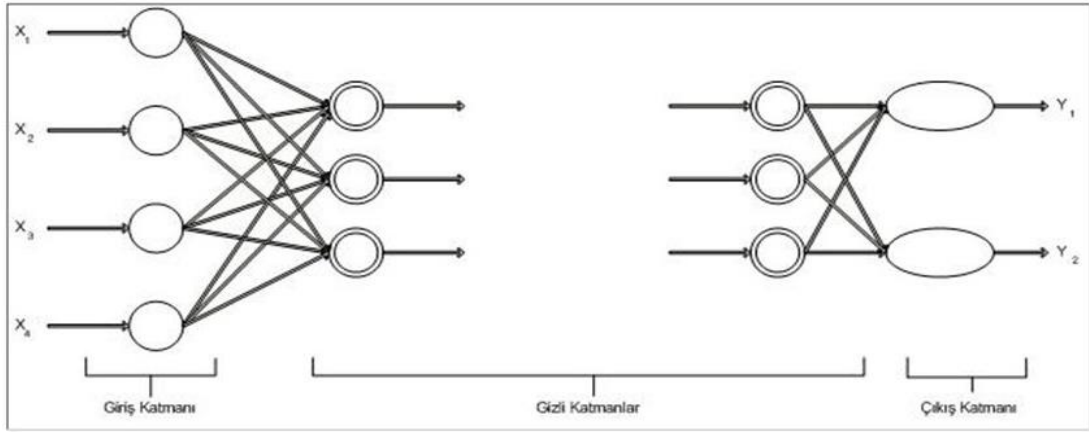
Kaynak: Çayiroğlu, İ. (2015), İleri Algoritma Analizi – 5 Yapay Sinir Ağları, <http://www.ibrahimcayiroglu.com/Dokumanlar/IleriAlgoritmaAnalizi/IleriAlgoritmaAnalizi-5.Hafta-YapaySinirAglari.pdf>, Karabük Üniversitesi

• Çıktılar

Aktivasyon fonksiyonundan çıkan değer nöronların çıktı değeri olmaktadır. Bu değer ister yapay sinir ağının çıktısı olarak dış dünyaya verilir veya işlem hatasına göre tekrar girdi olarak ağına içinde kullanılabilir.

1.2.1.4 Yapay Sinir Ağlarının Yapısı

Yapay sinir ağları 3 ana katmandan oluşur. Her katman içinde nöronlar paralel olarak bulunmaktadır. Şekil 9'da genel yapı verilmiştir.



Şekil 9. YSA Yapısı

Kaynak: Side Player, <http://slideplayer.biz.tr/slide/3395257/>, 2004

Giriş Katmanı: Dış dünyadan uygulanan girdi sayısı kadar nöron olur. Bu katmanda herhangi bir işlem yapılmaz. Gelen girdi değerleri hiçbir değişikliğe uğramadan ara katmanlara gönderilir.

Gizli Katmanlar (Ara katmanlar): Kurduğunuz YSA modeline göre birden fazla ara katman olabilir. Giriş katmanının ağırlıklandırılmış sinyalleri bu katmana gelir. Ara katman sayısı ağdan ağa değişebilir. Bazı yapay sinir ağlarında ara katman bulunmayabilir. Bazı yapay sinir ağlarında ise birden fazla ara katman bulunabilir. Ara katmanlardaki nöronların sayıları giriş sinyali ve çıkış sinyali sayılarından bağımsızdır. Birden fazla ara katman olan ağlarda ara katmanların kendi aralarındaki hücre sayıları da farklı olabilir. Ara katmanların ve bu katmanlardaki nöronların sayısının artması hesaplama karmaşıklığını ve süresini arttırmasına rağmen YSA'nın daha karmaşık problemlerin çözümünde de kullanılabilmesini sağlar.

Çıktı Katmanı: Gizli katmanlardan gelen bilgileri işleyerek çıktıları üreten katmandır. Çıktı katmanında üretilen çıktılar dış dünyaya gönderilir. Geri beslemeli ağlarda çıktı katmanında elde edilen çıktıların hata değerlerine göre giriş katmanındaki nöronların ağırlık değerleri tekrar hesaplanır.

1.2.1.5 Yapay Sinir Ağlarının Türleri

Yapay sinir ağları nöronların yapılarına ve bağlantı şekillerine göre çeşitlere ayrılmaktadır (Öztemel, 2012).

❖ Yapılarına Göre

- **İleri beslemeli:** İleri beslemeli yapay sinir ağlarında nöronlar arasında hiyerarşik bir yapı vardır ve bir katmandaki nöronlar sadece kendinden sonraki katmana veri iletir.
- **Geri beslemeli:** Bu YSA modelinde ise bir nöron kendinden sonraki katmana veri ilettiği gibi kendinden önceki katmana veya kendi katmanına da veri iletebilir.

❖ Öğrenme Algoritmalarına Göre

- **Danışmanlı:** YSA'ya verilen her giriş verisi için çıkış verisi de verilir. YSA hedeflenen çıktıyı elde etmek için katsayılarını günceller.
- **Danışmansız:** YSA'ya giriş verisiyle beraber çıkış verisi verilmez. YSA birbirine benzeyen çıktıları gruplamaya çalışır.
- **Destekleyici:** YSA'ya giriş verilerinin yanı sıra her veri setine ait çıkışa bir puan verilir. YSA katsayılarını en yüksek puanı alacak şekilde düzenler.

❖ Öğrenme Zamanına Göre

- **Statik:** YSA en baştan eğitim verileriyle eğitilir ve ağın yapısı kaydedilir. Ağ bundan sonra hep aynı yapıyla çalışır. Kullanımı sırasında herhangi bir değişikliğe uğramaz.
- **Dinamik:** YSA eğitim verileriyle eğitildikten sonra kullanımı sırasında da kendini eğitmeyi sürdürür. Böylece sürekli öğrenen bir YSA elde edilebilir. Bir yapay sinir ağında nöronların bağlanması sonucu oluşan topoloji, nöronların sahip oldukları toplama ve aktivasyon fonksiyonları öğrenme stratejisi ve kullanılan öğrenme kuralı ağın modelini belirlemektedir.

1.2.2 Yapay Sinir Ağlarının Avantajları ve dezavantajları

YSA'ların avantajları aşağıdaki gibi sıralanmıştır (Çayiroğlu, 2015).

1. **Paralellik:** YSA'lar birçok hücreden meydana gelir ve bu hücreler eş zamanlı olarak çalışarak karmaşık işlemleri yerine getirir. Karmaşık işlemler birçok hücrenin eş zamanlı çalışması sonucunda oluşur. Süreç içerisinde bu hücrelerden her hangi biri işlevini yitirse dahi sistem güvenli bir şekilde çalışmasına devam edebilir.

2. **Dağıtık Bellek:** YSA'larda bilgi ağ içerisinde dağıtılmış olarak tutulur. İşlem hücrelerinin bağlantı sayısı ve ağırlık katsayı değerleri ağın bilgisini gösterir. Tek bir işlem hücresinin tek başına görev yapacak yeteneği yoktur.
3. **Eğitilmemiş Örnekler Karşısında Bilgi Üretebilme:** Eğitim esnasında kullanılan sayısal değerlerden problemin genel özelliklerini elde etmesi ve bunun sonucunda eğitim sırasında kullanılmayan girdiler için de yanıtlar üretebilme yeteneği vardır.
4. **Doğrusal Olmama:** Yapı üzerinde dağıtılmış sayısal olmayan hücre elemanları sayısal olmayan problemlerin çözümünde kullanılmaktadır.
5. **Genelleme Yapabilme:** YSA'lar makina öğrenmesi gerçekleştirebilirler. Yapay sinir ağlarının temel görevi örnekler verilerek eğitimin sağlanmasıdır (Çayıroğlu,2015). Eğitildikten sonra benzer değerler verildiğinde uygun kararlar verebilirler. Daha önce görülmemiş örnekler hakkında bilgi üretebilirler. YSA'lar eğitimleri sırasında kendilerine verilen örneklerden genellemeler çıkarırlar ve bu genellemeler ile yeni örnekler hakkında bilgi üretebilirler.
6. **Bilginin saklanması:** Bilgiler ağın tamamında saklanır. Bilgiler ağın tamamına dağıtılarak ağ bağlantılarında saklanmaktadır. Hücrelerden bazılarının işlevini kaybetmesi eğitimde verilen bilginin kaybolmasına neden olmaz.
7. **Öğrenme:** Örnekleri kullanarak öğrenirler. YSA'nın eğitilebilmesi için örneklerin saptanması ve bu örneklerin uygulanarak istenen çıktılar sağlanana kadar ağın ağırlıklarının ayarlanması gerekmektedir. Ağın başarısı, seçilen örneklerin uygunluğu ve bütünü kapsaması ile doğru orantılıdır. YSA'ya uygun ve bütünlüğü sağlanmış veriler gösterilmez ise yanlış çıktılar üretebilir. Algılama ile ilgili olaylarda kullanılabilirler. YSA'ların en başarılı oldukları uygulamalar algılamaya yönelik uygulamalardır. Bu uygulamalarda başarılı olarak kullanılırlar. YSA'lar çevrimiçi (online) olarak öğrenebilirler ve kendi kendilerini eğitebilirler.

8. **Örüntü ilişkilendirme ve sınıflama yapabilme:** YSA'lar kendilerine verilen örüntüleri ilişkilendirebilirler. Ayrıca verilen örneklerin kümelenmesini sağlarlar. Bir sonraki örneğin hangi kümeye ait olacağına karar verirler.
9. **Eksik bilgi ile çalışabilme:** Eksik bilgi ile çalışabilmektedirler. Geleneksel sistemlerin aksine YSA'lar eğitildikten sonra veriler eksik bilgi içerse dahi çıktı üretebilirler. Eksik bilgi durumunda performans kaybı yaşanmaz. Performans kaybı eksik bilginin önemine bağlıdır.
10. **Hata toleransına sahip olma:** Hata toleransına sahiptirler. YSA'ların eksik verilerle veya bazı hücrelerinin bozulması halinde çalışabilmeleri hatalara karşı toleranslı olduklarını gösterir.
11. **Dereceli bozulma:** Bir YSA hatalı veriler karşısında zaman içerisinde yavaş olarak bozulma gösterir. Ağlar problemin ortaya çıktığı anda hemen bozulmazlar.
12. **Analiz ve Tasarım Kolaylığı:** Kurulum kolaylığı ve parametrelerin deneme yapılma yolu ile bulunmasından dolayı kullanımı oldukça kolaydır. Fakat yalnızca sayısal veriler ile çalışabilmektedir.

YSA'ların dezavantajları ise şunlardır (Çayıroğlu, 2015);

1. **Ağ yapısında kurlsızlık:** YSA'larda probleme uygun ağ yapısının belirlenmesi için geliştirilmiş bir kural yoktur. Uygun ağ yapısı deneyim ve deneme yolu ile belirlenmektedir.
2. **Parametrelerde kurlsızlık:** YSA parametre değerlerinin (öğrenme katsayısı, hücre sayısı, katman sayısı gibi) belirlenmesi oldukça zordur. Belirlenmesi ile ilgili bir yöntem bulunamamıştır.
3. **Sayısal veriler ile çalışabilme:** YSA'lar sayısal bilgiler ile çalışabilmektedirler. Problemler YSA'lara tanıtılmadan önce sayısal değerlere çevrilmek zorundadırlar.
4. **Öğrenmenin sonlanma zamanı:** Ağın eğitiminin ne zaman bitirilmesi gerektiğine ilişkin belli bir yöntem yoktur. Ağın örnekler üzerindeki hatasının belirli

bir değerin altına indirilmesi eğitimin tamamlandığı anlamına gelmektedir. YSA ile ilgili araştırmaların önemli bir kolunu oluşturmaktadır.

5. **Ağın kapalı kutu olması:** YSA davranışları açıklanamamaktadır. Bu sorun YSA'ların en önemli sorunudur. YSA bir probleme çözüm ürettiği zaman, bunun neden ve nasıl olduğuna ilişkin bir ipucu vermez.

1.2.3 Yapay Sinir Ağlarının Uygulama Alanları

YSA'lar başlıca şu amaçlar için kullanılmaktadır;

- Sınıflandırma,
- Modelleme,
- Tahmin uygulamaları

Başarılı uygulamalar incelendiğinde, YSA'ların çok boyutlu, gürültülü, karmaşık, kesin olmayan, eksik, kusurlu, hata olasılığı yüksek makine verilerinin olması ve problemi çözmek için matematiksel modelin ve algoritmaların bulunmadığı, sadece örneklerin var olduğu durumlarda yaygın olarak kullanıldıkları görülmektedir.

Bu amaçla geliştirilmiş ağlar genellikle şu fonksiyonları gerçekleştirmektedirler (Öztemel, 2012);

- Muhtemel fonksiyon kestirimleri
- Sınıflandırma
- İlişkilendirme veya örüntü eşleştirme
- Zaman serileri analizleri
- Sinyal filtreleme
- Veri sıkıştırma
- Örüntü tanıma
- Doğrusal olmayan sinyal işleme
- Doğrusal olmayan sistem modelleme
- Optimizasyon

YSA'lar pek çok sektörde değişik uygulama alanları bulmuştur. Bunlardan bazıları (Çayiroğlu, 2015) ;

Uzay: Uçuş simülasyonu, otomatik pilot uygulamaları, ekipmanların hata kontrolü vs.

Otomotiv: Otomatik yol takip, rehberlik hizmeti, yol şartlarına göre sürüş analizi vs.

Bankacılık: Kredi uygulamaları geliştirilmesi, müşteri analizi, kredi müracaat değerlendirilmesi, bütçe yatırım tahminleri vs.

Savunma: Radar, sonar sistemleri, sinyal işleme, görüntü işleme vs.

Elektronik: Mikroçip bozulma analizi, doğrusal olmayan modelleme vs.

Eğlence: Animasyonlar, özel gösteriler, pazarlama beklentisi vs.

Finans: Kıymetlendirme, pazar araştırma, bütçe öngörüsü, hedeflendirme vs.

Sigortacılık: Ürün iyileştirmesi ve ürün satış politikası geliştirme vs.

Üretim: Üretim işlem kontrolü, yeni ürün oluşturulması, makina yıpranmalarının tespiti, dayanıklılık analizi, kalite kontrolü, iş çizelgeleri hazırlanması vs.

Sağlık: Göğüs kanseri erken teşhis ve tedavisi, EEG, ECG, MR, kalite artırımı, ilaç etkileri analizi, kan analizi sınıflandırma, kalp krizi erken teşhis ve tedavisi gibi.

Petro kimya: Kaynak arama ve kaynak verim analizi

Robotik: Yörünge kontrol, forklift robotları, görsel sistemler, uzaktan kumandalı sistemler, en iyi rotayı belirleme vs.

Dil: Kelime tanıma, yazının çevrilmesi, tercümanlık vs.

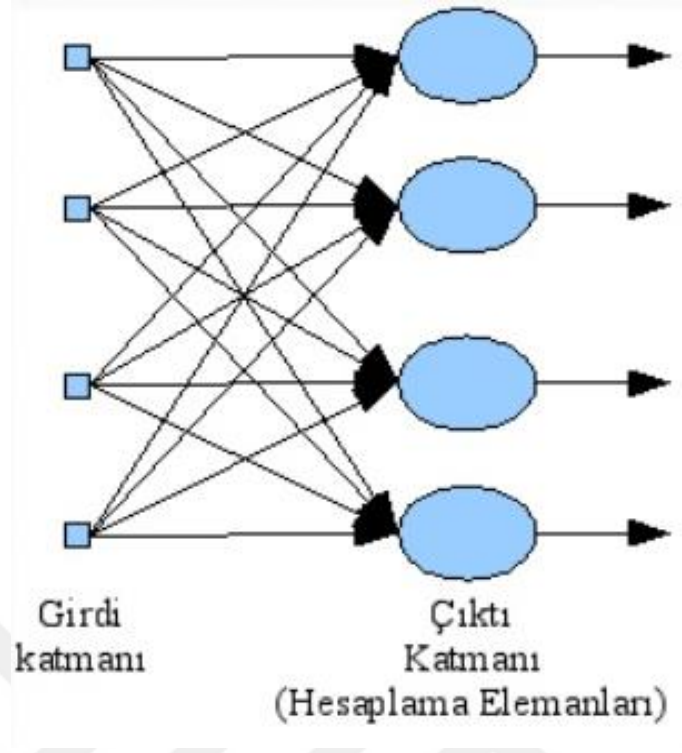
Haberleşme: Görüntü ve veri kıyaslama, filtreleme, eko ve gürültü önümlendirilmesi, ses ve görüntü işleme, trafik yoğunluğunun kontrolü ve anahtarlama vs.

Güvenlik: Parmak izini tespit, kart hilelerini bulma, retina araştırması, yüz tanıma vs.

1.2.4 İleri Beslemeli Yapay Sinir Ağları

Tek Katmanlı Algılayıcılar

Giriş ve çıkış katmanlarından oluşur. Giriş katmanındaki her girdi değerini çıkış katmanına bağlayan bağlantılar vardır. Her bağlantı bir ağırlık değerine sahiptir. Ağırlık çıktısı değerinin sıfır olmamasını sağlayan eşik değerleri vardır. Bu eşik değeri her zaman 1'dir. Bağlantılar tek yönlüdür ve hep ilerdir. Şekil 10'da tek katmanlı yapay sinir ağı modeli gösterilmiştir.



Şekil 10. Tek Katmanlı Yapay Sinir Ağı Modeli

Kaynak: ÖZTEMEL, E. , “Yapay Sinir Ağları”, Papatya Yayıncılık Eğitim, İstanbul, 2012

Ağın çıktısı ağırlık katsayıları ile çarpılmış girdi değerlerinin eşik değeri ile toplanması neticesinde elde edilir. Aşağıdaki şekilde formüle edilir (Öztemel, 2012).

$$\zeta = f \left[\sum_{i=1}^N w_i * x_i + \varphi \right]$$

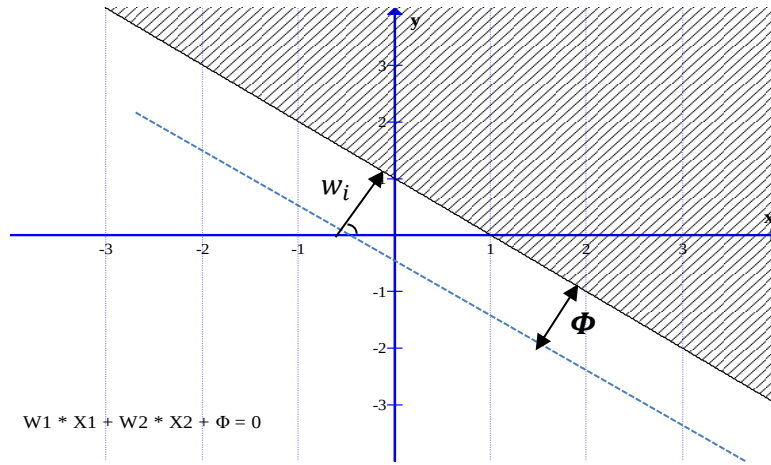
Tek katmanlı algılayıcılarda çıktı fonksiyonu doğrusal fonksiyondur. Bu model ile örnekler iki sınıfa ayrılmaya çalışılır. Ağın çıktısı ya 1 olmakta ya da -1 olmaktadır. Formülasyonu aşağıdaki gibidir (Öztemel, 2012).

$$f(g) = \begin{cases} 1 & ; \text{Çıktı} > 0 \text{ ise} \\ -1 & ; \text{aksi takdirde} \end{cases}$$

İki sınıfa ayıran bu doğrunun denklemi şu şekilde verilebilir.

$$w_1 * x_1 + w_2 * x_2 + \Phi = 0$$

Geometrik gösterimi ise aşağıdaki Şekil 11’deki gibidir.



Şekil 11. İki Sınıfa Ayıran Doğrunun Geometrik Gösterimi

Kaynak: Kaba S., “Yapay Sinir Ağları ile Değerli Kağıt Tanıma Sistemi”, Haliç Üni. Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı, <https://www.slideshare.net/Firari/yapay-sinir-alar-ile-deerli-kat-tanma-sistemi>

Bu ağlarda öğrenmeden kasıt ağın sınıf ayıracı doğrusunun pozisyonunu her iki grubu en iyi şekilde ayıracak şekilde belirlemektir. Bunu sağlamak için ağırlık değerlerini ve eşik değerlerini ayarlamaktan geçmektedir. Öğrenmeden kasıt verilen giriş çıkış değerleri ile bu iki değişken için en iyi değeri bulmaktır. Ağırlık değerleri doğrunun açısını belirlerken eşik değerleri de eksen kaymalarını belirlemektedir. Her t anında en doğru bulmak için ağırlık değerleri ve eşik değerleri aşağıdaki formüle göre değiştirilir (Öztemel,2012) .

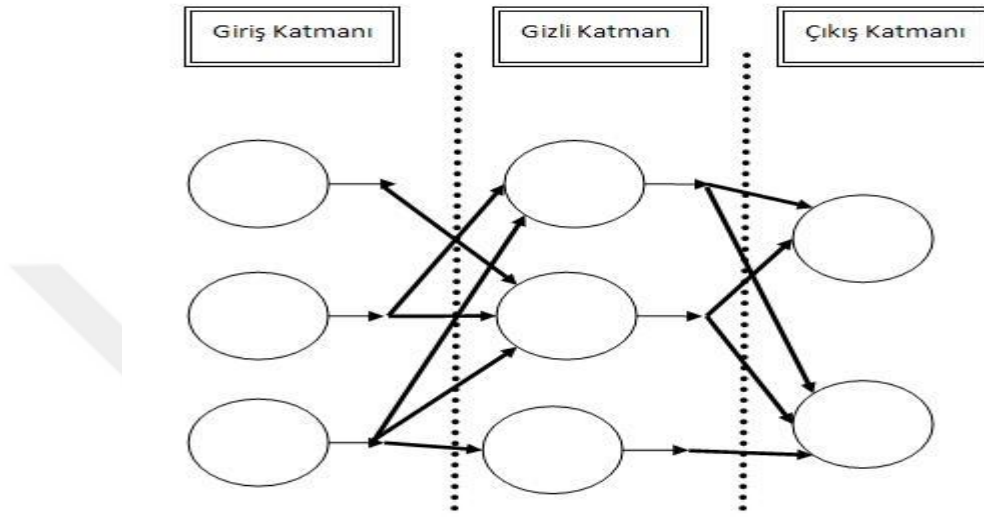
$$W_i(t + 1) = W_i(t) + \Delta W_i(t)$$

$$\Phi_i(t + 1) = \Phi_i(t) + \Delta \Phi_i(t)$$

Cok Katmanlı Algılayıcılar

Marvin Minsky ve Seymour Papert 1969 yılında yazdıkları “Perceptrons: an introduction to computational Geometry” kitapta doğrusal olmayan problemler için tek katlı katmanların uygun olmadığını XOR problemini kullanarak ispatlamışlardır. Bunun gibi doğrusal olmayan problemlerin çözümü için daha gelişmiş yapılara ihtiyaç olduğu ortaya konmuştur. 1969 yılındaki bu ispattan sonra çalışmalar yapay sinir ağları için durma noktasına gelmiştir. XOR problemini çözmek için çok katmanlı algılayıcı model Rumelhart ve arkadaşları[1986] tarafından geliştirilmiştir. Şekil 12’de çok katmanlı algılayıcı model örneği gösterilmiştir. Bu modele hata yayma modeli veya geriye yayım

modeli (backpropagation network veya backward propagation of errors) de denmektedir. Bu modelin geliştirilmesi ile bu modele ilgi hızla artmış; birçok araştırmacı bu model üzerinde çalışmaya başlamıştır. Doğrusal olmayan problemlerin çözümünde kullanılmaya başlanmıştır. Bu model günümüz mühendislik problemlerinin hemen hemen hepsine çözümler üretebilecek bir güce sahiptir (Öztemel,2012).



Şekil 12. Çok Katmanlı Yapay Sinir Ağı

Kaynak: Atmaca K., “Yapay Sinir Ağları Nedir?”, <http://kenanatmaca.com/yapay-sinir-aglari-nedir/>, Erişim Tarihi: 12.06.2017

Delta öğrenme kuralı öğrenme yöntemi olarak kullanılmaktadır. Basit algılayıcı modellerin öğrenme kuralının daha gelişmiş şeklidir. Bu öğrenme kuralının temel amacı giriş – çıkış arasındaki farkları en aza indirmektir. Bu yöntem bulunan hatayı bütün ağı yayarak nöronlardaki hata oranını minimize etmeyi sağlar. Bu nedenle de hata yayma ağı da denmektedir.

Çok Katmanlı Ağ modelinde bilgiler girdi katmanından sunulur ve ara katmanlardan geçerek çıktı katmanına gider ve ağı sunulan girdilere karşılık ağı cevabı dışarıya verilir. Ara katmanda birden fazla ara katman olabilir. Bilinen örneklerden yola çıkarak girdiler ağı verilir ve çıktılara göre ağırlıkların ayarlanması sağlanır. Bu işlem iki kısımdan oluşur. Önce belirlenen katsayıya göre çıktı hesaplanır. Eğer değer doğru değil ise geriye doğru ağırlıklar değiştirilerek tekrar hesaplanır.

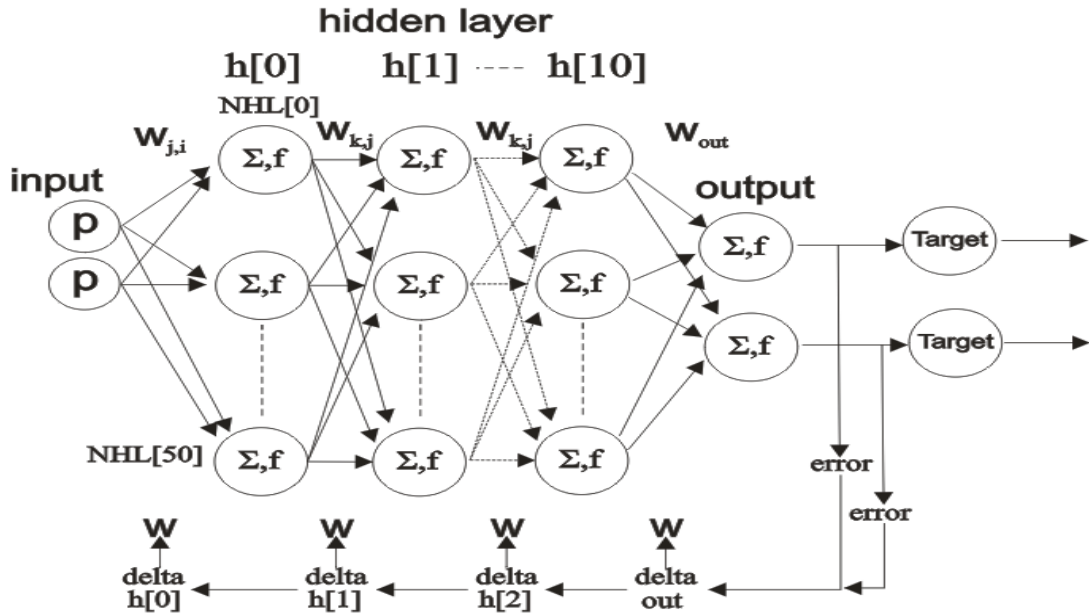
İleri doğru hesaplama şöyle yapılıyor:

Ara katmandaki k. elemanın çıktısı ζ_k^i

$$Toplam\ Girdi_{ara\ katman}^{girdi\ katmanı\ elemanı} = \sum_{k=1}^n w_{kj} * \zeta_k^i$$

Geri Yayılım Algoritması

Geri Beslemeli Yapay Sinir Ağları (BPN) çok sık kullanılan bir yapay sinir ağı yapısıdır. Standart geri yayılım algoritması kullanır. Ağ ağırlıklarının, performans fonksiyonunun negatif gradyanı yönünde ilerlediği gradyan iniş algoritmasıdır. Bir çok çeşidi olan geri yayılım algoritması, gradyan iniş ve newton metodu gibi standart optimizasyon tekniklerine dayanmaktadır (Hamzaçebi ve Kutay(2004)). Geri yayılım algoritması, en çok kullanılan öğreticili öğrenme algoritmasıdır. GBYSA doğrusal olmayan bir özellik gösterir. Şekil - 13'de geri beslemeli yapay sinir ağının çalışma mantığı görülmektedir.



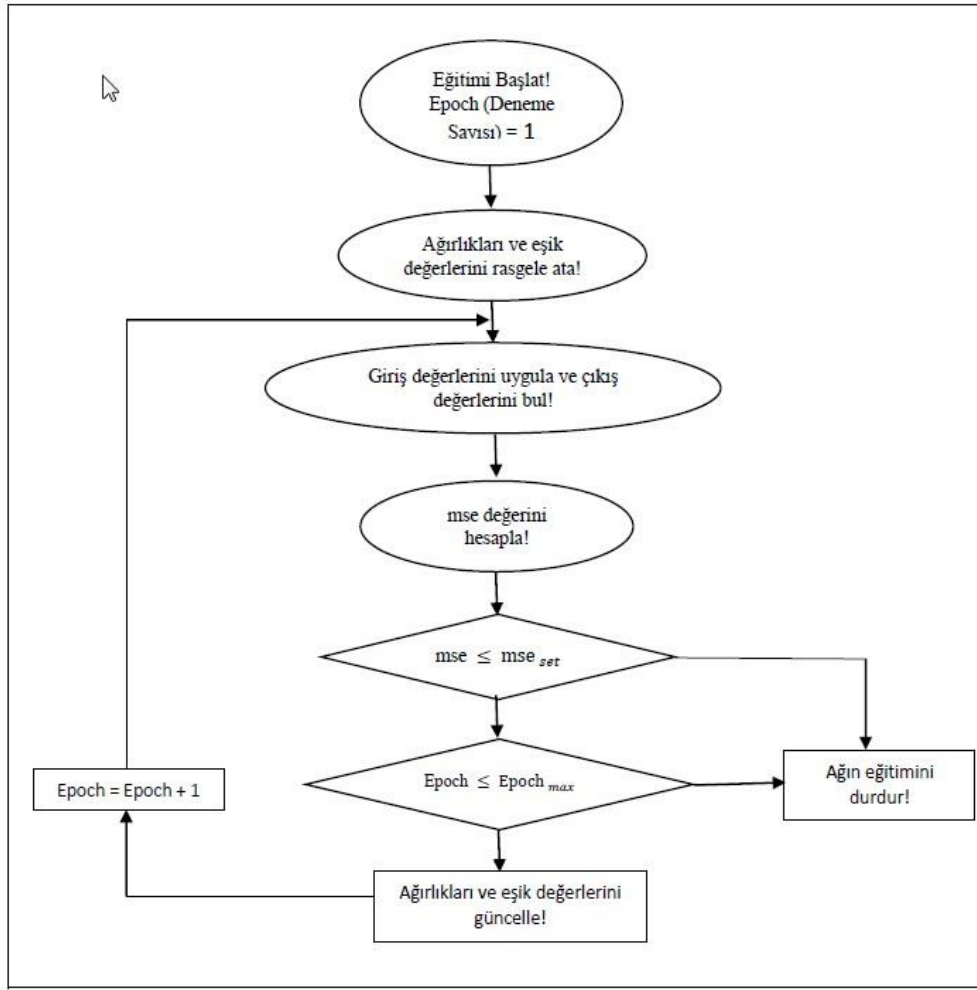
Şekil 13. Geri Yayılım Algoritması

Kaynak: Thiang St ve Khoswanto H., “Artificial Neural Network with Steepest Descent Backpropagation Training Algorithm for Modeling Inverse Kinematics of Manipulator”, World Academy of Science, Engineering and Technology, 2009

Geri yayılım adı, hataların geriye doğru yayılması dolayısıyla konulmuştur. Gradyent (yön türev) iniş (Gradient Descent) metodu kullanır. Bu metot ağ içerisindeki bütün ağırlık değerlerini kullanarak maliyet fonksiyonunun gradyanını hesaplar (Öztemel,

2012). Gradyan, optimizasyon metodu içerisinde ağırlık değerlerini kullanarak maliyet fonksiyonunu minimize etmeye çalışır.

Başlangıçta eğitim öncesinde ağırlık katsayıları rastgele seçilir. Nöronlar eğitim örneklerinden öğrenmeye başlarlar. Örnek olarak x_1, x_2 ve t üçlü setini düşünelim. x_1 ve x_2 giriş değerleri ve t ise çıkış değeri olsun. Bunlar eğitim değerleridir ve olması gereken değerlerdir. Ağa x_1 ve x_2 değerlerini uyguladığımızda t değeri yerine y değerini elde ettiğimizi düşünelim. Bu y değeri ağırlıkları rasgele seçtiğimiz için t değerinden farklı bir değerdedir. Beklenen çıkış t ve gerçek çıkış y arasındaki tutarsızlığı ölçmek için karesel hata metodu kullanılır. Bu değeri bulmak için; $E = (t - y)^2$ formülü kullanılır. Bu formülde E 'ye hata ya da tutarsızlık denir (Öztemel, 2012). Bundan sonraki amaç giriş ve çıkışlar arasındaki haritalama problemini azaltacak bir optimizasyon fonksiyonu bulmaktan geçmektedir. Bu problem aynı zamanda hatayı da en aza indirme problemi olarak gözükmektedir. Tipik geri yayılım işleminin akış diyagramı Şekil 14'de gösterilmiştir.



Şekil 14. Geri Yayılm İşleminin Akış Diyagramı

Kaynak: ÖZTEMEL, E. , “Yapay Sinir Ağları”, Papatya Yayıncılık Eğitim, İstanbul, 2012

1.2.5 Yapay Sinir Ağları Temel Öğrenme Algoritmaları

Öğrenme sistemlerinde kullanılan değişik öğrenme kuralları vardır. Yapay sinir ağlarında öğrenme kurallarının çoğu **Hebbian** kuralına dayanmaktadır.

Belli başlı kurallar aşağıdaki gibidir.

- **Hebbian Öğrenme Kuralı:** Bilinen en eski öğrenme kuralıdır. Biyolojik sistemlerin dinamik sistemleri üzerine kurulmuştur. Diğer öğrenme kuralları bu öğrenme algoritmasının farklılaştırılmış halidir. 1949 yılında neropsikolog Donald Hebb tarafından geliştirilen bu kurala göre, bir hücre (yapay sinir ağı elemanı) diğer bir hücreden bilgi alırsa ve her iki hücrede aktif ise (matematik olarak aynı işareti taşıyorsa) her iki hücrenin arasındaki bağlantı kuvvetlendirilmelidir. Diğer bir deyişle bu kural şu şekilde

özetlenebilir. Bir hücre kendisi aktif ise bağlı olduğu hücreyi aktif yapmaya pasif ise pasif yapmaya çalışmaktadır. Matematiksel olarak şöyle ifade edilir:

$$w_{ij}[n+1] = w_{ij}[n] + \eta x_i[n]x_j[n]$$

Bu denklemde, η öğrenme oranı katsayısıdır. x_i ve x_j elemanların çıkışlarıdır. Hebbian öğrenme kuralı lokal bölgede işletilir ve genel sistemin giriş/çıkış çalışma karakteristiğini dikkate almaz. Bu çalışma tarzı Hebbian Öğrenme Kural'ını biyolojik öğrenme yöntemleri için makul bir teori yapar (Öztemel,2012).

- **Hopfield Öğrenme Kuralı:** Hopfield ağırları verilen eşik değerini aşp aşmamasına göre 2 değer alabilir. Bazı çalışmalarda -1 ve 1 değerleri; bazılarında ise 0 ve 1 değerleri kullanılmıştır. Şekil 15'de tipik bir 4 düğümlü hopfield ağı gösterilmiştir. Formülü aşağıdaki gibidir (Dağ, 2006).

$$s_i \leftarrow \begin{cases} +1 & ; \sum_j w_{ij}s_j \geq \theta_i, \\ -1 & ; \text{diğerleri} \end{cases}$$

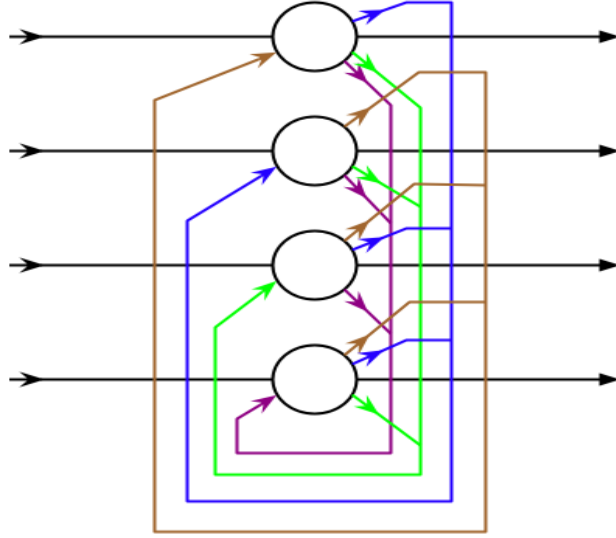
w_{ij} , bağlantılar arasındaki kuvvet ağırlığı

s_j , j. nöronun durumu

θ_i , i. nöronun eşik seviyesi

Şu sınırlamaları vardır:

- $w_{ii} = 0, \forall i$ (Nöronların kendileri ile bağlantıları yoktur.)
- $w_{ij} = w_{ji}, \forall i, j$ (Nöronlar arası bağlantılar simetriktir.)



Şekil 15. Tipik Hopfield Ağı

Kaynak: Yousefpour N., “Artificial Neural Networks: A Tutorial”, Texas A & M University, <https://www.slideshare.net/stellajoseph/artificial-neural-networks-15168117>, Erişim Tarihi: 12.06.2017

Hopfield ağı 2 şekilde güncellenir (Dağ, 2006).

- Asenkron: Birim zamanda tek bir birim güncellenir. Bu birim rastgele alınabilir, ya da önceden tanımlanmış bir düzen başından itibaren uygulanır.
- Senkron: Tüm düğümler aynı anda güncellenir. Bu senkronizasyon sağlamak için sisteme merkezi bir saat gerektirir. Biyolojik ve fiziksel sistemlerin zamanı takip edecek küresel bir saati olmadığından dolayı bu metot gerçekçi değildir.

İki nöron arasındaki ağırlık katsayısı değeri nöronların değerleri üzerinde önemli etkiye sahiptir. Eğer iki nöron arasındaki ağırlık w_{ij} ise ve $w_{ij} \geq 0$ ise;

$s_j = 1$ olduğunda ağırlıklı toplam içinde j 'nin katkısı pozitifdir. Böylece s_i, j tarafından $s_j = 1$ değerine doğru çekilir.

$s_j = -1$ olduğunda ağırlıklı toplam içinde j 'nin katkısı pozitifdir. Böylece s_i, j tarafından $s_j = -1$ değerine doğru çekilir.

Bunların i . ve j . nöron arasındaki ağırlık değeri pozitif ise nöronlar birbirine yakınsar. Benzer şekilde ağırlık değeri negatif ise birbirinden ayrılırlar.

Hopfield ağlarının enerji (E) diye adlandırılan skaler bir değeri vardır. Bu değer:

$$E = -\frac{1}{2} \sum_j w_{ij} s_j s_i + \theta_i s_i \text{ dir (Öztemel, 2012).}$$

. Bu değer enerji diye adlandırılır çünkü birimler güncelleme için rasgele seçildiğinden enerji değeri ya düşer ya da aynı kalır (Öztemel, 2012).

. Dahası bu güncellemeler sonucunda ağ bir minimum noktasına doğru yakınsayacaktır (Öztemel, 2012).

- **Delta kuralı:** Makine öğrenmesinde tek katmanlı sinir ağlarının yapay nöronlarının giriş ağırlıklarının güncellenmesinde kullanılan gradyan inişli öğrenme kuralıdır. Widrow ve Hoff tarafından geliştirilmiştir (Şekil 16). Genel geri yayılım algoritmasının özel bir durumudur. Delta kuralını şöyle tanımlayabiliriz (Sancakdar,2002).

$$\Delta w_{ij} = \alpha (t_j - y_j) g'(h_j) x_i ; h_j = \sum x_i w_{ij} ; y_j = g(h_j)$$

α , küçük değerli katsayı (öğrenme oranı)

$g(x)$, nöron aktivasyon fonksiyonu

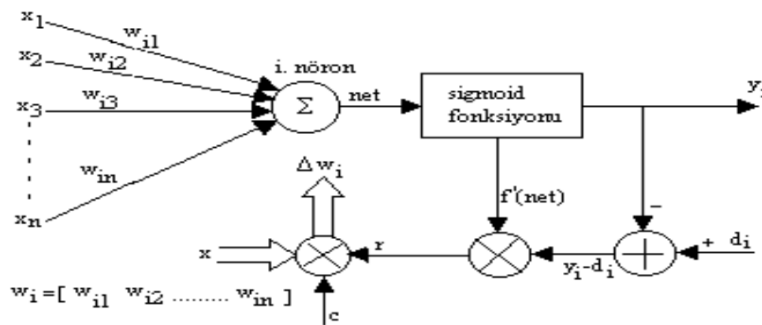
t_j , hedef çıktı

h_j , nöron girişlerinin ağırlıklandırılmış toplamı

y_j , gerçek çıkış

$x_i, i.$ giriş

Beklenen çıktı ile gerçekleşen çıktı arasındaki farklılığı azaltmak için yapay sinir ağının elemanlarının bağlantılarının ağırlık değerlerinin sürekli değiştirilmesi ilkesine dayanarak geliştirilmiştir. Ağın ürettiği çıktı ile üretilmesi gereken (beklenen) çıktı arasındaki hatanın karelerinin ortalamasını en düşüğe çekmesi hedeflenmektedir.



Şekil 16. Delta Öğrenme Kuralı Şematik Gösterimi

Kaynak: Sancakdar, 2002

- **Kohonen kuralı:** Kohonen ağı, Kohonen tarafından 1982 yılında geliştirilmiştir. Danışmansız öğrenme yöntemidir. Görselleştirme ve yüksek boyutlu verilerin analizi için kullanılır. Kendi kendini düzenleyen haritalar (SOM) olarak da bilinir (Takçı, 2017). Bu özellik bilinmeyen girişlere karşı adaptasyon gibi yeni özellikler getirmiştir. Bu süreç doğal olarak beynimizde hiçbir örnek tanımlanmadan çıkarım yapmak gibidir. Bu örnekler öğrenme sürecinde normal çalışma örnekleri ile birleştirilir. Yeni ağ girişler ağı uygulanır ve en iyi sonucu veren vektör kazanan olarak seçilir. SOM'un öğrenme mekanizması aşağıdaki gibi iki şekilde olmaktadır. 1. yöntem dik-dörtgen yöntemi, 2. Yöntem gauss yöntemidir (Takçı, 2017).

$$G(i, x) = \begin{cases} 1 & \text{eğer } d(i, w) \leq \lambda \\ 0 & \text{eğer } d(i, w) > \lambda \end{cases}$$

Bu kurala göre ağı nörönları ağırlıklarını deęiřtirmek için birbirleri ile kıyasıya rekabet halindedirler (Takçı, 2017). En büyük çıktıyı üreten hücre kazanan hücre olmakta ve bağlantı ağırlıkları deęiřtirilmektedir. Bu, o hücrenin yakınındaki hücelere karşı daha kuvvetli hale gelmesi demektir. Hem kazanan elemanların hem de komřuları sayılan elemanların (hücrelerin) ağırlıklarını deęiřtirmesine izin verilmektedir.

1.2.6 Yapay Sinir Ağlarının Eęitilmesi

YSA'nın nörönların bağlantılarının ağırlık deęerlerinin belirlenmesi işlemine “ağın eęitilmesi” denir (Öztemel,2012). Yapay sinir ağlarının öğrenme sürecinde dış ortamdan girişler alınır, aktivasyon fonksiyonundan geçirilerek bir çıkış üretilir. Şekil 8'de tipik bir yapay sinir ağı yapısı görölmektedir. Bu çıkış olması gereken çıkış deęeri ile karşılaştırılarak hata bulunur. Çeřitli öğrenme algoritmalarıyla hata azaltılıp gerçek çıkış deęerine yaklaşılmaya çalışılır. Bu eğitim sırasında YSA hücre elemanlarının ağırlık katsayıları ayarlanarak en uygun çıkış deęeri yakalanmaya çalışılır. Ağırlıklar her bir çevrimde deęiřtirilerek en uygun çıkışa ulaşılmaya çalışılır. Amaca ulaşmanın ölçüsü dışarıdan verilen ve kabul edilebilir varsaydığımız bir hata deęeridir. Eęer yapay sinir ağı verilen giriş ve çıkış çiftleriyle amaca ulaşmış ise ağırlık deęerleri saklanır. Ağırlıkların sürekli güncellenerek istenilen neticeye ulaşılan kadar geçen zamana öğrenme zamanı adı verilir (Öztemel,2012). Yapay sinir ağı öğrenme fazını bitirdikten sonra daha önce

verilmeyen girişler verilip, sinir ağı çıkışlarının gerçek çıkış değerlerine uygunluğu incelenir. Eğer yeni verilen örnekler için gerçek çıkış değerlerine hata oranında yakın çıkışlar elde ediliyorsa sinir ağı işi öğrenmiş yani öğrenme tamamlanmış demektir. Bazı durumlarda eğitilen ağ eğitim setindeki bütün örneklerle % 100 doğru cevap üretmesine rağmen test setindeki örneklerle doğru cevaplar üretememektedir. Bu durum ile karşılaştığında ağın iyi ve öğrenme setini ezberlediği sonucu çıkarılır. Genelde eldeki örneklerin %80'i ağa verilip ağ eğitilir. Daha sonra kalan %20'lik kısım ile ağın davranışı yani hata oranı bulunmaya çalışılır (Çayıroğlu,2015). Böylece ağın testi yapılmış olur.

1. **Örneklerin toplanması:** Ağın öğrenmesi istenilen olay için daha önce gerçekleşmiş örneklerin bulunması adıdır. Ağın eğitilmesi için eğitim seti örneklerinin toplandığı gibi ağın test edilmesi için de test seti örneklerin toplanması gerekmektedir. Eğitim setindeki örnekler ağa uygulanarak olayı öğrenmesi sağlanır. Ağ olayı öğrendikten sonra test setindeki örnekler gösterilerek ağın performansı ölçülür. Hiç görmediği örnekler karşısındaki başarısı ağın iyi öğrenip öğrenmediğini ortaya koyar.
2. **Ağın topolojik yapısının belirlenmesi:** Öğrenilmesi istenen olay için oluşturulacak olan ağın yapısı belirlenir. Girdi, ara katman, ara katmandaki nöronu ve çıktı eleman sayısı bu adımda belirlenmektedir. Giriş ve çıkış elemanları işe göre değişmekle birlikte ağın nöron ve katman sayısı denemeler yolu ile belirlenmektedir.
3. **Öğrenme parametrelerinin belirlenmesi:** Ağın öğrenme katsayısı, işlem elemanlarının toplama ve aktivasyon fonksiyonları, momentum katsayısı gibi parametreler bu adımda belirlenmektedir.
4. **Ağırlıkların başlangıç değerlerinin atanması:** İşlem elemanlarını birbirlerine bağlayan ağırlık değerlerinin ve eşik değer ünitesinin ağırlıklarının başlangıç değerlerinin atanması yapılır. Başlangıç değerleri için rasgele değerler atanır. Ağın uygun değerleri öğrenme sırasında ağ tarafından belirlenir.
5. **Öğrenme setinden örneklerin seçilmesi ve ağa gösterilmesi:** Ağın öğrenmeye başlaması ve öğrenme kuralına uygun olarak ağırlıkları değiştirmesi için ağa örnekler belirli bir düzeneğe göre gösterilir. Genellikle normalize edilmiş değerler ağa gösterilir. Sunulan girdi için ağın çıktı değerleri hesaplanır.
6. **Gerçekleşen çıktının beklenen çıktı ile karşılaştırılması ve ağırlıkların değiştirilmesi:** Ağın ürettiği hata değerleri bu adımda hesaplanır. Geri hesaplama

yöntemi uygulanarak üretilen hatanın azalması için ağırlıkların değiştirilmesi yapılır.

7. **Öğrenmenin tamamlanması:** İleri beslemeli sinir ağı öğrenmeyi tamamlayınca, yani gerçekleşen ile beklenen çıktılar arasındaki hatalar kabul edilir düzeye ininceye kadar devam eder (Çayıroğlu,2015).

Ağın kendisine gösterilen girdi örneği için beklenen çıktıyı üretmesini sağlayacak ağırlık değerleri başlangıçta rastgele atanmakta ve ağı örnekler gösterildikçe ağırlıklar değiştirilerek istenen değerlere ulaşması sağlanmaktadır. İstenen ağırlık değerlerinin ne olduğu bilinmemektedir. Değerlerin bilinmemesinden dolayı YSA davranışlarını anlamak mümkün değildir. Bazı durumlarda YSA'nın eğitimi bitirdiği yer hata düzeyinin üstünde kalabilir. Bu durumda ağın olayı öğrenmesi için bazı değişiklikler yapılarak yeniden eğitilmesi gerekir. Bu değişiklikler sonucunda YSA başlangıç parametre değerlerinde farklılıklar oluşur. Ağ topolojisinde değişiklikler yapılabilir. YSA'ya gösterilen örneklerin değerleri değiştirilerek yeni örnek seti oluşturulabilir. Öğrenme setindeki verilerin sayısı artırılabilir veya ezberleme olmaması için azaltılabilir. İleri beslemeli YSA'larda lokal sonuçlara takılıp kalınmaması için momentum katsayısı diye bir parametre geliştirilmiştir. YSA'ların eğitilmesinde önemli bir sorun ise her veri çok fazla ise hafıza sorunu ortaya çıkması ve öğrenme süresinin çok uzun olmasıdır. Ağırlık değerleri başlangıçta büyük değerler olması durumunda ağın yerel sonuçlara düşmesi ve bir yerel sonuçtan diğerine sıçramasına sebep olmaktadır. Eğer ağırlıklar küçük aralıkta seçilirse o zaman da ağırlıkların doğru değerleri bulması uzun zamanlar almaktadır. Bazı problemlerin çözümü sadece 200 iterasyon sürerken bazıları 5-10 milyon iterasyon sürmektedir. Ağın öğrenmesinin gösterilmesinin en güzel yolu hata grafiğini çizmektir. Her bir iterasyonda hata değerinin düştüğü grafik ekrandan takip edilebilir. Önceden belli olmayan bir iterasyon değerine geldiğinde hatanın azalmadığı veya azalsa bile çok fazla değişmediği görülür. Bu daha iyi bir sonuç bulunamayacağı anlamına gelir. Artık ağın öğrenme işleminin durmuştur. Eğer elde edilen çözüm kabul edilemez ise o zaman ağ yerel bir çözüme takılmış demektir (Çayıroğlu,2015).

Katsayıların Belirlenmesi: İleri beslemeli sinir ağında bağlantıların ağırlık değerlerinin başlangıçta belirlenmesi ağın performansını yakından ilgilendirmektedir. Genel olarak ağırlıklar belirli aralıklarda atanmaktadır. Bu aralık eğer büyük tutulursa yerel çö-

zümleler arasında sürekli dolaştığı küçük olması durumunda ise öğrenmenin geç gerçekleştiği görülmüştür. Bu değerlerin atanması için henüz standart bir yöntem yoktur. Tecrübeler 0.1 ile 1.0 arasındaki değerlerin başarılı sonuçlar ürettiğini göstermektedir. Fakat bu tamamen öğrenilmesi istenen problemin niteliğine bağlıdır. Başlangıç değerleri kadar öğrenme ve momentum katsayılarının belirlenmesi de ağırlık öğrenme performansı ile yakından ilgilidir. Öğrenme katsayısı ağırlıkların değişim miktarını belirlemektedir. Eğer büyük değerler seçilirse o zaman yerel çözümler arasında ağırlık dolaşması ve osilasyon yaşaması söz konusu olmaktadır. Küçük değerler seçilmesi ise öğrenme zamanını artırmaktadır. Tecrübeler genellikle 0.2 - 0.4 arasındaki değerlerin kullanıldığını göstermiştir (Çayıroğlu,2015). Bazı uygulamalarda ise öğrenme katsayısı 0.6 değeri alındığı zaman en başarılı sonuçların oluştuğu gözlenmiştir. Bu durum tamamen probleme bağlıdır. Benzer şekilde momentum katsayısı da öğrenmenin performansını etkiler. Momentum katsayısı bir önceki iterasyondaki değişimin belirli bir oranının yeni değişim miktarına eklenmesidir. Bu özellikle yerel çözümlere takılan ağırlıkların sıçrama ile daha iyi sonuçlar bulmasını sağlamak amacıyla önerilmiştir (Ersan, 2007). Momentum katsayısının küçük seçilmesi lokal optimum noktalarına takılmayı sağlar ve çözümü zorlaştırabilir. Çok büyük değerler ise çözüme ulaşmada sorunlar yaşatabilir. Tecrübeler momentum katsayısının 0.6-0.8 arasında seçilmesinin uygun olacağını göstermiştir. Problemin niteliğine göre kullanıcının seçmesinde fayda vardır.

1.3 Çelik Genel Yapısı, Alaşımlama ve Tavlama

Üretimde kullanılacak çelik malzemelerin doğru seçimi üretimin doğru ve hızlı yapılabilmesi, uzun ürün ömrü ve müşteri memnuniyeti açısından çok önemlidir. Seçilecek malzemelerde aşağıdaki konular göz önüne alınır (Şahin,2012):

- Ürün hangi gerilmeler altındadır? (Gerilmenin büyüklüğü, statik veya dinamik oluşu)
- Ürün hangi ortamda (Hava, su, yağ, asit gibi) ve sıcaklıkta çalışacaktır?
- Sürtünme veya aşınma var mıdır?
- Üretim metotları nelerdir? (döküm, şekil verme gibi)

1.3.1 Çelik Genel Yapısı ve Alaşımlama

Yukarıdaki koşullar göz önüne alınarak piyasada bulunan çelik malzemelerinden mukavemeti, sertliği, tokluğu, çekme değeri, esnekliği, yüksek ısıya dayanma ve korozyon direnci değeri uygun özelliklere sahip malzemelerden en uygun olanın seçilmesi sağlanacaktır. Bu özelliklerin çeliğe verilebilmesi için saf demirin bazı malzemeler ile alaşımlanması ve daha sonra ısıl işleminden geçirilmesi gerekmektedir. Isıl işlemin daha iyi anlaşılabilmesi için demirin ve alaşımların demir içerisindeki davranışlarını iyice anlamamız gerekmektedir. Bu nedenle ısıl işleme başlamadan önce demirin yapısı, alaşımlanması ve demir ve alaşımların ısı karşısındaki davranışları hakkında kısaca bilgi verilecektir.

Saf demir mukavemetinin düşük olması nedeniyle teknolojik öneme sahip değildir. Fakat yine hiçbir öneme sahip olmayan karbonla bir araya getirilirse çelik ve dökme demir adı altında iki alaşım meydana gelir. Alaşım, bir metal elementinin en az bir başka element ile homojen karışımıdır. Çelik, demir ve karbon alaşımının içerisinde en fazla % 2 karbon, % 1 mangan, % 0,5 silisyum ve % 0,05 kükürt ve fosfor bulunan alaşımdır (Demirhan,2016).

Kimyasal bileşimine göre çelikler aşağıdaki alt gruplara ayrılır (Demirhan,2016):

- **Sade Karbonlu Çelikler:** İçinde yalnızca karbon bulunan çeliklerdir. Şu şekilde sınıflandırılır.
 - **Ötektik Altı Çelikler;** içerisinde % 0,85'den az karbon bulunan çeliklerdir.
 - Yumuşak çelikler (% 0,1 – 0,2 karbonlu çelikler)
 - Az karbonlu çelikler (% 0,2 – 0,3 karbonlu çelikler)
 - Orta karbonlu çelikler (% 0,3 – 0,85 karbonlu çelikler)
 - **Ötektik Üstü Çelikler;** içerisinde % 0,85'den fazla karbon bulunan çeliklerdir.
- **Alaşımlı Çelikler:** İstenilen özelliklere sahip olabilmeleri için içlerine alaşım elemanları eklenerek elde edilen çeliklerdir. Alaşımlı çelikler de kendi aralarında şu şekilde sınıflandırılır.
 - **Basit Alaşımlı Çelikler:** Bu tip çeliklerde katkı elemanı olarak karbondan başka yalnızca bir eleman vardır. İçerisinde nikel varsa nikelli çelik, krom varsa kromlu çelik gibi adlar alırlar.

- **Çift Alaşımli Çelikler:** Bu çeliklerde karbondan başka iki çeşit katkı elemanı vardır. Buna göre adlandırılırlar. Örnek: Krom-Nikel Çelik.
- **Çok Alaşımli Çelikler:** Bu çeliklerde katkı eleman sayısı sınırlı değildir. Çelikte istenildiği kadar katkı elemanı bulunabilir (Demirhan,2016).

İngiliz Harry Brearly,1913 yılında tüfek namlusu geliştirme projesi üzerinde çalıştığı esnada karbon çeliğe krom eklemenin çeliğin sağlamlık ve dayanıklılığını arttırdığını keşfeder yani paslanmaz çelik üretir. Çeliğin paslanmaya karşı olan direnci ve diğer çelik çeşitlerinden daha az paslanması, minimum % 10,5 krom ilavesiyle mümkün olur (Akdoğan,2008). Demir-karbon alaşımına ek olarak konulan elementlerin çeliğe yaptığı katkılar aşağıda listelenmiştir.

Karbon: Çelik için temel alaşım elementidir. Karbon oranının artmasıyla sertlik ve dayanıklılık özelliği önemli ölçüde artar. % 0,8 karbon oranına kadar çeliğin çekme gerilimi değeri ve akma sınırı değeri artar. Karbon miktarının artması aynı oranda şekil değiştirebilme (süneklik), dövülebilirlik, derin çekilebilirlik ve kaynak yapılabilme kabiliyetini düşürür. Yüksek karbonlu çeliklerin ısıtılmasında çatlama riski de fazladır (Akdoğan,2008).

Mangan: Çeliğin sünekliği azaltmakla birlikte dayanımını (Malzemenin kırılma olmadan taşıyabileceği en yüksek gerilme) artırma özelliğine sahiptir. Yüzde 3 değerine kadar, her yüzde 1 mangan için çekme dayanımı yaklaşık 100 mpa kadar artar. Yüzde 3 ile yüzde 8 arası artış oranı azalır. Yüzden 8'den itibaren dayanımda düşme görülür. Çeliğin dövülebilirliğini ve sertleşebilirliğini iyileştirici özelliktedir. Kaynak yapılabilme yeteneğini azaltmaz ve kaynaklanabilir malzemeler için yüzde1,6 oranına kadar yükseltilebilir. Çeliğin içindeki karbon oranının artması manganın olumlu yöndeki etkisini artırır (Akdoğan,2008).

Silisyum: Çelik üretimi esnasında deoksidan (erimiş çelikte bulunan demir oksidin oksijenini alan element) olarak kullanılır. Döküm çeliklerde, döküme akıcılık sağlamak için ilave edilebilir. Ferrit (Çok az miktarda C içeren, ancak pratikte saf demir (Fe) olarak kabul edilen kübik hacim merkezli şekil alabilen, kısmen esnek, manyetik kristallerdir.) içerisinde çözünebilme özelliğine sahip olduğu için malzemenin süneklik ve tokluğunu düşürmeden, dayanımı ve sertliği artırır. Yüksek silis içeren çeliklerin ısı dayanımı da yüksektir. Genel olarak sertleşebilirliği, aşınma dayanımını ve

elastikiyeti yükseltmesine karşın yüzey kalitesini olumsuz yönde etkiler (Akdoğan,2008).

Kükürt: Demir ile birlikte FeS bileşiği oluşturarak, tane sınırlarında birikerek malzemenin gevrek (kırılgan) olmasına yol açar. Bu nedenle çelik için çok zararlı bir elementtir ve giderilmesi yönünde çalışılır (Erişir,2013).

Nikel: Nikel alaşımlı paslanmaz çeliklerde yüzde 5'e varan oranlarda kullanılır. Nikel çeliğin mukavemetini ve tokluğunu (çeliğin kopma noktasına kadar biriktirdiği enerji) artırır. Nikelin çelik tanelerinin hacmini küçültme etkisine de sahiptir.

Molibden: Molibden düşük nikel ve krom alaşım malzemeleri bulunduran çeliklerde temper gevrekliğini (düşük tavlama sıcaklıklarında oluşan süneklik azalması) azaltmak için kullanılır (Akdoğan,2008).

Volfram: Çelik dayanımını artıran özelliği olan bir alaşım elementidir (Akdoğan,2008).

Titanyum: Kuvvetli karbür (karbonun bir element ile birleşme sonucu oluşturduğu yeni element) yapıcı özelliği vardır ve sertliği artırır. Çelik üretimi esnasında deoksidan olarak da kullanılır. Tane inceltici yapıya sahiptir (Akdoğan,2008).

Krom: Krom paslanmaz çeliklerin temel alaşım elementidir. Krom, korozyon (aşınma) ve oksidasyon (siyah oksit tabakasının oluşması) direnci sağlar. Sertleşebilme kabiliyetini artırır. Yüksek karbonlu çeliklerde korozyon direncini yükseltir. Krom karbon ile tane sınırlarında biriken bileşeni ($Cr_{23}C_6$) oluşturur. Oluşan bu bileşik paslanmaz çeliklerde tane sınırlarındaki krom miktarını paslanmazlık sınırı olan yüzde 12 'nin altına çeker (Akdoğan,2008)

Alaşım elementlerinin etkilerini gösteren Tablo 10 aşağıda sunulmuştur.

Tablo 10
Alaşım Elementlerinin Çeliğe Etkisi

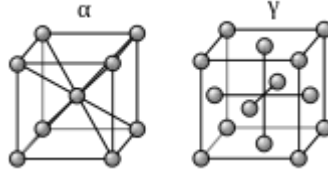
	Si	Mn*	Mn**	Cr	Ni*	Ni**	Al	W	V	Co	Mo	S	P
Sertlik	artar	artar	azalır	artar	artar	azalır	-	artar	artar	artar	artar	-	artar
Mukavemet	artar	artar	artar	artar	artar	artar	-	artar	artar	artar	artar	-	artar
Akma Noktası	artar	artar	azalır	artar	artar	azalır	-	artar	artar	artar	artar	-	artar
Uzama	değişmez	değişmez	azalır	azalır	değişmez	artar	-	azalır	değişmez	azalır	azalır	azalır	azalır
Kesit Büzülmesi	azalır	değişmez	değişmez	azalır	değişmez	artar	-	azalır	değişmez	azalır	azalır	azalır	azalır
Darbe Direnci	artar	değişmez	-	azalır	değişmez	artar	azalır	-	artar	azalır	artar	azalır	azalır
Elastisite	artar	artar	-	artar	-	-	azalır	-	artar	-	-	-	-
Sıcaklığa Dayanım	azalır	değişmez	-	artar	artar	artar	-	artar	artar	artar	artar	-	-
Soğutma Hızı	azalır	azalır	azalır	azalır	azalır	azalır	-	azalır	azalır	artar	azalır	-	-
Karbür Oluşumu	azalır	değişmez	-	artar	-	-	-	artar	artar	-	artar	-	-
Aşınma Direnci	azalır	azalır	-	artar	azalır	-	-	artar	artar	artar	artar	-	-
Dövülebilirlik	azalır	artar	azalır	azalır	azalır	azalır	azalır	azalır	artar	azalır	azalır	azalır	azalır
İşlenebilirlik	azalır	azalır	azalır	-	azalır	azalır	-	azalır	-	değişmez	azalır	artar	azalır
Oksitlenme Eğilimi	azalır	değişmez	azalır	azalır	azalır	azalır	azalır	azalır	azalır	azalır	artar	-	azalır
Korozyon Direnci	-	-	-	azalır	-	artar	-	-	artar	-	-	azalır	artar

Kaynak: <http://www.omurcelik.com/teknik-bilgiler/alasim-elementlerinin-celiklere-etkisi>

Alaşım malzemelerin özellikleri içerdikleri fazların sayısına ve içeriklerine göre de değişmektedir. Malzemelerdeki faz, farklı kristal yapıya sahip bölgeler olarak adlandırılmaktadır. Bu bölgeler mekanik olarak birbirinden ayırt edilebilmekte ve fiziksel ve kimyasal özellikleri de değişmemektedir (Öztürk, 2016).

Çok özel bir şekilde bir araya gelmiş atomlardan oluşan ve aralarında metalik bağ bulunan kristal yapılara “Kafes” adı verilir. Elementlerin yapılarında meydana gelen bu

tekrar düzenlenmeler basınç veya sıcaklık gibi değişkenler sayesinde olur. Bu tekrar düzenlenme işlemine allotropi veya polimorfizm denir. Alotropi, kimyasal özellikleri benzer, fiziksel özellikleri ve kristallerinin dizilişleri birbirinden farklı olan aynı cins atomlardan oluşan kafeslerdir. Alfa demir (düşük sıcaklık - Hacim Merkezli Kübik) ve gama demir (yüksek sıcaklık - Yüzey Merkezli Kübik) allotropi kafes yapıları arasındaki fark Şekil 17’de verilmiştir (Öztürk, 2016).

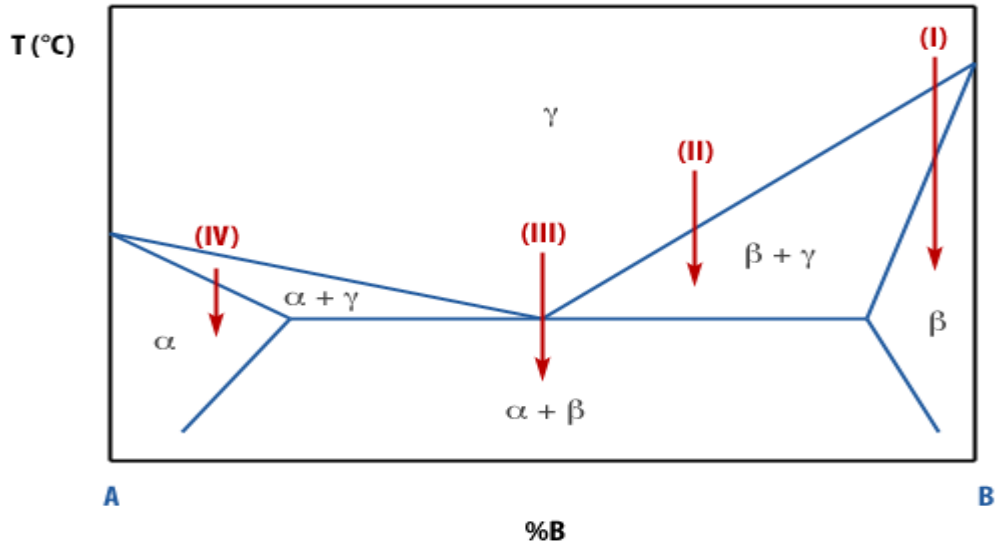


Şekil 17. Allotropi Yapı

Kaynak: Öztürk, 2016

Kristal yapıdaki metallerde faz dönüşümleri üç durumda gerçekleşir. (Şekil 18):

- **Faz sayısında değişiklik olmayan yayınmalı dönüşümler** (saf metalin ergimesi veya katılaşması, allotropi dönüşümler, yeniden kristalleşme gibi.). Yeniden kristalleşme, deformasyona uğramış tanelerin tamamının yeni çekirdeklenen ve büyüyen tanelere dönüşmesini sağlayan bir işlemdir. Yeniden kristalleşme, soğuk işlem görmüş metallere uygulanıp, faz değişikliği olmadan yeni tanelerin çekirdeklenmesi ve büyümesi için yapılır. Yoğun soğuk biçimlendirmeye uğramış metallerde oluşan plastik deformasyon sonucu oluşan hasarları düzeltmek için uygulanan bir ısıl işlemdir (Öztürk, 2016).



Şekil 18. Faz Dönüşüm Eğrisi (Ötektoit)

Kaynak: Mühendishane,2017

- **Faz sayısında değişiklik olan yayınmalı dönüşümler (Ötektoit dönüşümler)**

- $\gamma \rightarrow \beta$ dönüşüm gerçekleşiyor. γ yüzey merkezli kafes, β ise hacim merkezli kafes yapısına dönüşüyor. Şekil 18'deki I. çizgi bu tür bir dönüşümü gösteriyor.
- $\gamma \rightarrow \beta + \gamma$ türü dönüşümdür. Çökelme (precipitation) denir. Şekil 18'deki II. çizgi bu tür bir dönüşümü gösteriyor.
- $\gamma \rightarrow \alpha + \beta$ türü dönüşümdür. Yeni iki faz oluştu. Ana faz tamamıyla yok oldu. Bu tür faz dönüşümüne ayrışma (decomposition) denir. Şekil 18'deki III. çizgi bu tür bir dönüşümü gösteriyor.
- $\alpha + \gamma \rightarrow \alpha$ türü dönüşümdür. Çözünme (desolution) denir Şekil 18'deki IV. çizgi bu tür bir dönüşümü gösteriyor (Öztürk, 2016).

- **Yayınmaz dönüşümler (martenzitik dönüşümler)**

Atomların kristal içinde uzun mesafeler boyunca yayılmadan, sadece toplu halde konum değiştirmeleriyle gerçekleşir. Alman metalurjist Adolf Martens 'in çalışmalarına ithafen martenzit dönüşümleri olarak da adlandırıyoruz. Çeliklerde martenzit dönüşümleri, yüksek sıcaklıkta kararlı östenit fazının hızlı soğutulması sonucunda atomların topluca hareket ederek martenzit fazını ortaya çıkarmasıyla gerçekleşiyor (Öztürk, 2016).

- **Faz sayısında deęişiklik olmayan yayınmalı dönüşümler**

Çok yavaş gerçekleşir ve son yapıları ısıtma/soğutma hızına baęlıdır. Sıcaklığın yanında zamana da baęlıdır.

Faz diyagramları malzeme biliminde bir yada daha fazla sayıda saf elementin veya bileşimin sıcaklık ve basınç gibi deęişkenlerin deęişmesi sonucunda yapının ne hale geleceğini gösteren grafiklerdir (Mühendishane,2017). Faz diyagramları yavaş soğumalar sonucu elde edilen denge koşullarının gösterimidir. Demir – karbon (Fe-C) faz denge diyagramı Şekil 19’da verilmiştir. Grafikten de anlaşılacağı gibi yüzde 2 karbondan az bileşen içeren bileşime Çelik, yüzde 2’den fazla karbon içeren bileşene dökme demir denir. Çeliklerde yüzde 0,8’den az karbon içeren bileşene ötektoit altı çelik ve yüzde 0,8’den fazla karbon içeren bileşene ötektoit üstü çelik olmak üzere ikiye ayrılırlar.

A1 sıcaklığı: 723 °C derece boyunca uzanan çizgidir. Bu sıcaklık birçok faz dönüşümü için önemli bir sıcaklıktır. Ötektoit sıcaklık çizgisi olarak bilinir. Bu sıcaklığın altında demir hacim merkezli kübik (HMK) yapıda bulunurken, bu sıcaklığın üstünde yüzey merkezli kübik (YMK) yapıda bulunur (Mühendishane,2017).

A2 sıcaklığı: 760 °C’de bulunur. Çelik, bu sıcaklığın altında demir manyetik yapıdadır. Bu sıcaklığın üzerinde paramanyetik yapıdadır. Paramanyetik, manyetik alan içerisine yerleştirildiğinde geçici olarak manyetik özellik kazanan maddelere denir (Mühendishane,2017).

A3 sıcaklığı: 910 °C ‘den ötektoit noktasına doğru çizilen eğri çizgidir. Östenit bölgesini sınırlandırır. Bu eğrinin altında östenit, östenit + ferrite dönüşür (Mühendishane,2017).

Ötektoit nokta: Faz diyagramında 723 °C (A_1) ve % 0,8 karbonun kesiştiği noktadır. Östenit bölgesinden soğumaya başlayan çelik bu noktaya geldiğinde tek fazlı (γ) yapıdan çift fazlı ($\alpha + Fe_3C$) yapısına dönüşür (Mühendishane,2017). Bu noktadan itibaren östenit, ferrit ve yüksek miktarda karbon içeren sementit fazına dönüşür. Bu dönüşüm esnasında ferrit ve sementit lamelli bir yapı oluşturur ve bu yapı perlit olarak adlandırılır. Dönüşüm izotermal olarak gerçekleşir. Katı fazlar arasında geçiş yaşanır (Özkan,2017). Ötektoit nokta Şekil 18’deki I olarak gösterilen noktadır.

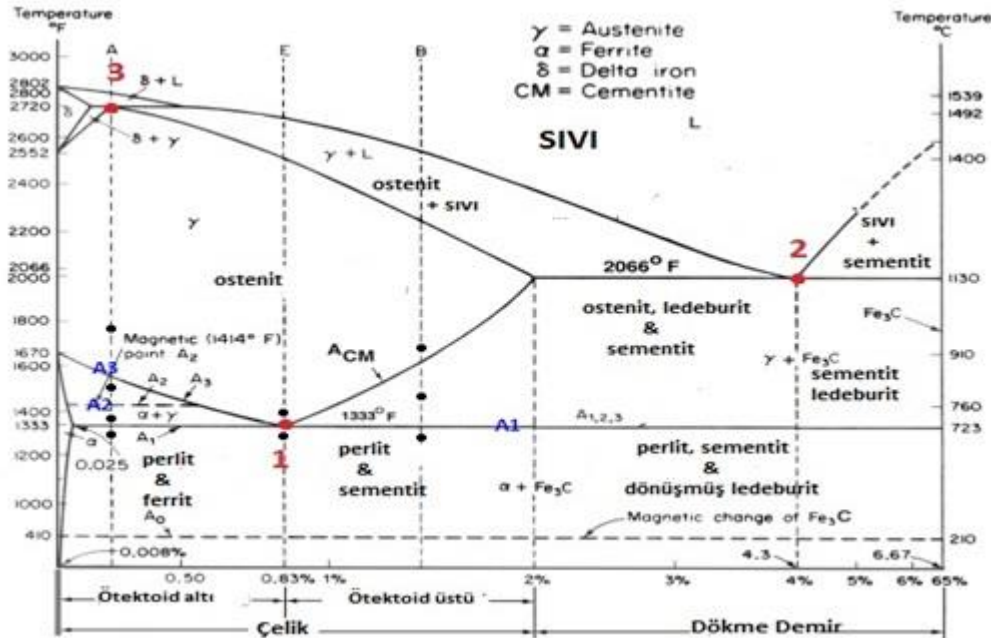
Ötektik nokta: 1130 °C ve % 4,3 karbon içeren dökme demirin kesiştiği noktada bulunur. Bu noktada sıvı faz katılarak iki ayrı katı fazdan ($\gamma + \text{Fe}_3\text{C}$) oluşan ledeburite dönüşür. Dönüşüm izotermal olarak gerçekleşir. Sıvı fazdan katı faza dönüşüm söz konusudur. Ötektik nokta Şekil 18'deki II olarak gösterilen noktadır (Özkan,2017).

Peritektik nokta: Grafikte sıcaklık değeri 1492 °C ile karbon değeri yüzde 0.25 olan eğrinin kesiştiği noktada bulunur. Bu noktada üç faz ($\delta + \gamma + \text{sıvı}$) dengede bulunur. Sıvı faz önceden meydana gelen bir katı fazla reaksiyona girerek yeni bir katı faz oluştururlar yani sıvı faz + δ demir, östenite dönüşür. İzotermal bir dönüşüm söz konusudur. Peritektik nokta Şekil 18'deki III olarak gösterilen noktadır (Özkan,2017).

Yukarıda sözü edilen dönüşümlerin açıklamaları aşağıdaki gibidir.

Perlit ($\alpha + \text{Fe}_3\text{C}$)

Ötektik bileşime sahip çeliğin östenit bölgesinden yavaş soğutulmasıyla 723°C sıcaklıkta meydana gelen ötektik dönüşüm sonucu elde edilir. Perlit ferrit ana yapısı üzerine sementit katmanların dizilmesiyle oluşan mikro yapıdır. Yüzde 12 sementit , yüzde 88 ferritten oluşur. Perlit, demir ve çelik teknolojisinde çok önemli bir yere sahiptir. Uygun ısıl işlem ile bütün çeliklerde oluşturulabilir. Yüzde 0,8 karbon içeren çeliğin oda sıcaklığı şartlarındaki mikro yapısı yüzde yüz perlitir (Özkan,2017) (Şekil 19).



Şekil 19. Fe-C Denge Diyagramı

Kaynak: Metalürji ve Malzeme Platformu, “Demir Karbon Denge Diyagramı”, <http://www.metalurjik.net/tag/demir-karbon-diyagrami-nasil-cizilir>

Sementit (Fe_3C)

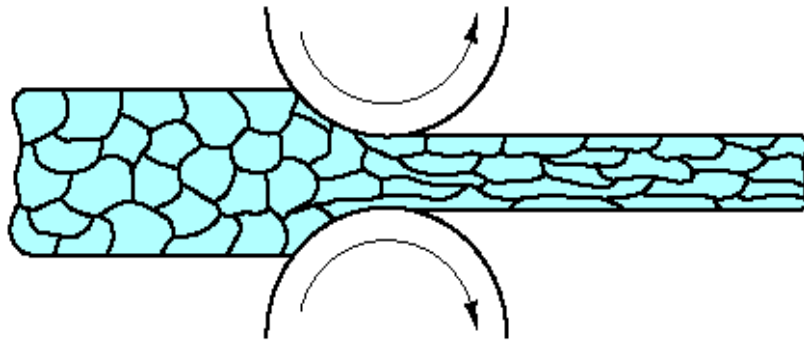
Demir karbit olarak bilinir. Maksimum yüzde 6.67 karbon bulundurur. İntermetalik (metal-metal bileşimi) bir bileşiktir. Bu nedenle düzenli bir kristal yapısı vardır. Sert ve kırılgan bir yapısı vardır. 215°C altında ferromanyetiktir. Basma dayanımı yüksektir. Ticari çeliklere sertleştirme özelliği kazandırılmasında önemli rol oynar (Özkan,2017) (Şekil 19).

Martenzit

Östenitin hızlı soğutulmasıyla elde edilir. Kristal yapısı hacim merkezli dörtgen şeklindedir. Karbona aşırı doymuştur. Yüzde 0,6'den daha fazla karbon içeriyorsa hacim merkezli kübik yapıda bulunur. Martenzit fazı oldukça sert ve kırılgan bir fazdır. Mukavemet değeri oldukça yüksektir (Özkan,2017) (Şekil 19).

1.3.2 Tavlama

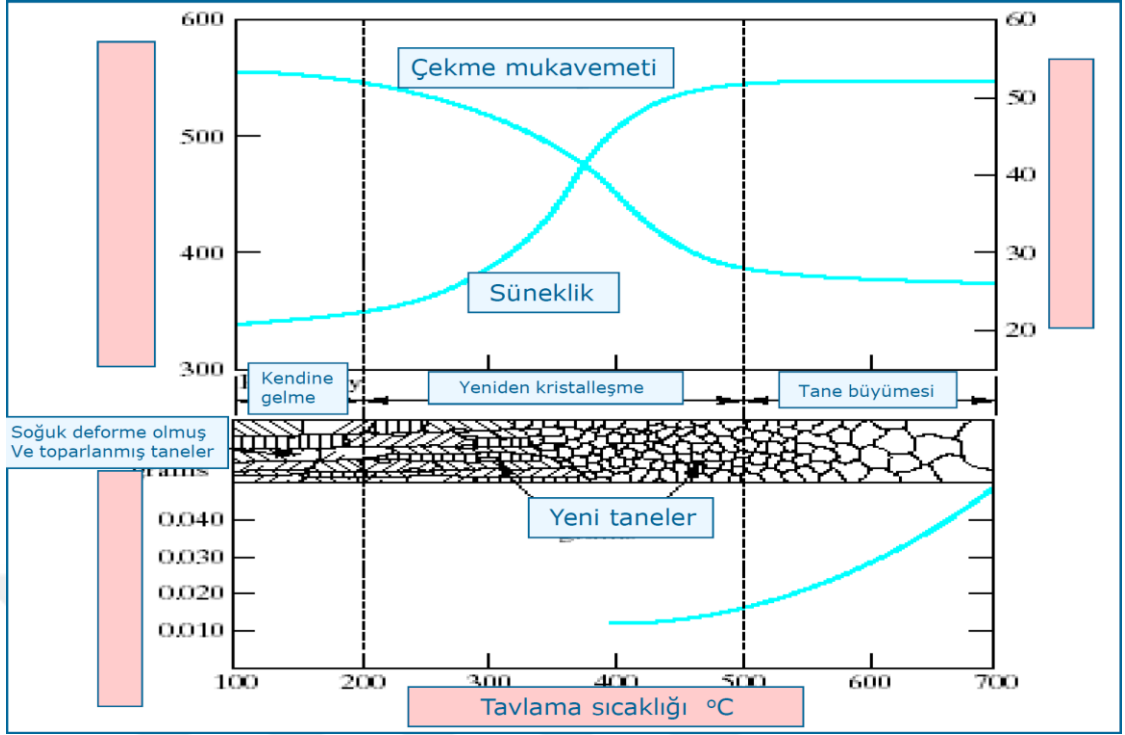
Yukarıda anlatılan özelliklerin üretim aşamasında malzemeye verilebilmesi iki şekilde olur. Çelik sıvı halden katı hale geçişi sırasında birincil kristalleşme gerçekleşir ve bu sırada oluşan katılaşma içyapısı (döküm içyapısı) birincil içyapı olarak adlandırılır. Isıl işlemle (örneğin tavlama) belirli derecelere ısıtılan malzeme katı duruma geçerken tekrar kristalleşebilir. Bu şekilde ikincil yapı ortaya çıkar (Karaboğa,2011). Çelik üretildikten sonra piyasada kullanılabilmesi için birçok işlemden daha geçmektedir. Bunlardan biri olan soğuk haddeleme sırasında çelik yapısında düzensizlikler oluşur (Şekil 20).



Şekil 20. Çelik Yapısındaki Deformasyon

Kaynak: Metalurji ve Malzeme Platformu, 2013

Soğuk haddeleme sırasında metalin üzerinde oluşan düzensizlikler ısıl işlem yöntemiyle düzeltilebilmektedir. Isıyı bağlı olarak çelik moleküler yapısının değişimi Şekil 21'de verilmiştir.



Şekil 21. Çelik İçindeki Moleküler Yapının Isı ile Değişimi

Kaynak: Çalışkan, 2017

Metal metal bir parçanın özelliklerinde belirgin bir değişiklik sağlamak amacıyla metal veya alaşımların katı halde ısıtılıp soğutulması işlemlerine “ısıl işlem” denir. (Metalurji ve Malzeme Platformu, 2013). En çok metalürjik uygulamalarda kullanılır. Isıl işlem, cam gibi daha birçok malzemenin özelliklerinin değiştirilmesinde de kullanılır. Isıl işlem, malzemeyi arzu edilen derecelerde sertleştirmek ve yumuşatmak için yüksek sıcaklıklara ısıtma ve soğutma işlemlerini içerir. Isıl işlemin (tavlama) beş çeşidi vardır (Khaira,2013).

Tavlama, soğuk haddeleme sırasında oluşan deformasyonun etkisini yeniden kristalleştirme yolu ile ortadan kaldırmak için kullanılır (Khaira,2013). Tavlama kullanıldığı yerlere göre aşağıdaki gibi farklılıklar gösterir (Akdoğan (2),2008):

1. Tür Tavlamalar

- **Yayınma Tavlaması (Homojenleştirme Tavlaması):** Birincil kristalleşme sonucunda oluşan kimyasal bileşim farklılıklarını (mikrosegregasyonu) gidermek için yaklaşık 1100 °C ile 1300 °C arasında 6-12 saat süre ile uygulanır. Böylelikle ikincil kristalleşmede (örneğin sıcak şekillendirmede) ortaya çıkacak bantlı içyapının oluşması önlenir (Akdoğan (2),2008).

- **Tane İrileştirme Tavlaması:** Düşük karbonlu çeliklerde işleme kabiliyetini iyileştirmek amacıyla uygulanır. Tav sıcaklığı 950 °C ile 1100°C arasında seçilir. Tav süresi ise yaklaşık 3 - 5 saattir. Tanelerin irileşmesi ile kısa, kırılğan talaş elde edilir. Tavlama sonucunda gevrek ve kaba taneli bir içyapı elde edilerek, talaşın sürekliliği engellenir (Akdoğan (2),2008).
- **Gerilme Giderme Tavlaması:** Plastik deformasyonun, tekdüze olmayan soğutmanın ve faz dönüşümlerinin neden olduğu gerilmeleri gidermektir (Khaira,2013).
- **Yeniden Kristalleştirme Tavlaması:** Soğuk şekil verme sonucunda pekleşen malzemenin özelliklerini (örneğin yüksek dayanım, düşük süneklik ve tokluk gibi) başlangıç durumuna getirmek amacıyla yapılır. Tavlama sıcaklığı 600 °C - 700°C'dir. Yaklaşık 1 saatlik sürede gerçekleşir. Bu tavlamanın uygulanabilmesi için malzemenin en az % 10 oranında soğuk şekillendirilmiş olması gerekir. Böylece metalik malzemenin, katı halde bozulmadan yeniden kristalleşmesi sağlanır ve soğuk şekillendirme sonucunda oluşan pekleşme, sertleşme giderilerek malzemeye daha sonraki soğuk şekillendirme işlemleri için gerekli olan süneklik kazandırılır (Akdoğan (2),2008).

2. Tür Tavlamalar

- **Normalleştirme Tavlaması (Normalizasyon Tavlaması):** Tane yapısı bozulmuş iri taneli çeliği küçük taneli yapmak için çeliğin östenit bölgesine kadar ısıtıldıktan sonra hava ile soğutma işlemi yapmaktır (Khaira,2013).
- **Yumuşatma Tavlaması (Küreleştirme Tavlaması):** İyi işlenebilirlik için çeliği ötektoit sıcaklığın hemen altında 15 - 25 saat bekleterek iyice yumuşatmaktır. Şekillendirilebilme özelliğini iyileştirmek amacıyla kaba perlit yapısı elde etmek için çeliği östenit bölgesine ısıtıp soğutmaktır (Khaira,2013).

Şekil 22'de yukarıdaki tavlama çeşitlerinin uygulama bölgeleri gösterilmiştir. Tavlama metal ya da alaşım rekristalizasyon derecesine kadar ısıtılır. Burada moleküllerin ya da atomların homojen olmasına kadar beklenir. Yavaş yavaş soğutularak yapı içindeki dengenin oluşması beklenir.

Ekil 22. elikler İin Yeniden Kristalleřtirme ve Gerilim Giderme İle İlgili Kritik Blgeleri

Kaynak: Akdoğan(2), 2008

Isıtma işlemin herhangi bir anında, ısıtılan parçanın merkezi ve yüzeyi arasındaki sıcaklık değerleri farklıdır. Çelik yüzeyi belirli bir sıcaklığa ulaştığında merkezi, çeliğin kalınlığına ve ısı iletme kabiliyetine bağlı olarak daha geç ısınacağından sıcaklığı daha düşük olur. Parçanın yüzeyi ile merkezi arasındaki sıcaklık farkı, ısıl gerilmeler oluşturur ve bu ısıl gerilmeler sonucu parça çatlayabilir. Bu sıcaklık farkının, dolayısıyla ısıl gerilmelerin önlenmesi için, parçaya ısıtma sıcaklığının altındaki bir ya da iki sıcaklıkta ara ısıtma uygulanabilir. Ön ısıtma adı verilen bu işlemle, parçalarda daha düzenli ve homojen ısıtma sağlanır. Aynı zamanda belirli sıcaklıkta tutma süresi de kısalmır. Ayrıca ısıtma süresi arttırılarak da çeliğin yüzeyi ile merkezi arasındaki sıcaklık farkı azaltılabilir. Bunu sağlamak için çeliğe Şekil 28'deki gibi bir ısıtma döngüsü uygulanır.

Isıl işlem çeşitlerinden olan tavlamanın amaçlarını aşağıdaki gibi sıralayabiliriz (Kha-
ira,2013):

- Esneme yeteneğini arttırma
- Sertliği azaltma
- Şekillendirilebilirliğini arttırma
- Haddelenmiş metallerin yapısını tekrar oluşturma (yeniden kristalize etme)
- İç gerginliği azaltma
- Kırılganlığı azaltma
- İşlenebilirliğini arttırma
- Elektriksel direnci azaltma
- Manyetik özelliklerini arttırma

BÖLÜM 2: SÜREKLİ TAVLAMA HATLARI TANITIMI

2.1 Soğuk Haddeleme

Üretilecek malzemenin, eksenleri etrafında dönen silindirler (merdaneler) arasından geçirerek uygulanan basma kuvvetleri etkisiyle plastik şekil verme işlemine haddeleme denir. Erişir (2013), haddelemeyi “Kendi eksenini etrafında dönen takımlar (merdane) arasından geçirerek sürekli ya da çok adımda basma ile şekillendirmedir.” olarak tanımlamıştır. Merdaneler yüksek kaliteli çeliklerden yapılırlar. Haddeleme, metalin sahip olduğu sıcaklığın yeniden kristalleşme derecesine bağlı olarak sıcak haddeleme veya soğuk haddeleme olarak adlandırılmaktadır. Metal sıcaklığı, yeniden kristalleşme sıcaklığının üstünde ise sıcak haddeleme işlemi yapılır. Metal sıcaklığı, yeniden kristalleşme sıcaklığından düşükse, soğuk haddeleme işlemi olarak adlandırılır (MMP,2013). Soğuk haddeleme süreci Şekil 2’de gösterilmiştir.

Sıcak haddelemede belirli kalınlığa getirilen malzeme önce asitleme hattında yüzeyi temizlenir, malzemenin kalınlığı yeterli düzeyde değil ve daha inceltmek isteniyor ise bu aşamadan sonra Tandem Hadde adı verilen kısma gelir. Bu aşamada merdaneler ve tamburlar sayesinde malzemeye baskı ve gergi uygulanarak kalınlığı 0,4 ile 2 mm ye kadar inceltilebilir. Maksimum giriş kalınlığı ise 1,8 ile 4,5 arasındadır. Bu sınırlar haddeden haddeye değişiklik gösterir. İstenen kalınlığa getirilen malzemede meydana gelen atomik düzeydeki çarpıklıkları gidermek amacıyla kaliteye bağlı olarak yaklaşık 690 ile 850 °C’ler arasında 40 saniyeden fazla süre ile tavlama işlemi gerçekleştirilir. Bu işlem sürekli tavlama hatları için geçerlidir. Yığın tavlama hatları için bu değerler değişir. Malzeme soğuduktan sonra daha parlak yüzey elde etmek amacıyla %2 gibi kimyasal maddeler içeren solüsyonlar (deterjan olarak tabir edebileceğimiz) ile fırçalanarak temizlenir. Malzemeye uygulanan işlemlerden sonra yüzeyi ve kenarları bombeli bir hâl almıştır. Bunu düzeltme adına da temperleme işlemi yapılır ve istenirse dilme hattında kesilerek müşteriye teslim edilir.

Temperleme işleminden önce ve tavlama işleminden sonra yine müşterinin isteği doğrultusunda malzemenin kullanım amacına göre yüzeyi kalay, krom ya da galvanizle kaplanır ve yine istenirse ardından dilme hattında makaslar vasıtasıyla istenilen ölçü/ebatlarda kesilerek müşteriye sunulur.

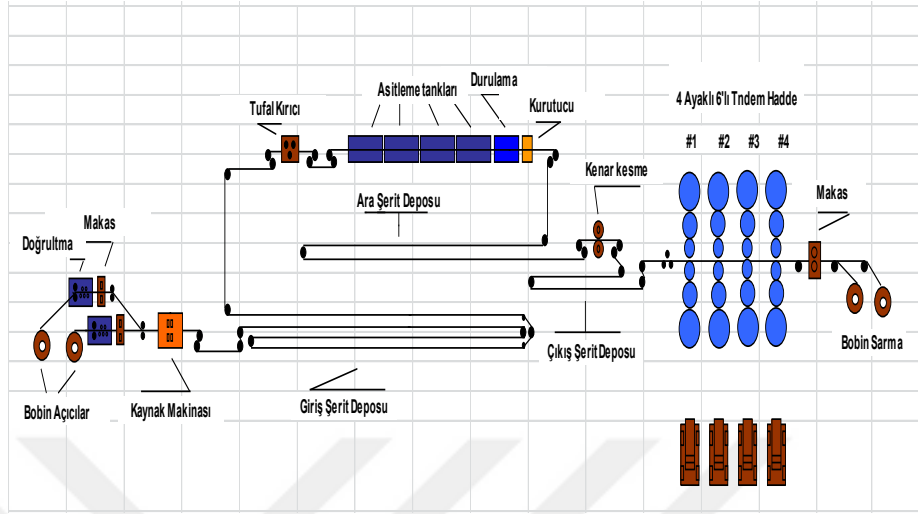
Sıcak haddelemeden sonra uygulanan soğuk haddeleme işleminde levha, çubuk veya şerit biçimindeki malzeme merdanelere ısıtılmadan verilir ve istenilen biçim elde edinceye kadar haddeden geçirilir (Şekil 20). Sıcak haddelemeden sonra uygulanan soğuk haddeleme işlemi, malzemenin mekanik işlenebilir özelliklerini geliştirir, yüzeyini parlattır (Sert,2105). Özellikle çeliğin haddelenmesi, sadece biçim vermek değil, aynı zamanda çeliğin mekanik özelliklerinin geliştirilmesini de sağlar (Erişir,2013).

Soğuk haddeleme, sağladığı yararların yanı sıra çizgisel kusur (Dislokasyon) yoğunluğunun artması ve sünekliğinin azalması gibi istenmeyen durumlara da yol açmaktadır. Çizgisel kusur, malzeme biliminde, kristal yapıların atomsal dizilişlerinde bir çizgi boyunca görülen kusurlardır. Denge konumundan ayrılan atomlar sonucunda çizgi çevresinde artık gerilimler meydana gelir ve şekil değiştirme enerjisi depo edilir. Çizgisel kusurlar genellikle malzemelerin katılma sürecinde oluşmakla birlikte, plastik şekil değiştirme sırasında da sayıları artar. Soğuk haddeleme sırasında oluşan çizgisel kusur yığılmaları genellikle tavlama ile giderilir (Erişir,2013).

Günümüzde kullanılan hadde tezgâhlarında üst üste veya art arda yerleştirilmiş birden fazla merdane çifti bulunmaktadır. Böylece malzeme merdanelere kesintisiz olarak verilerek belli bir biçimi alıncaya kadar defalarca haddelenmesi sağlanır. Haddeden çekme işi, çamaşırları eski çamaşır makinesinin sıkıcısının silindirleri arasından geçirmekten farklı değildir. Çamaşırdaki düşünülen şey suyu sıkıp çıkarmaktır, madenlerde ise sıkıştırıp daha ince bir hale getirmektir (Sert,2105). Şekil 23’de tipik bir soğuk haddeleme hattı görülmektedir.

Malzemenin özelliği, belirli bir etkiye karşı verdiği cevaptır; mekanik özelliği ise gevreklik, süneklik, tokluk, yorulma, sürünme vs. özellikleridir. Süneklik, bir cismin çekme gerilmesi altında çatlamaya veya kırılmaya uğramadan geri dönüşümsüz şekil değiştirme yeteneğidir. Tokluk, bir malzemenin plastik deformasyon esnasında kırılana kadar olan süre içerisinde yuttuğu enerji miktarıdır. Yorulma; malzeme biliminde bir malzemenin devirli olarak sürekli yüklemeye uğraması sonucu ilerlemeli ve yerel yapısal hasara uğramasıdır. Sürünme, malzemelerde ergime sıcaklığının altında sabit bir yüksek sıcaklıkta ve sabit bir yük altında meydana gelen kalıcı şekil değişimine sürünme denir. Fiziksel özellikleri, iletkenlik, ısı, özgül ağırlık, optik özellikleri (şeffaflık) gibi özellikleridir. Kimyasal özellikleri, bileşimi, ortamdan etkilenmesi-korozyon, oksidasyon

gibi özellikleridir. Malzemenin performansı, herhangi bir ortamda istenilen fonksiyonları yerine getirebilmesidir.



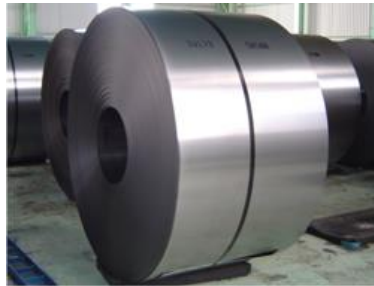
Şekil 23. Soğuk Haddeme Hattı

Kaynak: Erdemir (CPT), 2017

2.2 Sürekli Tavlama Hattı Tanıtımı

2.2.1 Sürekli Tavlama Hattı Genel

Bu şekilde göre sıcak haddehaneden gelen ve kalınlıkları 1,8 mm ile 4,5 mm arasında olan bobinler sol taraftaki bobin açıcılar tarafından hatta verilir. Doğrultmadan geçen bobinleri ucundaki bozuk kısımlar kesilerek kaynak makinası tarafından öndeki bobine kaynak yapılır. Hattan çıkan bobin görüntüsü Şekil 24'da verilmiştir.



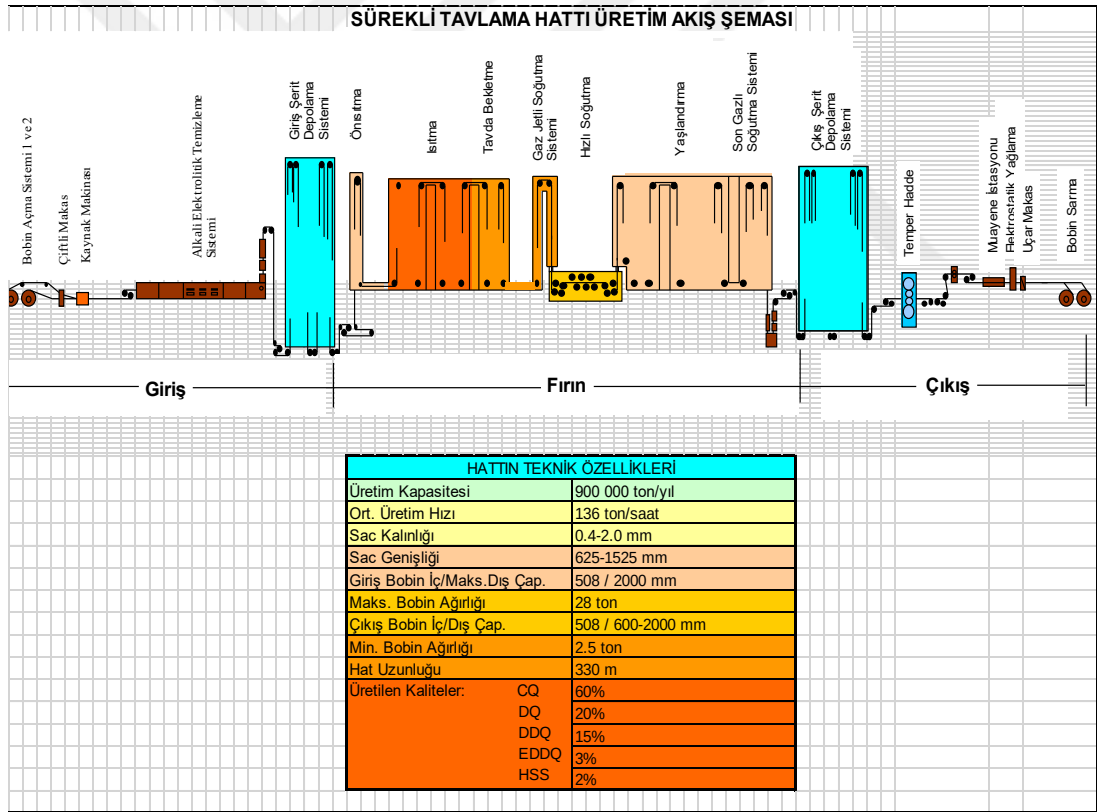
Şekil 24. Tipik Bobin Görüntüsü

Kaynak: Erdemir (CPT), 2017

Bu surette hattın sürekli olarak çalışması sağlanır. Kaynak yapılma sırasındaki zaman kaybını önlemek ve hattın çalışmasını sağlamak amacıyla giriş şerit deposu kullanılır. Şerit yüzeyindeki tufalları kırmak için tufal kırıcı ve şerit yüzeyini temizlemek için asitleme tankları kullanılır. Sıcak metal işlemede metal yüzeyinde meydana gelen oksit tabakasına tufal denir. Aside maruz kalan şeridi temizlemek için durulama ve kurulama

işlemleri uygulanır. Sıcak haddelenen gelen şeridin kenar kusurlarının giderildiği ve talebe göre şerit genişliğinin kesilerek azaltıldığı kenar kesme kısmından geçen şerit daha sonra siparişine göre şeridin inceltileceği tandem hadde kısmına gider. Kenar kesme kısmında bıçak değiştirme ve kaynaklama sırasında oluşacak duruşları önlemek için ara ve çıkış şerit deposu kullanılır. Haddelenen malzeme ara stok sahasına konur. Buradan da vinçler vasıtasıyla Şekil 25’deki tavlama hattına gider.

Soğuk haddeleme işlemindeki en son işlem olan tavlama işlemi, soğuk haddelenmiş ürünlerin kalitesini arttırmak için önemli bir rol oynar. Nippon Steel Kimitsu İşletmesi’nin 1972’de ilk sürekli tavlama hattını kurmasından sonra, yüksek üretim kapasitesi, düşük maliyet, yüksek kalite ve çeşitlilik avantajları nedeniyle dünyada kullanımı yaygınlaşmıştır. Yığın tavlama işleminde tavlama için 25 saat beklenmesine karşılık sürekli tavlama işleminde bu süre 10-15 dakika civarındadır.



Şekil 25. Sürekli Tavlama Hattı Üretim Akış Şeması

Kaynak: Erdemir (STH), 2017

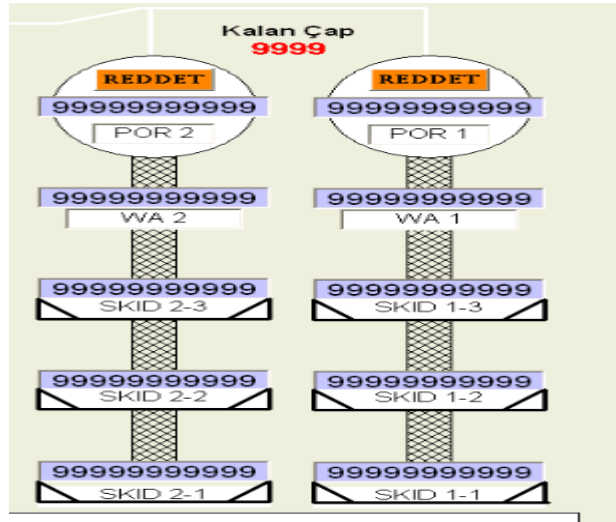
Çelik, yüksek basınçta ince bir şerit haline haddelendiği zaman sertleşir ve en son ürünler olan otomobil parçaları, bilgisayar kasaları, yapı elemanları ve benzerleri gibi ürün haline getirilmeye çalışıldığı zaman başarılı olunamaz. Sürekli tavlama hattı fırın bölü-

münde; sac bünyesinde soğuk haddeleme sırasında oluşan içyapı bozukluğu ve iç gerilme birikiminin yeniden kristalleşme sıcaklığına kadar tavlanıp giderilmesi amaçlanmaktadır. Saca soğuk haddeleme sırasında bozulan istenilen fiziksel özellikleri kazandırmak için çeşitli ısıtma ve soğutma işlemleri uygulanır. Bu işlemler, sacın işleme performansını artırır.

2.2.2 Sürekli Tavlama Hattı Fırın Bölümü

Tavlama hattı fırın bölümündeki şeridin boyu yaklaşık olarak 330 metredir; bu hatta şerit maksimum 320 metre/dakika hızla hattan geçer. Bu hız çeliğin tavlama özelliklerine ve boyutlarına göre değişmektedir. Hattın özellikleri, sıcaklık sınır aralıkları ve maksimum hız sınırlamaları Ekler bölümündeki Ek-1, Ek-2 ve Ek-3'deki tablolarda verilmiştir. Fırın ısıtma kapasitesi kuruluş esnasında belli olduğundan dolayı kalın ve geniş malzemelerin maksimum hızda çekilmesi mümkün olamamaktadır.

Fırın giriş tarafında (Bobin açma sistemlerinin bulunduğu taraf 10 bobin alabilecek bir konveyör sistemi vardır (Şekil 26). Üretim sırasına alınan ve konveyöre konulan bobinler sıra ile açma sistemlerinde uçları kesildikten sonra kaynak yapılarak hatta verilirler. Şerit yüzeyindeki yağ ve demir parçacıklarını temizlemek için içeriğinde NaOH(sodyum hidroksit) ve aşındırıcı bulunan bir solüsyon kullanılır.



Şekil 26. STH Giriş Bölümü Konveyör Görünümü

Kaynak: Erdemir(STH), 2017

Fırın bölümünde ön ısıtma, ısıtma, bekleme, gaz jet soğutma, merdaneli soğutma, yaşlandırma, gaz jet son soğutma ve son sulu soğutma bölümleri bulunur. Fırın bölümünde koruyucu atmosfer gazı olarak HNX gazı kullanılmaktadır.

Hidrojen tesisinde ‘‘Catalytically Reforming’’ metoduyla doğal gazdan üretilen hidrojen azot gazı ile karıştırılarak (yaklaşık %7 H_2 ve % 93 N_2) fırın içerisine verilir. Fırın bölgesinde oksijen miktarı 10 ppm’in altında, çiğ noktası da 10 °C’nin altında tutulur. Fırın içi basıncı 8-12 mmH_2O (mm su sütunu)’dur.

Ön ısıtma bölümünde şeridin sıcaklığının 150 – 200 °C ‘ler arasına çıkarılmasına çalışılır (Ek - 2). RTH 1, RTH 2 ve RTS bölümlerinde bulunan radyan tüplerde doğalgazın yakılması sonucunda ortaya çıkan egzoz gazları 2 adet egzoz fanı tarafından emilerek ön ısıtma bölgesinde şeridin üzerlerine direkt olarak üflenir.

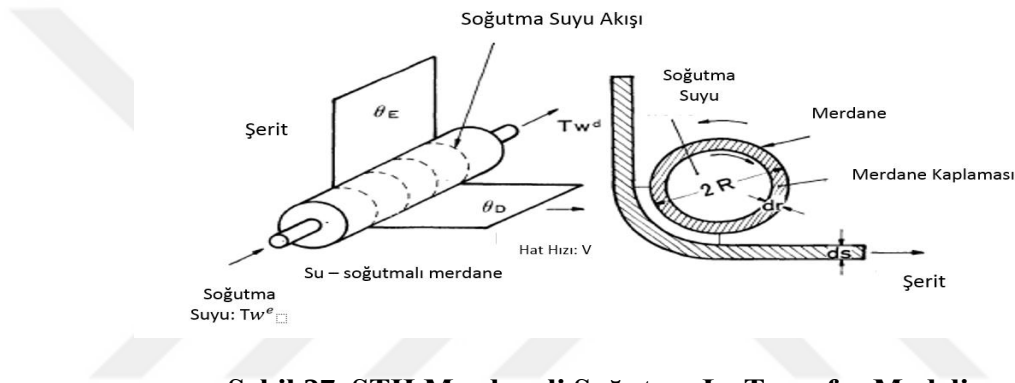
Ön ısıtma bölgesinde şerit yüzeyinde oksitlenmeyi önlemek için şerit sıcaklığı maksimum 165 °C ‘de tutulur. Ön ısıtma yapılmasının nedeni şeridi yavaş yavaş ısıtma bölgesine yaklaştırarak ısı gerilmelerinin önlenmesini sağlamaktır. Baca gazları bu bölgeye gönderilmeden önce soğuk hava ile karıştırılarak sıcaklık kontrolü sağlanır. Soğuk havayla karıştırılmış egzoz gazı ön ısıtma bölgesinde şerit üzerine jet üfleyiciler vasıtasıyla üflenir. Ön ısıtma işlemini tamamlayan egzoz gazları bölgenin üst kısmından bir fan vasıtasıyla alınarak bacadan atmosfere bırakılır.

Ön ısıtma bölgesini terk eden şerit, alt kısımdaki ara geçiş tüneline geçtikten sonra gaz ateşlemeli radyan tüplerin kullanıldığı RTH bölgesine gelir. Bu kısımda şerit maksimum sıcaklık derecesine ısıtılır. Radyan Tüplü Isıtma Bölgesi; RTH1 ve RTH2 olmak üzere 2 bölümden oluşmaktadır. RTH1 kısmında şerit 13, diğer kısımda 12 geçiş yapmaktadır. Her radyan tüp brülöründe 1 adet ana brülör, 1 adet pilot brülör bulunmaktadır. Toplam radyan tüp miktarı RTH1’de 256 adet, RTH2’de 119 adettir. Brülörlerde yakıt olarak doğalgaz kullanılmaktadır. Yanma havası brülörde bulunan reküperatör ile yaklaşık 400 °C‘ ye ulaşır. Brülörlerin kapasitesi 85000-120000 Kcal/h ‘dır.

Gaz jet soğutma (GJC) bölgesi her birinin içerisinden tek şerit geçişi olan 2 bölümden oluşmuştur. Şerit fırın bekleme bölgesinden merdaneli soğutma (RQ) bölgesine gitmeden önce burada ara soğutma işlemine tabi tutulur. Şeritin soğutulması işlemi için bölge 5 ayrı soğutma bölgesine ayrılmış ve her bölge değişken hız özelliğine sahip soğutma fanları ile teçhiz edilmiştir.

Soğutma işlemi sırasıyla fırın atmosfer gazı (HNX) fan vasıtasıyla çekilerek eşanjörden geçirilmek suretiyle soğutulur. Isısı düşürülen gaz tekrar fırın bölgesine şeritin her iki tarafına yerleştirilmiş tüpler vasıtasıyla üflenir. Her fan devresinde 2 adet su soğutmalı eşanjör bulunur. Soğutma fanlarının kapasitesi zon 1 – 4 için $4 \times 92000 \text{ m}^3/\text{h}$ ($4 \times 1963000 \text{ Kcal/h}$), zon 5 için $1 \times 7000 \text{ m}^3/\text{h}$ ($1 \times 1868000 \text{ Kcal/h}$)’dır. HNX maksimum çıkış sıcaklığı 47°C ‘dir.

Şerit GJC Bölgesinden çıktıktan sonra RQ Bölgesine girer. Bu bölgede şerit içerisinde su dolaşan 6 adet merdane arasından geçer. Merdanelerin baskısı ayarlanmak suretiyle şerit istenilen sıcaklığa düşürülür (Şekil 27).



Şekil 27. STH Merdaneli Soğutma Isı Transfer Modeli

Kaynak: Erdemir (STH), 2017

Merdaneli Soğutma Bölgesinden geçen şerit yaklaşık 400°C ’de Yaşlandırma Bölümünden (OA) 2 – 3 dakikada geçirilir. Bu bölümde bobinin üretiminden sanayide kullanıncaya kadar geçecek süre içerisinde, sac bünyesinde oluşabilecek akma dayanımı artışı (yaşlanarak gevrekleşme) önlenmektedir. Yaşlandırma bölgesi 2 kısımdan oluşur. 1. kısım 17 şerit geçişi, 2. kısım ise 15 şerit geçişi içerir.

Yaşlandırma Bölgesinde ısı kayıplarını karşılayacak Elektrikli Radyant Tüp ısıtıcılar bulunmaktadır. OA1 ve OA2 bölümlerinde toplam 120 adet (2160 KW) elektrikli ısıtıcı vardır. OA1 bölgesinde soğuma doğal olarak ortamın soğumasından kaynaklanır. OA2 bölgesinde soğutma işlemi 2 adet fan soğutucu ile yapılır. Soğutma fanı kapasitesi $2 \times 65000 \text{ m}^3/\text{h}$ ve $2 \times 72000 \text{ Kcal/h}$ ‘dır.

Şerit OA bölümünden sonra fırının son bölümü olan Son Jet Soğutma bölümüne (FGJ) girer. Burada amaç şerit fırın atmosferinden çıkarken sıcaklığını oksitlenmeyecek bir seviyeye yani 160°C ’nin altına indirmektir.

Soğutma müstakil jet soğutucular vasıtasıyla sağlanmaktadır. Her soğutucuda; bir su soğutmalı eşanjör, bir adet tek hızlı fan (15 000 m³/h), şeridin her iki tarafına gelecek şekilde toplam 6 adet nozul kolektörü bulunmaktadır.

2.3 Sürekli Tavlama Hattı Modeli

2.3.1 Sürekli Tavlama Hattı Modeli Genel

Tez çalışması yapılan işletmede ısıtma ve soğutma değerlerinin değişimine neden olan 5 çeşit kalite kullanılmaktadır. Bu kaliteler çeşitli mekaniksel özelliklerine bağlı olarak CQ, DQ1, DQ2, DDQ ve EDDQ adlarını alırlar. Bunların karbon, mangan ve kükürt miktarları sırasıyla yüksek değerden düşük değere sıralanırlar. Ayrıca bu değerlere bağlı olarak malzeme akma ve çekme mukavemetleri değişir. Bu değerlere göre de değişik işlerde kullanılırlar.

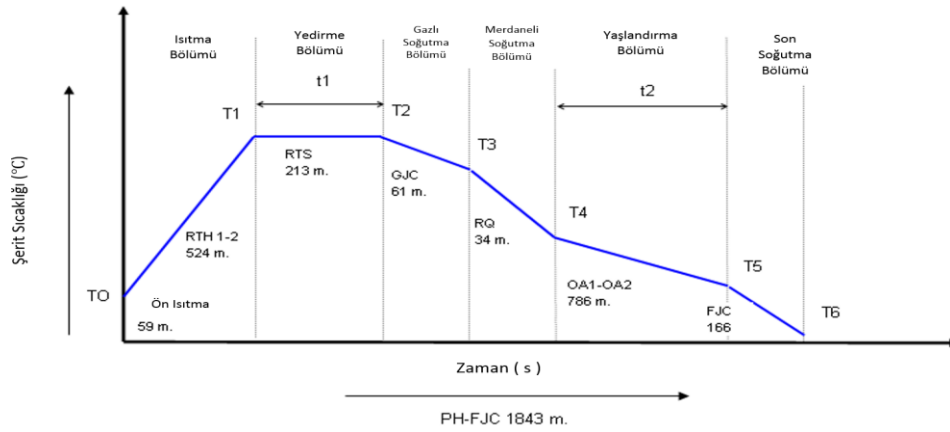
CQ kalite ticari kalite olup soğuk haddeleme ürünleri için genel bir uygulamadır. Sırasıyla basit uygulamalardan daha karmaşık şekilli ürünlerin üretilmesinde kullanılırlar. EDDQ kalite eşya ve otomotiv sanayiinde çok fazla derin çekme ve son derece karmaşık şekilli parçaların imalatında kullanılır. Bunların tavlama sıcaklıkları ve tavlama kalma süreleri farklı farklıdır. Yine sırasıyla tavlama sıcaklıkları CQ kalite için olan 720 ± 30 dereceden başlayıp en son EDDQ kalite için 830 ± 20 dereceye kadar çıkar ve hıza bağlı olarak tavlama fırınında şeridin belirli sürenin üstünde kalması gerekmektedir. Kalitelere göre sıcaklık ve maruz kalma süre dağılımları EK-2’de gösterilmiştir.

Fırının içinde şeride olan ısı transferi fırın sıcaklığına, şeridin ebatlarına ve hat hızına doğrudan bağlıdır. Bu nedenlerle malzeme kalite performansını etkileyen faktörler arasında, şerit sıcaklığı, kalınlığı, genişliği, fırın sıcaklığı ve hat hızı anahtar durumdadır.

Hız, kalınlık, genişlik ve kalite işin içine girince STH fırınının sıcaklık kontrol sonucunu geleneksel kontrol yöntemini kullanarak elde etmek zordur. Tavlama işlemi kontrolü zor ve karmaşık bir endüstriyel işlem olup, çok iyi bir performans ile iyi bir ısı kontrol yöntemi büyük önem taşımaktadır (Guo, Zhang, You, Chen ve Zhang (2009)). Her fırın çıkışında fırını ne kadar doğru kontrol ederseniz, ürün kalitesi o kadar etkilenir (Ming, Datai, Jiangang (2008)). Şerit sıcaklık kontrolü odaklı araştırmalar, geçmiş yıllardan günümüze her zaman sürekli tavlama hatları için önemli bir konu olmuştur.

Dinamik şerit sıcaklığı kontrol sisteminin ana amacı, çeşitli kısıtlamalar altında (örneğin fırın kapasitesi) şerit boyut değişimlerine ve maksimum hat hızında çalışmaya olanak

sağlamak üzere hat hızı, fırın sıcaklığı gibi parametreleri kontrol ederek bir uygun ısı şablonunda belirtilen her fırın bölümünün çıkışında şerit sıcaklığını tutmaktır. Dinamik şerit sıcaklık kontrol sistemi set noktası değerlerinin hesaplanması için veya çıkış şerit sıcaklık tahmini için ısı transferi teorisine dayanan matematiksel modeller kullanır. Bu sistemde kullanılmak üzere ısı transferine dayanan matematiksel modeller olarak Ek-4’de listelenmiştir. Hattın ısıl modeli Şekil 28’ de verilmektedir.

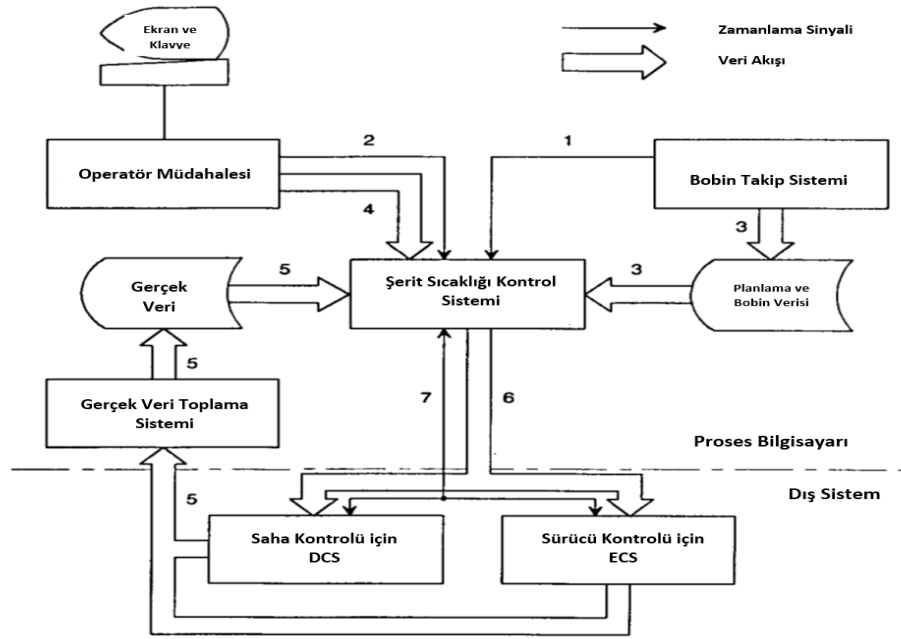


Şekil 28. Şerit Tavlama Döngüsü

Kaynak: Erdemir (STH), 2017

2.3.2 Sürekli Tavlama Hattı Sıcaklık Kontrolü

Mevcut sistemin otomasyon sistemi çalışma prensipleri Şekil 29’deki gibidir. Sistem şerit sıcaklığı üzerine kurulmuştur. Operatörden, gerçek ölçüm değerlerinden ve hat takibinden gelen bilgiler ışığında şerit sıcaklık değerini tutturmaya çalışır.



Şekil 29. Şerit Sıcaklık Kontrolünün Diğer Sistemlerle olan İlişkisi

Kaynak: NKK Corporation, 1995

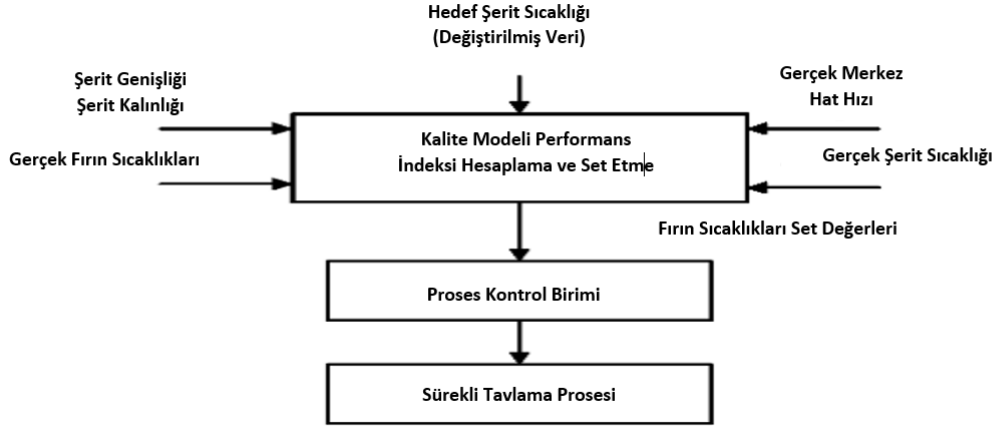
Şerit sıcaklık kontrol sistemi aşağıdaki kontrol mesajlarını alır:

1. Bobin izleme sinyali
2. Operatör veri girişi zamanlaması
3. İzlenen veya programlanmış bobin verisi
4. Operatör veri girişi
5. Proses gerçek verisi
6. Preset verisi
7. Proses kontrol sistemi devrede / devrede değil

Bu mesajlara göre hat otomasyon sistemine yeni verileri göndererek hattın yeni değerlerle çalışmasını sağlar.

Bu kalite modeline göre eğer şerit özellikleri veya boyutları değişti ise model programından alt seviyede kontrol yapan otomasyon sistemlerine yeni set değeri gönderilmektedir. Otomasyon sistemleri ise aldığı bu değere sistemi ayarlamaya çalışmaktadır. Model programı ise sürekli olarak şerit ve fırın sıcaklıklarını kontrol ederek fırın durumunu

kontrol altında tutmaya devam etmektedir. Bu hiyerarşik yapı genelde tavlama hatlarında fırın optimizasyonu ve kontrolü için kullanılmaktadır. Fırın kontrol sisteminin girdi ve çıktı değerleri Şekil-30'de verilmiştir.



Şekil 30. Tavlama Fırının Sıcaklık Kontrolü Yapısı

Kaynak: NKK Corporation

Yukarıda da belirtildiği üzere fırın bölümü 7 alt bölgeden oluşmaktadır ve her bir alt bölgenin kendine özgü çalışma sıcaklık değerleri vardır ve şeritin bu sıcaklık değerinde belirli süre kalması gerekmektedir. Bu çalışma sıcaklık değerleri şeridin özelliklerine (kalite) göre değişmektedir. Eğer peşi sıra giden bobinlerinin özellikleri aynı ise hat hızını da sabit tutmak mümkündür. Bu durumda fırının sıcaklığını belirli bir dereceye set ederek şerit sıcaklığını belirli dereceler arasında kalmasını sağlamak olasıdır. Radyan tüplü ısıtma ve radyan tüplü soğurma modelinin denklemleri Ek-4 'de verilmiştir.

Her matematiksel model, bir diferansiyel denklem ve model parametreleri güncelleme (U-değeri) ile ifade edilir. Bu model güncelleme parametre modeli, her zaman gerçek saha koşullarına uygunluk sağlaması için sabit dönemlerde güncellenir. Parametreleri güncellenmiş modeli parametrelerinin bu güncelleme işlemine "Uzun süreli hata düzeltme fonksiyonu" denir. Yukarıda belirtilen modellerin hatasının düzeltilmesi amacıyla parametre güncellemesi yapılmasına rağmen bu güncelleme hatayı hemen düzeltecek seviyede olmaz. Zamanla yavaş yavaş olacak bir güncelleme söz konusu olur.

Buna göre, hesaplama istendiğinde model gerçek verileri (giriş şerit sıcaklığı, fırın sıcaklığı, hat hızı vb) hesaplanmış değer ve çıkış şerit sıcaklığı arasındaki farka karar vermek için kullanır ve bu değerleri yeni hedef değerlerin hesaplanması için kullanır. Bunu şöyle açıklayabiliriz.

1. Gerçek veriler (giriş şerit sıcaklığı, fırın sıcaklığı, hat hızı, şerit kalınlığı vb.) kullanarak modeli uygula ve T_{SC} sıcaklığını bul.
2. T_{SC} ile T_{SA} 'yı karşılaştır.
3. T_E karar ver.

$$T_E = T_{SC} - T_{SA}$$

4. Bu hata hedef T_{SS} yansıtılır.

$$T_{ST} = T_{SS} + T_E$$

5. T_{ST} değerinin elde edilmesi için yeni set değerleri (fırın sıcaklığı vb) hesaplanır.

T_{SC} = çıkış şerit sıcaklığı

T_{SA} = gerçek çıkış şerit sıcaklığı

T_E = model hatası

T_{SS} = çıkış hedef şerit sıcaklığı

T_{ST} = çıkış hedef fırın sıcaklığı

Bu fonksiyon “Kısa süreli hata düzeltme fonksiyonu” olarak adlandırılır.

Dinamik şerit sıcaklık kontrol modeli 4 alt bölüme ayrılır.

- Normal tavlama şartları altında kontrol
- Şerit boyutlarının değişmesi şartında uygulanan kontrol
- Isıl modelin değişmesi şartında uygulanan kontrol
- Hat hızının değişmesi şartında uygulanan kontrol (NKK,1995)

Yukarıdaki şartların oluşması durumunda matematiksel model her bir şart için hazırlanmış programı çağırarak sıcaklığı kontrol etmeye çalışır.

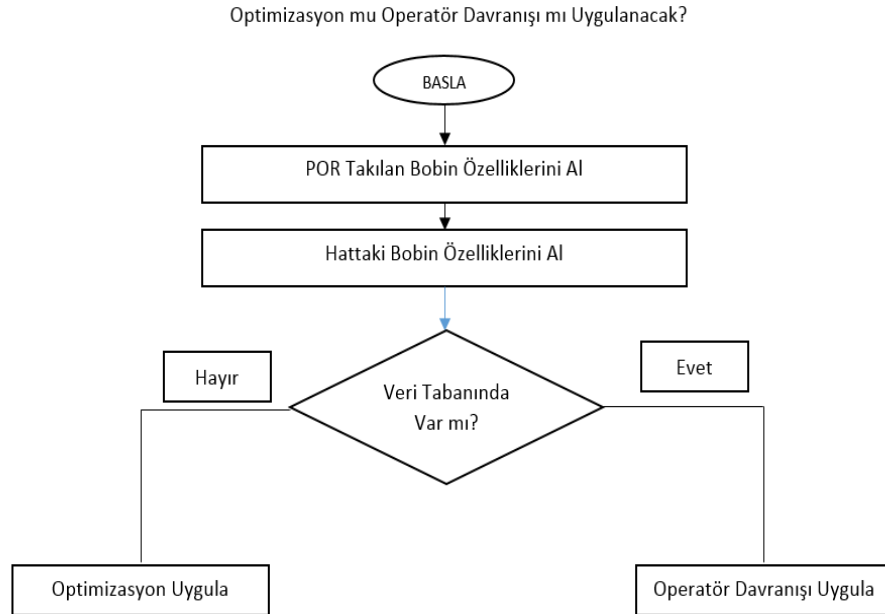
BÖLÜM 3: HEDEF PROGRAMLAMA VE YAPAY SİNİR AĞI UYGULAMALARI

3.1 Problem Tanımı

Tez konusu olan STH hattında birbirini takip eden iki problem bulunmaktadır. Bunlardan birincisi hatta girecek olan bobinlerin hangi sıcaklık set değeri ile tavlama yapılması gerektiğini belirlemektir. Bunun amacı en uygun tavlama sıcaklıklarını kullanarak sıcaklık geçişleri arasındaki farkı en az seviyede tutmak ve bu sayede enerji tasarrufu sağlamaktır. Bu çalışmanın aynı zamanda uygulamalar arası standardı sağlamak ve olası gelen bobinlerin uygun sıraya sokma problemini gidermek amacı da vardır. Bu konu ile ilgili yazılan program Ek-5’de verilmiştir. Hat normal şartlarda üretilen bu set değerlerine göre çalışır.

Yalnız bu çalışma, araya üretim harici bir bobin almayı gerektiren yani fırın kapasitesinin ısıtma için yeterli gelemeyeceği durumlarda geçerli olmayacaktır. Bu amaçla ikinci bir çalışma yapılmıştır. Bu çalışmanın amacı, kalınlık, genişlik ve kalite geçişlerinde operatör davranışlarını belirleyerek tavlama sisteminin bu doğrultuda hareket etmesini sağlamaktır. Bu sayede kullanılamayan geleneksel tavlama modellerinin yerine bu yöntem geliştirilecektir.

Bu durumları normal çalışma sırasında ayırt edebilmek için Şekil-31’de verilen akış diyagramı uygulanacaktır.



Şekil 31. Program Seçim Akış Diyagramı

Birinci problemimiz bir karar problemidir. Gerçek hayatta önümüze çıkan karar problemleri genelde çok hedefli karar problemleridir. Bu amaçlar genelde birbiri ile çatışırlar. Amaçlardan birini yerine getirmek diğer amacımızı etkileyebilir. Hepsini karşılayacak sınırlı kaynaklarımız vardır. Fakat bunların yanı sıra bu amaçları genelde önem durumuna göre bir sıraya sokabiliriz. Önümüze çıkan bu amaçların yerine getirilmesi için ürettiğimiz kararlardan en uygununu seçme işi çok hedefli karar problemi olarak karşımıza çıkar. Bu karar probleminde de hem maksimum kapasitede çalışmak isteği yer almakta aynı zamanda da üretilen malzemenin belirli sıcaklıklar arasında tavlmasını sağlayarak kaliteden ödün vermeme isteği bulunmaktadır.

Karar vericinin en uygun karar vermesini sağlayacak - kişisel değer yargılarını da karar sürecine dâhil ederek- yardımcı olan bilimsel teoriler ve analitik yöntemler geliştirilmiştir. 1. bölümde de bahsedildiği gibi bu çalışmada kullanılan karar problemi matematiksel modelleme gerektiren ve belirli kurallara bağlı olduğu için kurallı modeller sınıfına giren bir modelledir. Uygulama bölümünde de görüleceği gibi birden fazla karar değişkeni olduğu için çok hedefli karar verme modellemesi uygulanmıştır. Daha sonra sunulacak olan matematiksel modelden de anlaşılacağı gibi model, doğrusal olmayan bir modelleme içermektedir.

Problemin ikinci kısmı ise kullanılamayan dinamik matematiksel modellerin yerine konulabilecek bir model geliştirmektir. Normal tavlama şartları altında kontrol, şerit boyutlarının değişmesi, şerit kalitesinin değişmesi ve hat hızının değişmesi ile ilgili matematiksel modellerde uygun geçişler olduğunda 1. problemde ele alınmış ve bu problem en iyi çözümü verecek optimizasyon problemi yazılmıştır. Yalnız hat ısıtma kapasitesinin uygun olmadığı geçişlerde ise kullanılan bu modelin sonuçlarını direkt uygulamak mümkün değildir. Çünkü geçişlerde istenilen şerit ve fırın sıcaklıklarını ayarlamak ve geçişleri fire vermeden yapmak mümkün olamamaktadır. Aralara “ara bobin” denilen bobinlerin konulması ve bu bobinlerin belirli kullanımlarından sonra tekrar işlenmek üzere çelikhane tarafına hurda olarak gönderilmesi gerekmektedir. Fakat işletme pratiklerinde uygulanan ise daha düşük kaliteli olan bobinlerin son veya baş bölümlerinde uygulanan sıcaklıkları dikkate almamak ve sonra veya önce gelen daha yüksek kaliteli ürünlerin kalitelerinin tavlama sıcaklıklarına erişebilmektir.

Ayrıca matematiksel model fonksiyonlarında kullanılan güncelleme katsayıların fırın koşullarına göre zaman zaman güncellenmesi gerekmektedir. Bu yapılmadığı sürece matematiksel modelin gerçek değerlere yakın değerler vermesi mümkün değildir. Bunun içinde işi iyi bilen ve ilgili mühendislerin olması gerekmektedir. Bu modellerin yerine tezimizde yapay sinir ağları önerilmektedir.

3.2 Metodoloji

3.2.1 Giriş Konveyörü Optimizasyon Uygulaması

Birinci problemdeki işlenmek üzere konveyörlere konmuş bobinlerin en uygun tavlama sıcaklıklarını bulmak işi de çok hedefli karar verme problemidir. En uygun çözümü bulmak için bu çalışmada Microsoft Solver Foundation yazılımının hedef programlama ile ilgili fonksiyonları kullanılmıştır.

Microsoft Visual Studio .Net C# kullanılarak optimizasyon programı yazılmış ve algoritma olarak HybridLocalSearchDirective (hybrid local search solver) kullanılmıştır. Bu algoritmaya göre hatta girecek 10 adet bobin için en uygun tavlama sıcaklıklarının ne olması gerektiği bulunmuştur. İlgili kaynak kodu Ek-5’da verilmiştir. Microsoft HybridLocalSearchDirective metodu **stokastik tepe tırmanma** algoritması kullanılmaktadır. Bu yöntem doğrusal olmayan programlamanın çözüm yöntemleri bölümünde ayrıntılı olarak anlatılmıştır.

Bobinlerin bu konveyörlerdeki konumları Şekil 26’de verilmiştir. Bobinler belirli bir sıra dâhilinde konveyöre konulmaktadır. Hatta giriş sırasına göre bobinler sıralanmakta ve bu sıraya göre fırınları set olması gereken sıcaklık değerleri belirlenmektedir. Bu sıcaklık değerlerinin hesaplanmasında kısıtlılıklar vardır. Aşağıdaki ısı transfer tanıtımı bölümünde şeridin üzerine verilen ısısal enerjinin yükseltilmesi sırasında ortaya çıkacak olan kısıtlamalar anlatılmıştır.

Isı Transferi Tanıtımı

STH hattı fırın bölümlerinde, malzeme ve ısıtma / soğutma donanımları arasındaki ısı transferi temel olarak ışıma ve/veya konveksiyon şeklinde gerçekleşir.

Bu mekanizmalarda gerçekleşen ısı transferi fonksiyonu:

Işıma fonksiyonu:

$$Q_r = A.B. E_s. H_r. (T_2 - T_1) \quad (1)$$

Konveksiyon fonksiyonu:

$$Q_c = A. H_c. (T_2 - T_1) \quad (2)$$

A = Isı transferi yüzeyi = W.L.2

W = Şerit genişliği

L = Isı transfer alanı net uzunluğu

E_s = Işıma yayım kabiliyeti katsayısı (emissivity) (Işıma yayılım Katsayısı)

B = Kara cisim katsayısı

H_r, H_c = Isı transferi katsayıları

2 çarpanı şeridin 2 yüzeyi için kullanılmıştır

Fırın Isıtma Kapasite

$$P = g.W. T.V.60 \quad (3)$$

W = Malzeme şerit genişliği

T = Malzeme şerit kalınlığı

G = Özgül ağırlık

V = Proses Hızı

P = Yük kapasitesi (1 saat için)

Bu kapasite için gerekli olan ısı miktarı;

$$Q_s = C_p. (T_2 - T_1).P \quad (4)$$

C_p = Özgül Isı

Isı Balansı

Isı eşitlik denklemlerinden;

Alınan ısı = Verilen ısı

$Q_s = Q_r + Q_c$ (Bu denklemde % 10'dan az olan fırın kayıpları ihmal edilmiştir.)

(1), (2), (3) ve (4) numaralı denklemleri kullanarak;

$$C_p.(T_2 - T_1).g.W.t.V.60=A.[H_r.B. E_s+H_c]$$

$$C_p.g.W.t.V.60 =W.L. [H_r.B. E_s+H_c]$$

$$g.W.t.V.60 = \frac{W.L.[H_r.B.E_s+H_c]}{C_p}$$

$$P= f (W.L)$$

Isıtma ve soğutma işlemlerinde ısı transfer mekanizmalarının değişmediği varsayılırsa; kapasitesinin şerit genişliği ve ısı transfer alanı net uzunluğunun (fırın uzunluğu) fonksiyonu olduğu görülmektedir.

Kapasite bu iki değişken ile orantılıdır.

$$C_p.g.t.V.60 = W.L.[H_r.B. E_s+H_c]$$

$$t.V = \frac{L.[H_r.B.E_s+H_c]}{C_p.g.60}$$

“Şerit kalınlığı x Hız” değeri ısıtma ve soğutma işlemleri ile fırın uzunluğuna bağlı olarak sabittir.

Kapasitenin tav çevrimine göre değişimi

$Q_r + Q_c$ toplamı fırın kuruluş standartlarına göre sabit kabul edilirse;

$$Q_s =C_p .(T_2 - T_1).P= Q_r + Q_c = \text{sabit}$$

Tavlama Sınırlılıklarını ise aşağıdaki Tablo 11’deki gibi gösterebiliriz.

Tablo 11
Kaliteye Göre Şerit Tavlama Sıcaklıkları

Ürün Kalitesi	Şerit Sıcaklığı (°C)	Zaman (Sn.)
CQ	720 ± 30	≥ 40
DQ-1	770 ± 30	≥ 40
DQ-2	780 ± 30	≥ 40
DDQ	800 ± 30	≥ 40
EDDQ	830 ± 30	≥ 40

Hat hızı maksimum 320 m/d. olacaktır.

Radyan Tüp Isıtma Bölümü Isıtma Kapasiteleri ise aşağıdaki Tablo-12'deki gibidir.

Tablo 12
Radyan Tüp Isıtma Bölümü Isıtma Kapasiteleri

Bölüm	Bölge	Tüp Adedi	Teorik Gerçek Kapasite (Kcal/H)	
			Tüp	Zon
RTH1	1	59	85 000	4 900 000
RTH1	2	58	85 000	4 975 000
RHT1	3	69	85 000	5 649 000
RHT1	4	70	85 000	5 621 000
RTH2	5	29	120 000	3 460 000
RTH2	6	30	120 000	3 626 000
RTH2	7	30	120 000	3 260 000
RTH2	8	30	120 000	3 260 000
TOPLAM	8	375	-	34430 000

Matematiksel Modelleme

Amacımız yukarıdaki verileri kullanarak hattın ısı kapasitesi sabit olduğu için kalite geçişlerindeki harcanacak olan enerjiyi minimum yapacak olan çok hedefli programlama denklemi gerçekleştirmektir. Burada kısıtlar şeridin kalite bazına olan tavlama sıcaklıkları, şeridin tavlama bölgesindeki kalma zamanı (40 saniyeden büyük olmalı) ve ısıtıcıların ısıtma değerleridir. Bu denklemde;

Amaç:

- ✓ Kaliteye, maksimum hıza, genişliğe ve kalınlığa bağlı olarak en uygun ısı değerlerini bulmak ve enerji verimliliği sağlamaktır.

Kısıtlamalar:

- ✓ Kalitelerin belirli sıcaklık aralıkları arasında çekilmesini sağlamak.

- ✓ Isıtıcıların verebileceği maksimum kalori miktarı
- ✓ Şeridin alabileceği genişlik ve kalınlık sınırlamaları
- ✓ Fırının alabileceği maksimum şerit uzunluğu
- ✓ Hattın maksimum ve minimum hız miktarı (Üretim hızı şeridin 40 m/dakika içinde fırın bölgesinden geçilmesini sağlamalı.)

$$\text{Gerekli ısı miktarı} = C_p * g * w * t * v * 60 * (T_2 - T_1) * L$$

C_p	= özgül ısı
g	= özgül ağırlık
w	= şerit genişliği
t	= şerit kalınlığı
v	= şerit hızı
T_n	= sıcaklık
L	= fırın uzunluğu

İşlem kısıtları:

Kalınlık sınırlamaları:

$$0.40 \text{ mm} \leq t \leq 2.00 \text{ mm}$$

Genişlik sınırlamaları:

$$625 \text{ mm} \leq w \leq 1525 \text{ mm}$$

Hız sınırlamaları:

$$30 \text{ m/d.} \leq v \leq 320 \text{ m/d.}$$

Kalite sıcaklıkları sınırlamaları, Tablo 11’da verilmiştir.

Fırın uzunluğu = RTH1 uzunluk + RTH2 uzunluk

$$L = 292 + 278 = 570 \text{ metre}$$

Toplam ısıtma kapasitesi fırın boyunca (375 adet radyan tüp için)

Değişkenler:

1. bobin için şerit sıcaklığı x_1 ve artan değer d_1^-

2. bobin için şerit sıcaklığı x_2 ve artan değer d_2^-
3. bobin için şerit sıcaklığı x_3 ve artan değer d_3^-
4. bobin için şerit sıcaklığı x_4 ve artan değer d_4^-
5. bobin için şerit sıcaklığı x_5 ve artan değer d_5^-
6. bobin için şerit sıcaklığı x_6 ve artan değer d_6^-
7. bobin için şerit sıcaklığı x_7 ve artan değer d_7^-
8. bobin için şerit sıcaklığı x_8 ve artan değer d_8^-
9. bobin için şerit sıcaklığı x_9 ve artan değer d_9^-
10. bobin için şerit sıcaklığı x_{10} ve artan değer d_{10}^-

Sabitler:

Özgül ağırlık: $7,80 \cdot 10^3 \text{ kg/m}^3$

Özgül ısı: $0,1170 \text{ kilo Kalori / kg } ^\circ\text{C}$

Fırın Isı Kapasitesi = $3440000 \text{ Cal} \cdot \text{m} / \text{h}$

Sabit $C = C_p \cdot g \cdot 60 \cdot L$

Denklemlerin ve Kısıtların oluşturulması:

$$0,40 \text{ mm} \leq t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10} \leq 2,00 \text{ mm}$$

$$625 \text{ mm} \leq w_1, w_2, w_3, w_4, w_5, w_6, w_7, w_8, w_9, w_{10} \leq 1525 \text{ mm}$$

$$1800 \text{ m/saat} \leq v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10} \leq 19200 \text{ m/saat}$$

Şerit sıcaklıkları kaliteye bağlı olarak Tablo 11’de verilmiştir.

Hedef Programlama Denklemlerinin ve Kısıtların Oluşturulması

$$C \cdot w_1 \cdot t_1 \cdot v_1 \cdot (x_1 - x_0) \leq 3440000$$

$$C \cdot w_2 \cdot t_2 \cdot v_2 \cdot (x_2 - x_1) \leq 3440000$$

$$C \cdot w_3 \cdot t_3 \cdot v_3 \cdot (x_3 - x_2) \leq 3440000$$

$$C \cdot w_4 \cdot t_4 \cdot v_4 \cdot (x_4 - x_3) \leq 3440000$$

$$C \cdot w_5 \cdot t_5 \cdot v_5 \cdot (x_5 - x_4) \leq 3440000$$

$$C * w_6 * t_6 * v_6 * (x_6 - x_5) \leq 3440000$$

$$C * w_7 * t_7 * v_7 * (x_7 - x_6) \leq 3440000$$

$$C * w_8 * t_8 * v_8 * (x_8 - x_7) \leq 3440000$$

$$C * w_9 * t_9 * v_9 * (x_9 - x_8) \leq 3440000$$

$$C * w_{10} * t_{10} * v_{10} * (x_{10} - x_9) \leq 3440000$$

$$1800 \leq v_1 \leq 19200 \text{ m/h}$$

$$1800 \leq v_2 \leq 19200 \text{ m/h}$$

$$1800 \leq v_3 \leq 19200 \text{ m/h}$$

$$1800 \leq v_4 \leq 19200 \text{ m/h}$$

$$1800 \leq v_5 \leq 19200 \text{ m/h}$$

$$1800 \leq v_6 \leq 19200 \text{ m/h}$$

$$1800 \leq v_7 \leq 19200 \text{ m/h}$$

$$1800 \leq v_8 \leq 19200 \text{ m/h}$$

$$1800 \leq v_9 \leq 19200 \text{ m/h}$$

$$1800 \leq v_{10} \leq 19200 \text{ m/h}$$

$$x_1 + d_1^- = x_{1 \min}$$

$$x_2 + d_2^- = x_{2 \min}$$

$$x_3 + d_3^- = x_{3 \min}$$

$$x_4 + d_4^- = x_{4 \min}$$

$$x_5 + d_5^- = x_{5 \min}$$

$$x_6 + d_6^- = x_{6 \min}$$

$$x_7 + d_7^- = x_{7 \min}$$

$$x_8 + d_8^- = x_{8 \min}$$

$$x_9 + d_9^- = x_{9 \min}$$

$$x_{10} + d_{10}^- = x_{10 \min}$$

$x_{i \min}$ değerleri kaliteye bağlı olarak Tablo-11'deki kaliteye bağlı en küçük sıcaklık değerlerine karşılık gelmektedir.

$x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}$ değerleri kaliteye bağlı olarak Tablo-11'deki sıcaklık aralıklarında değişebilmektedir.

Buna göre oluşturulan minimizasyon denklemi:

$$\text{Minimum } \sum_{i=1}^n d_i^-$$

Yukarıda anlatılan optimizasyon denkleminin uygulaması analiz bölümünde anlatılmıştır.

3.2.2 Yapay Sinir Ağının Oluşturulması

İkinci problemimizde belirli bir fonksiyon akışı olmadığından ve bilgisayara alınan veriler sayesinde makine öğrenmesi yapacağımızdan yapay sinir ağı kullanılmasına karar verilmiştir. 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 zordur veya mümkün değildir. O nedenle, yapay sinir ağları bilim dalının, programlanması çok zor veya mümkün olmayan olaylar için geliştirilmiş kendi kendini geliştirebilen bilgi işleme ile ilgilenen bir bilgisayar bilim dalı olduğu söylenebilir.

Fakat yapay sinir ağı oluştururken iki sorun karşımıza çıkmaktadır. Bunlardan birincisi hangi YSA yapısının kullanılacağı, ikincisi ise ara katman sayısı ve katmanlardaki nöron sayılarıdır. Eğer az sayıda nöron ile ağı oluşturulursa düşük eşleme (underfitting) denilen performansın düşük çıkacağı ve tüm verileri kapsamayan bir durum ile karşı karşıya kalınabilir.

Bunun yanında çok fazla sayıda nöron ile ağı oluşturulursa bu durumda da aşırı uyum (overfitting) denilen durum meydana gelir. Eğitim sırasında istenilenden daha başarılı bir hata değerine ulaşılabilir. Buna rağmen örnek verilerde ya da gerçek uygulamalarda büyük hata değerleri ile karşılaşılabilir. Bu duruma aşırı uyum veya ezberletme denir.

Ara katmanda bulunan her nöronu eğitecek kadar eğitim kümesi bulunmaz ise doğabilecek diğer bir problem ise geç öğrenmedir. Buna göre sistemdeki bütün sinir hücrelerinin eğitimi normalden çok daha uzun zaman almaktadır.

Aşırı uyum sorununa yol açmayacak nöronların sayısı aşağıdaki denklemdeki gibi bulunabilir (stackexchange, 2016).

$$N_n = N_s / (\alpha * (N_i + N_o))$$

N_n = Ara katman nöron sayısı

N_i = Giriş nöron sayısı

N_o = Çıkış nöron sayısı

N_s = Eğitim setindeki örnek sayısı

α = Rasgele ölçeklendirme faktörü (Genellikle 2 – 10 arası olur.)

Genellikle $\alpha = 2$ faktörü ile denemelere başlanıp 10'a kadar denemeler yapılır. Bu arada en düşük hata değerini veren nöron sayısı kabul edilir.

Yapay sinir ağlarının eğitilmesi için Matlab yazılımı kullanılmıştır. Matlab yazılımında bulunan newff fonksiyonunun çeşitli alternatifleri denenerek minimum hata değeri ile sonuç alınması hedeflenmiştir.

Newff matlab fonksiyonu bir ileri beslemeli geri yayılım ağı oluşturur. Aynı zamanda ekranda bir toolbox ekranı olarak gözükür. Ek-10'da örnek bir programlama örneği verilmiştir. Orada da görüldüğü üzere parametreleri aşağıdaki gibidir:

$$Net = newff (PR, [S_1 S_2 S_3 \dots S_i], \{ TF_1 TF_1 TF_1 \dots TF_i \}, BTF, BLF, PF)$$

PR – R x 2 matris (Giriş ve çıkış değerleri verilir.)

S_i – i. katman için nöron sayısı (N katman için ayrı ayrı girilir.)

TF_i – i. katmanın transfer fonksiyonunu verir. Eğer herhangi bir değer girilmemiş ise “tansig” fonksiyonu giriş fonksiyonu olarak kullanılır.

BTF – İstedğimiz ağ eğitim fonksiyonu girilir. Eğer herhangi bir değer girilmemiş ise “traingdx” fonksiyonu eğitim fonksiyonu olarak kullanılır. Bu fonksiyonların kullandığı eğitim algoritmaları ise Tablo 14'de verilmiştir.

BLF – Ağı ağırlık ve eşik değerlerini öğrenme fonksiyonu girilir. Eğer herhangi bir değer girilmemiş ise “learnsgdm” fonksiyonu eğitim fonksiyonu olarak kullanılır.

PF – Performans fonksiyonudur. Değer girilmemiş ise “mse” kullanılır.

Geri Yayılım Ağı'nı kullanan newff fonksiyonu ile ilgili temel kullanım parametreleri

ve varsayılan değerleri Tablo 13’de verilmiştir.

Tablo 13
Matlab Newff Fonksiyonu Temel Kullanım Parametreleri

Parametreler	Ön Değerler	Açıklama
net.trainParam.epochs	1000	Maksimum eğitim iterasyon sayısı
net.trainParam.goal	0	Hedef performans
net.trainParam.max_fail	6	Maksimum doğrulama hatası
net.trainParam.min_grad	1e-7	Minimum performans gradyanti
net.trainParam.mu	0.001	Başlangıç μ değeri
net.trainParam.mu_dec	0.1	μ azalma faktörü
net.trainParam.mu_inc	10	μ arttırma faktörü
net.trainParam.mu_max	1e10	Maksimum μ
net.trainParam.show	25	Gösterimler arası iterasyon sayısı
net.trainParam.showCommandLine	False	Komut satırı çıktısı üretme
net.trainParam.showWindow	true	Eğitim ekranını göster
net.trainParam.time	inf	Eğitim için maksimum zaman

Kaynak: Mathworks,2015

Show parametresi, kaç tekrarda bir eğitim durumunun ekrana yansıtılacağını belirlemektedir. Epoch parametresi öğrenme boyunca kaç tekrar yapılacağını, goal parametresi hedeflenen maksimum hata değerini verir. Epoch veya goal parametrelerinden birisi belirlenen değere ulaştığında eğitim işlemi durdurulur. Tablo 14’de newff fonksiyonunun kullandığı bazı algoritmalar verilmiştir.

Tablo 14
YSA Geri Yayılım Ağı Eğitim Algoritmaları

Fonksiyon	Algoritma
trainlm	Levenberg-Marquardt
trainbr	Bayes Regülasyonu (Bayesian Regularization)
trainbfg	BFGS Quasi-Newton
trainrp	Esnek Geri Bildirim (Resilient Backpropagation)
trainscg	Ölçekli Eşlenik Gradyan (Scaled Conjugate Gradient)
traincgb	Eşlenik Gradyan ile Powell/ Beale Tekrar Başlatma (Conjugate Gradient with Powell/Beale Restarts)
traincgf	Fletcher-Powell Eşlenik Gradyan (Fletcher-Powell Conjugate Gradient)
traincgp	Polak-Ribière Eşlenik Gradyan (Polak-Ribière Conjugate Gradient)
trainoss	Bir Adım Kesişen (One Step Secant)
traingdx	Değişken Öğrenim Oranlı İniş Gradyan (Variable Learning Rate Gradient Descent)
traingdm	Momentum ile Eğim Azaltma (Gradient Descent with Momentum)
traingd	İniş Gradyan (Gradient Descent)

Kaynak: Mathworks (1), 2015

Yukarıdaki algoritmalar ile ilgili olarak yapılan bazı hız ve bellek karşılaştırmaları aşağıdaki tablolarda belirtilmiştir.

Hız ve Bellek Karşılaştırması

Verilen bir problem için hangi eğitim algoritmasının en hızlı olacağını bilmek çok zordur. Sorunun karmaşıklığı, eğitim kümesindeki veri noktalarının sayısına, ağıdaki ağırlık ve eşik değerlerinin sayısına, hata hedefine ve ağı model tanıma (Diskriminant Analiz) için mi yoksa fonksiyon yaklaşımı (regresyon) için mi kullanılıp kullanılmadığına bağlıdır. Çeşitli eğitim algoritmalarının ölçüt karşılaştırmaları aşağıdaki tablolarda verilmiştir. İleri beslemeli ağılar altı farklı problem üzerinde eğitilmiştir. Problemlerden üç tanesi örüntü tanıma kategorisinde, diğeri ise fonksiyon yaklaşım kategorisinde yer almaktadır. Sorunlardan ikisi basit "oyuncak" problemler iken diğeri "gerçek dünya" problemleridir. Çeşitli farklı mimarilere ve karmaşıklıklara sahip ağıları kullanılmıştır.

Aşağıdaki tabloda, test edilen algoritmalar ve bunları tanımlamak için kullanılan kısaltmalar listelenmiştir (Tablo 15).

Tablo 15
YSA Geri Yayılım Ağı Eğitim Algoritmalarının Kısaltılmış İsimleri

Kısaltma	Algoritma
LM	trainlm - Levenberg-Marquardt
RP	trainrp - Resilient Backpropagation
SCG	trainscg - Scaled Conjugate Gradient

Tablo 16’da altı adet kıyaslama fonksiyonu, problem yapısı, ağ yapısı, süre hedefi ve bilgisayarın bazı özellikleri verilmektedir.

Tablo 16
Kıyaslama için Kullanılan Fonksiyonlar

Problem	Problem Çeşidi	Ağ Yapısı	Hedef	Bilgisayar
SIN	Fonksiyon yaklaşımı	1-5-1	0.002	Sun Sparc 2
PARİTY	Örüntü tanıma	3-10-10-1	0.001	Sun Sparc 2
MOTOR	Fonksiyon yaklaşımı	2-30-2	0.005	Sun Enterprise 4000
KANSER	Örüntü tanıma	9-5-5-2	0.012	Sun Sparc 2
KOLESTROL	Fonksiyon yaklaşımı	21-15-3	0.027	Sun Sparc 20
DİYABET	Örüntü tanıma	8-15-15-2	0.05	Sun Sparc 20

Kaynak: Mathworks (2), 2017

Tablo 16’da bahsedilen kıyaslama problemleri için Tablo 17, Tablo 18, Tablo 19, Tablo 20, Tablo 21 ve Tablo 22’de kıyaslama sonuçları verilmiştir.

Tablo 17
Sin Veri Seti için Kıyaslama Sonuçları

Algoritma	Ort. Zaman(s)	Oran	Min. Zaman (s)	Maks. Zaman (s)	Std. (s)
LM	1.14	1.00	0.65	1.83	0.38
RP	5.67	4.97	2.66	17.24	3.72
SCG	6.09	5.34	3.18	23.64	3.81

Kaynak: Mathworks (2), 2017

Tablo 18
Parity(Eşlik) Veri Seti için Kıyaslama Sonuçları

Algoritma	Ort. Zaman(s)	Oran	Min. Zaman (s)	Maks. Zaman (s)	Std. (s)
RP	3.73	1.00	2.35	6.89	1.26
SCG	4.09	1.10	2.36	7.48	1.56
LM	13.07	3.50	6.48	23.78	4.96

Kaynak: Mathworks (2), 2017

Tablo 19
Motor Veri Seti için Kıyaslama Sonuçları

Algoritma	Ort. Zaman(s)	Oran	Min. Zaman (s)	Maks. Zaman (s)	Std. (s)
LM	18.45	1.00	12.01	30.03	4.27
SCG	36.02	1.95	19.39	52.45	7.78
RP	65.91	3.57	31.83	134.31	34.24

Kaynak: Mathworks (2), 2017

Tablo 20
Kanser Veri Seti için Kıyaslama Sonuçları

Algoritma	Ort. Zaman(s)	Oran	Min. Zaman (s)	Maks. Zaman (s)	Std. (s)
RP	83.41	1.04	59.51	109.39	13.44
SCG	86.58	1.08	41.21	112.19	18.25
LM	110.33	1.37	58.94	201.07	38.20

Kaynak: Mathworks (2), 2017

Tablo 21
Kolesterol Veri Seti için Kıyaslama Sonuçları

Algoritma	Ort. Zaman(s)	Oran	Min. Zaman (s)	Maks. Zaman (s)	Std. (s)
SCG	99.73	1.00	83.10	113.40	9.93
LM	261.50	2.62	103.52	398.45	102.06
RP	1519.00	15.23	581.17	2256.10	557.34

Kaynak: Mathworks (2), 2017

Tablo 22
Diyabet Veri Seti için Kıyaslama Sonuçları

Algoritma	Ort. Zaman(s)	Oran	Min. Zaman (s)	Maks. Zaman (s)	Std. (s)
RP	323.90	1.00	187.43	576.90	111.37
SCG	390.53	1.21	267.99	487.17	75.07
LM	1028.10	3.17	802.01	1269.50	166.31

Kaynak: Mathworks (2), 2017

Bu yapılan testler sonucunda trainrp fonksiyonunun, trainlm fonksiyonunun ve trainscg fonksiyonunun daha başarılı olduğu anlaşılmıştır. Biz bu 3 fonksiyonu çalışmalarımızda kullanacağız. Bu fonksiyonların kısa tanıtımları aşağıda verilmiştir.

- **Trainrp – Bayesian Regularization Geri Yayılm Fonksiyonu**

Trainrp, Levenberg-Marquardt optimizasyon algoritmasına göre ağırlık ve eşik değerlerini güncelleyen bir ağ eğitim fonksiyonudur. Hataların karesinden ve ağırlıklardan oluşan bir kombinasyonu minimize eder ve doğru kombinasyonu belirleyerek iyi genelleştirilmiş bir ağ oluşturmasını sağlar. Bu işleme Bayesci regülasyon denir. Bu fonksiyon, hesaplamalarda performansı, hataların ortalaması veya hata karelerinin toplamı olduğunu varsayan Jacobian hesaplama yöntemini kullanır. Bu nedenle, bu fonksiyonla eğitilen ağlar, mse veya sse performans fonksiyonunu kullanmalıdır. Bu Bayesci düzenleme, Levenberg-Marquardt algoritması içinde gerçekleşir. Geri yayılım, ağırlık ve eşik değişkenleri X'e göre performansın Jacobian jX'sini hesaplamak için kullanılır. Her değişken, Levenberg-Marquardt'a göre ayarlanır (Mathworks (5),2017);

$$jj = jX * jX$$

$$Je = jX * E$$

$$DX = - (jj + I * \mu) \setminus je$$

Burada tüm hataları ve I ise birim matrisi gösterir.

Yukarıda gösterilen değişiklik, düşük bir performans değeri ile sonuçlanıncaya kadar adaptif değer μ , μ_{inc} tarafından arttırılır. Değişiklik daha sonra şebekeye yapılır ve μ , μ_{dec} kadar azaltılır.

Eğitim, bu koşullardan herhangi biri meydana geldiğinde durur:

- Maksimum epoch sayısı (tekrar sayısı) ulaşılır.
- Maksimum süre aşılır.

- Performans hedefe indirgenir.
- Performans gradyan min_grad'ın altına düştü.

μ , $\mu_{\max}$ 'aşar. (Mathworks (5),2017)

- **Trainscg – Scaled Conjugate Gradient Geri Yayılım Fonksiyonu**

Eşlenik gradyant algoritmalar her iterasyonda bir vektör araması gerçekleştirir. Bu vektör araması hesaplama açısından pahalı, çünkü her eğitim girişi için ağ yanıtının her arama için birkaç kez hesaplanmasını gerektirir. Moller tarafından geliştirilen ölçekli eşlenik gradyan algoritması (SCG), zaman alan tarama işlemini önlemek için tasarlanmıştır. Bu algoritma birkaç satırda açıklanamayacak kadar karmaşıktır, ancak temel fikir, eşlenik gradyan yaklaşımı ile model-güven bölgesi yaklaşımını (Levenberg-Marquardt algoritmasında da kullanılır.) birleştirmektir.

Trainscg fonksiyonu, diğer eşlenik gradyant algoritmalarından daha fazla iterasyon gerektirebilir, ancak her iterasyonda hesaplamaların sayısı önemli ölçüde azaltılır; çünkü herhangi bir vektör araması yapılmaz (Mathworks (4),2017).

- **Trainlm - Levenberg-Marquardt Geri Yayılım Fonksiyonu**

Levenberg-Marquardt optimizasyon algoritmasına bağlı olarak ağırlıklarını ve eşik değerlerini güncelleyen ağ eğitim fonksiyonudur. Eğitim fonksiyonları içinde en hızlı geri yayılım algoritmasıdır. Diğer algoritmalarla göre daha fazla bellek gerektirmesine rağmen, denetimli algoritmalar arasında birinci seçenek olarak önerilir. Bu fonksiyonda doğrulama vektörleri doğrulama vektörleri üzerinde ağ performansının artmadığı veya bir örnekteki maksimum hata durumuna erişildiği durumlarda eğitimin durdurulmasına neden olur. Test vektörleri ağı iyi genelleştirdiğini ve eğitim üzerinde herhangi bir etkisinin bulunmadığını kontrol etmek için kullanılır. Trainlm, newff fonksiyonu için varsayılan eğitim fonksiyonudur (Mathworks (3),2017).

Quasi-Newton yöntemleri gibi Levenberg-Marquardt algoritması, Hessian matrisini hesaplamaya gerek kalmadan ikinci dereceden eğitim hızına yaklaşmak üzere tasarlanmıştır. Performans işlevi kareler toplamı biçiminde olduğunda (İleri beslemeli YSA ağlarında kullanılan eğitime algoritmalarında normaldir), Hessian matrisi şu şekilde hesaplanabilir (Mathworks (3),2017):

$$H = J^T J$$

ve gradyanı şöyle hesaplanır.

$$\mathbf{g} = \mathbf{J}^T \mathbf{e}$$

$\mathbf{J}$ = Ağ hatalarının ağırlık ve eşik değerlerine göre birinci türevlerini içeren Jakobiyen matrisi

$\mathbf{e}$ = ağ hata vektörü

Jakobiyen matrisi, Hessian matrisinin hesaplanmasından daha az karmaşık olan standart bir geri yayılım tekniği ile hesaplanabilir. Levenberg-Marquardt algoritması aşağıdaki Newton benzeri bir güncelleme ile Hessian Matrisine yaklaşımı kullanır (Mathworks (3),2017).

$$\mathbf{x}_{k+1} = \mathbf{x}_k - [\mathbf{J}^T \mathbf{J} + \mu \mathbf{I}]^{-1} \mathbf{J}^T \mathbf{e}$$

Skaler μ sıfır olduğunda yukarıdaki formül yaklaşık Hessian Matrisi olur. μ büyük olduğunda ise formül kısa adımlı aşağı iniş gradyan metoduna doğru kayar.

Newton'un yöntemi, minimum hata yakınına olabildiğince hızlı ve doğru varmaktır, bu yüzden yukarıdaki formülde de amaç Newton'un yöntemine olabildiğince çabuk kaymaktır.

Böylece, μ , her başarılı adımdan sonra düşürülür (performans fonksiyonunda azalma) ve yalnızca geçici olarak bir deneme adımı performans fonksiyonunu artıracak zaman arttırılır. Bu şekilde, algoritmanın her yinelemesinde performans fonksiyonu her zaman azaltılacaktır. (Mathworks (3),2017)

3.3 Veri Toplama

Birinci çalışma için verileri çevrimiçi olarak alıp veritabanına yazan program parçacığı mevcut çalışan operatör ekranları programına yerleştirilmiştir. Buradan alınan bilgiler ışığında optimizasyon programı çalıştırılmaktadır [Ek-5].

İkinci çalışma için ise çok uzun süreli veri toplanması, bu veriler içinden hatalı ve aynı tip olanların ayıklanarak veri analizi için hazır hale getirilmesi gerekmektedir. Bu amaçla veri toplanması şu şekilde olmaktadır.

Şekil 3'da verilen STH'ndaki bobin ve şeridin takip edilmesi iki şekilde olmaktadır. Hattın giriş konveyörü, çıkış konveyörü ve paketleme bölümleri bobin temelli izlenirken diğer bölümler ise şerit temelli izlenmektedir. Şerit temelli izleme şerit üzerine açılan bir çentik ve hat içine konulan çentik detektörleri sayesinde yapılmaktadır. Bu nedenle hat izleme sistemlerinin aşağıdaki Şekil 32'deki gibi çalıştığı söylenebilir.

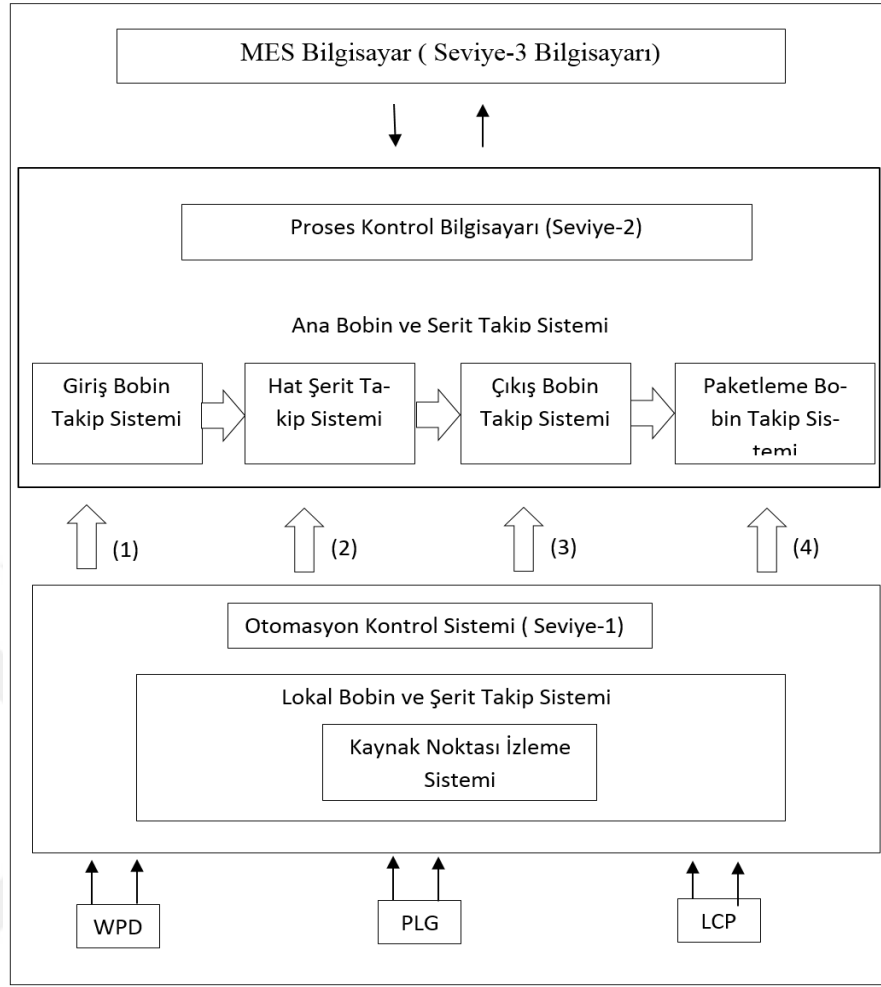
1. Ana bobin ve şerit takip sistemi (Proses kontrol tarafında)

- a. Giriş bobin takip sistemi
 - b. Hat içindeki şerit takip sistemi
 - c. Çıkış bobin takip sistemi
 - d. Paketleme bobin takip sistemi
2. Lokal bobin ve şerit takip sistemi (Elektriksel kontrol bilgisayarı tarafında)
 - a. Kaynak noktası takip sistemi
 3. Veri toplama sistemi (Elektriksel kontrol bilgisayarı tarafında)

Şekil 32’de de görüldüğü üzere otomasyon kontrol sisteminden aşağıdaki 4 sinyali alarak bobin takip sistemi yapılabilmektedir.

1. POR veya giriş konveyörüne herhangi bir bobin yüklendiğinde otomasyon sisteminden gelen sinyal.
2. Bobin takip için otomasyon sisteminden gelen ve bobinin hangi izleme bölgesine eriştiğini belirten sinyal.
3. Çıkış konveyör hareketleri ve önden giden bobinin çıkış gergi mandreline sonu geldiğinde kesildiğini belirten sinyal.
4. Paketleme hattı için oluşturulmuş sinyalleri.

Hat şerit takip sistemi bobin öndeki bobine kaynak yapıldığı anda başlar ve bobinin gergi mandrelinden ayrılışına kadar devam eder. Bu ayrılış bobin kuyruğunun kesme makası ile kesilmesi sonucunda olmaktadır. Üretilen bobin bilgileri MES bilgisayarına akmakta olup bir sonraki hat için girdi olmaktadır.



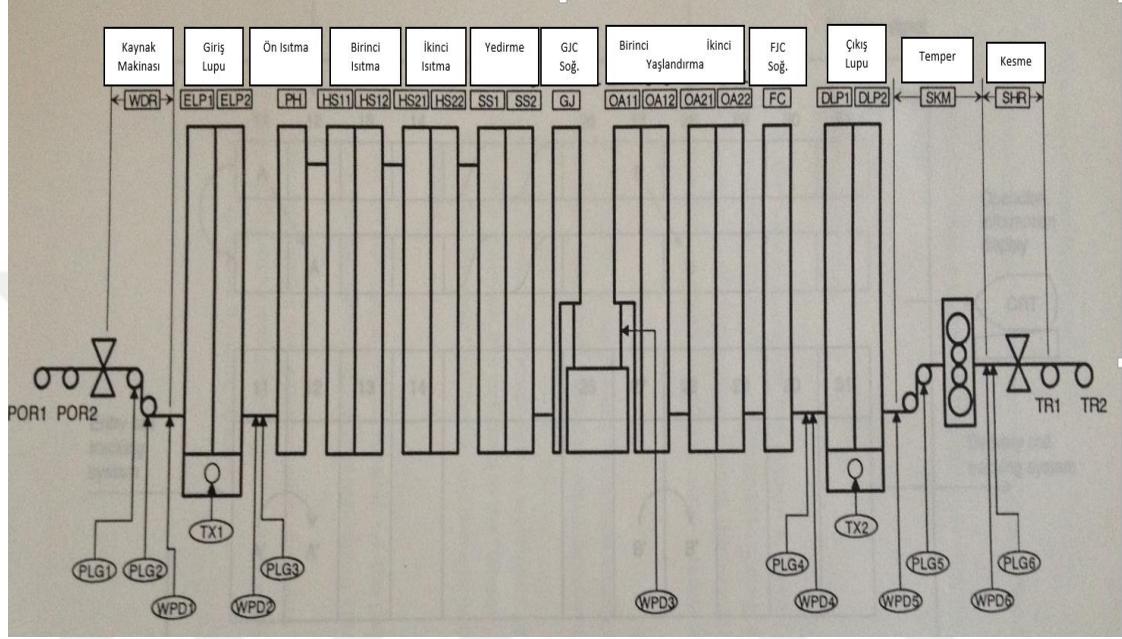
Şekil 32. Bobin Takip Sistemi Genel Şeması

Kaynak: NKK Corporation, 1995

Ayrıca eğer bir problem anında Seviye-1 bilgisayarından sinyal alınamadığı anda operatör bu sinyali Seviye-2 ekranlarından simule edebilmektedir.

Şerit takip sistemi ise bobin takip sistemi ile aynı anda yürütülmekte ve kaynak makinasının “kaynak tamamlandı” sinyalini göndermesi ile başlamakta ve kaynak noktasının kesme makasını kesmesi ile bitmektedir. Bu sistemde üç ayrı takip donanımı bulunmaktadır. 4 adet WPD hattın çeşitli yerlerine konulmuştur. Bu detektörlerin takip edebilmesi için şerit başlangıcında bir oyuntu açılmaktadır. Bu oyuntu sayesinde hat uzunluk takibi kendini düzeltebilmektedir. Ayrıca şerit ucunun bölgeler arasındaki geçişlerinin takibi yapılabilmektedir. Giriş ve çıkış taraflarında bulunan şerit depolama birimlerinin içindeki şerit varlığına göre konumlarını belirleyen taşıyıcı pozisyon sensörleri de (LPC)

şerit takip sisteminin en önemli unsurlarından biridir. Bu sensörler sayesinde şerit depolama birimi içerisinde ne kadar şerit bulunduğu bilgisi hesaplanmaktadır. PLG'ler ise hattın hızına göre sinyal üreterek üzerinden geçen gerçek şerit uzunluğunu verebilen donanımlardır. WPD, LPC ve PLG'lerin hat içindeki konumları aşağıdaki Şekil 33'deki gibidir.



Şekil 33. Hat Bobin İzleme Bölgeleri Ve Sensör Konumları

Kaynak: NKK Corporation, 1995

Bunlar sayesinde şerit belirli bir yerden geçen uzunlukları hesaplanabilmektedir. Hatta giren bobin takibi için bölgeler aşağıdaki gibidir (Tablo 23).

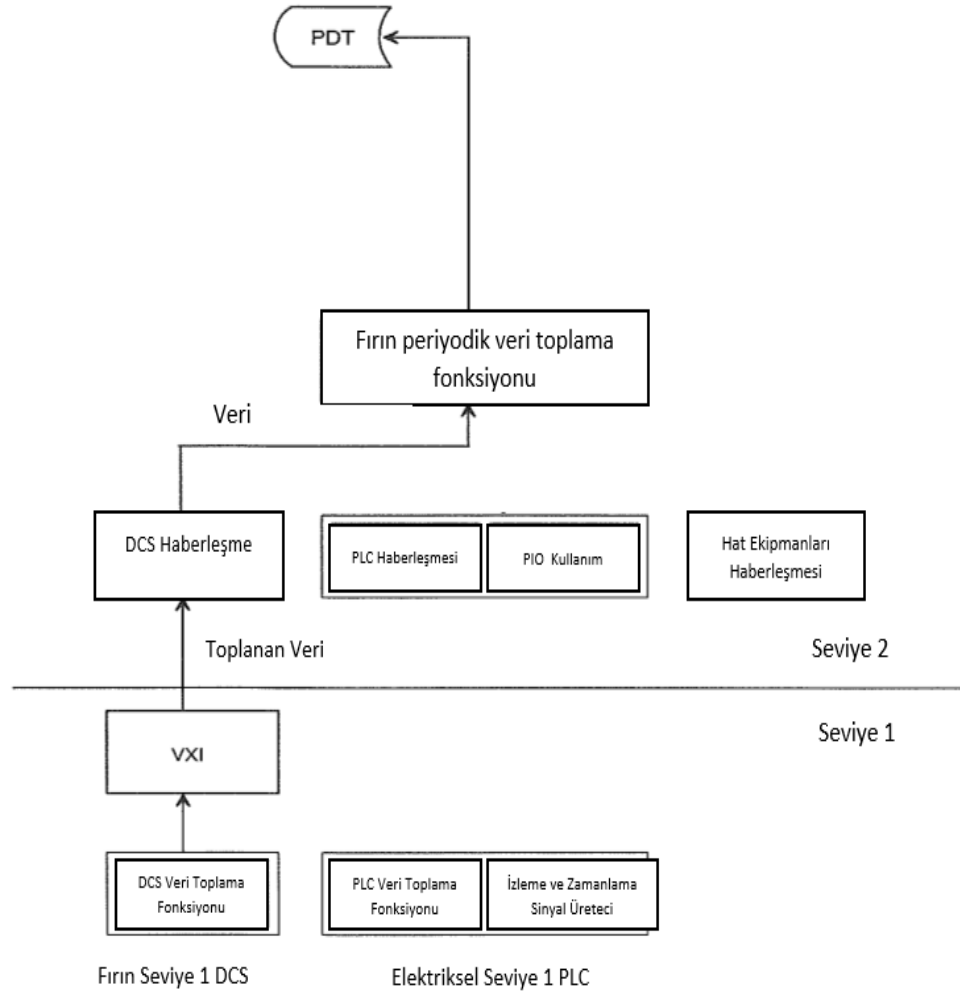
Tablo 23
Hat Şerit Takip Sistemi Bölgeleri

Bölge No	Kod	Şerit Takip Bölge İsmi
15	HS11	1.Bölge Isıtma Kısım 1
16	HS12	1.Bölge Isıtma Kısım 2
17	HS21	2.Bölge Isıtma Kısım 1
18	HS22	2.Bölge Isıtma Kısım 2

Kaynak: NKK Corporation, 1995

Belirli zaman aralıklarında verilerin alınarak kaydedilmesi

Seviye-1 DCS bilgisayarı kısa zaman aralıklarında fırının durumunu gözlemler ve bu zaman aralıklarındaki o anki değerleri önceden belirlenmiş DCS içindeki alanlara kaydeder. Şekil 35 'de tipik bir periyodik veri toplama fonksiyonunun işleyişi görülmektedir.



Şekil 35: Fırın Periyodik Veri Toplama Fonksiyonu

Kaynak: NKK Corporation, 1995

Seviye-2 Bilgisayarı da bu veriyi alarak kendi bilgisayarındaki periyodik veri toplama tablosuna kaydeder. Seviye-1 bilgisayarının depolama kapasitesi düşük olduğu için bu yola başvurulmuştur. Ayrıca toplama değerlerini ve sayma değerlerini de bu tablo içerisine kaydını yapar. Bu toplanan veriler daha sonra ortalama veri bulmada, set değeri parametrelerinin düzeltilmesinde ve raporlama işlemlerinde kullanılmaktadır. Aşağıdaki Tablo 24'de anlık ve periyodik olarak tutulan verilerin listesi bulunmaktadır.

Tablo 24
Ürün Veri Toplama İsim Listesi

N o	Veri Adı	Birim	Kaynak	Zamanlaması
1	PH Şerit Sıcaklığı	°C	DCS	Her bir bobin için her 100 milisaniyede
2	HS2 Şerit Sıcaklığı	°C	DCS	Her bir bobin için her 100 milisaniyede
3	PH #1 Bölge Fırın Sıcaklığı	°C	DCS	Her bir bobin için her 100 milisaniyede
4	PH #1 Bölge Fırın Sıcaklığı	°C	DCS	Her bir bobin için her 100 milisaniyede
5	PH #2 Bölge Fırın Sıcaklığı	°C	DCS	Her bir bobin için her 100 milisaniyede
6	HS1 #1 Bölge Fırın Sıcaklığı	°C	DCS	Her bir bobin için her 100 milisaniyede
7	HS1 #2 Bölge Fırın Sıcaklığı	°C	DCS	Her bir bobin için her 100 milisaniyede
8	HS1 #3 Bölge Fırın Sıcaklığı	°C	DCS	Her bir bobin için her 100 milisaniyede

Tablo 24'ün devamı

9	HS1 #4 Bölge Fırın Sıcaklığı	°C	DCS	Her bir bobin için her 100 milisaniyede
10	HS2 #1 Bölge Fırın Sıcaklığı	°C	DCS	Her bir bobin için her 100 milisaniyede
11	HS2 #2 Bölge Fırın Sıcaklığı	°C	DCS	Her bir bobin için her 100 milisaniyede
12	HS2 #3 Bölge Fırın Sıcaklığı	°C	DCS	Her bir bobin için her 100 milisaniyede
13	HS2 #4 Bölge Fırın Sıcaklığı	°C	DCS	Her bir bobin için her 100 milisaniyede
14	Bobin Numarası	-	MES	Her bir bobin için 1 defa
15	Bobin Genişliği	-	MES	Her bir bobin için 1 defa
16	Bobin Kalınlığı	-	MES	Her bir bobin için 1 defa
17	Hat hızı	m/d	PLC	Her bir bobin için her 100 milisaniyede
18	Bobin Uzunluğu	m	PLC	Her bir bobin için her 100 milisaniyede

Verilerin Kaydedilmesi

Sunucu Bilgisayar’da hazırlanan veri yapısı haberleşme programı yardımıyla verilerin kaydedileceği bilgisayara gönderilmektedir. Bunun için sunucu bilgisayarda program değişikliği yapılmış, 3.3 ’de anlatılan olay temelli ve periyodik veriler alınarak bilgisayarda oluşturulan MYSQL veri tabanına kaydedilmeye başlanmıştır. İstemci tarafın sözde kodu (pseudocode) aşağıdaki gibidir.

Veri toplama sözde kodu

```
initialize failures to zero  
  
assign static values to variables  
  
create thread for receiving message from server  
  
send request message to server  
  
initialize timer for sending request every 15 seconds  
  
do while exit button is pressed  
    do while the message is received  
        assign values to structure  
        write to MYSQL database  
    end while  
end while
```

Veri tabanına kayıt edilen veriler aşağıda sıralanmıştır.

Bu sıralamaya göre;

PHSCoilNo, ön ısıtma bölümündeki bobin numarasını,
RTH1CoilNo, 1. radyan tüplü ısıtma bölümündeki bobin numarasını,
RTH2CoilNo, 2. radyan tüplü ısıtma bölümündeki bobin numarasını,
RTSCoilNo, Isı emdirme bölümü bobin numarasını,
PHSStripTempAct, ön ısıtma bölüm çıkışı bobin sıcaklığını,
RTH2StripTemp2Act, 2. radyan tüplü ısıtma bölümü çıkışı bobin sıcaklığını,
RTSStripTemp3Act, Isı emdirme bölümü çıkışı bobin sıcaklığı,

PHS1FurnaceTempAct, ön ısıtma bölümünde alınan ilk fırın sıcaklığını,
PHS2FurnaceTempAct, ön ısıtma bölümünde alınan ikinci fırın sıcaklığını,
RTH1FurnaceTemp1Act, 1. radyan tüplü ısıtma bölümü ölçülen ilk fırın sıcaklığını,
RTH1FurnaceTemp2Act, 1. radyan tüplü ısıtma bölümü ölçülen 2. fırın sıcaklığını,
RTH1FurnaceTemp3Act, 1. radyan tüplü ısıtma bölümü ölçülen 3. fırın sıcaklığını,
RTH1FurnaceTemp4Act, 1. radyan tüplü ısıtma bölümü ölçülen 4. fırın sıcaklığını,
RTH2FurnaceTemp1Act, 2. radyan tüplü ısıtma bölümü ölçülen ilk fırın sıcaklığını,
RTH2FurnaceTemp2Act, 2. radyan tüplü ısıtma bölümü ölçülen 2. fırın sıcaklığını,
RTH2FurnaceTemp3Act, 2. radyan tüplü ısıtma bölümü ölçülen 3. fırın sıcaklığını,
RTH2FurnaceTemp4Act, 2. radyan tüplü ısıtma bölümü ölçülen 4. fırın sıcaklığını,
RTSFurnaceTemp1Act, Isı emdirme bölümü ölçülen 1. fırın sıcaklığını,
RTSFurnaceTemp2Act, Isı emdirme bölümü ölçülen 2. fırın sıcaklığını,
RTHTubeTemp1, radyan tüplü ısıtma bölümü 1. tüp sıcaklığını,
RTHTubeTemp2, radyan tüplü ısıtma bölümü 2. tüp sıcaklığını,
RTHTubeTemp3, radyan tüplü ısıtma bölümü 3. tüp sıcaklığını,
RTHTubeTemp4, radyan tüplü ısıtma bölümü 4. tüp sıcaklığını,
RTHTubeTemp5, radyan tüplü ısıtma bölümü 5. tüp sıcaklığını,
RTHTubeTemp6, radyan tüplü ısıtma bölümü 6. tüp sıcaklığını,
RTHTubeTemp7, radyan tüplü ısıtma bölümü 7. tüp sıcaklığını,
RTHTubeTemp8, radyan tüplü ısıtma bölümü 8. tüp sıcaklığını,
RTHTubeTemp9, radyan tüplü ısıtma bölümü 9. tüp sıcaklığını,
RTHTubeTemp10, radyan tüplü ısıtma bölümü 10. tüp sıcaklığını,
RTHTubeTemp11, radyan tüplü ısıtma bölümü 11. tüp sıcaklığını,
RTHTubeTemp12, radyan tüplü ısıtma bölümü 12. tüp sıcaklığını,
RTHTubeTemp13, radyan tüplü ısıtma bölümü 13. tüp sıcaklığını,
RTHTubeTemp14, radyan tüplü ısıtma bölümü 14. tüp sıcaklığını,
RTHTubeTemp15, radyan tüplü ısıtma bölümü 15. tüp sıcaklığını,
RTHTubeTemp16, radyan tüplü ısıtma bölümü 16. tüp sıcaklığını,
RTHTubeTemp17, radyan tüplü ısıtma bölümü 17. tüp sıcaklığını,
RTHTubeTemp18, radyan tüplü ısıtma bölümü 18. tüp sıcaklığını,
RTHTubeTemp19, radyan tüplü ısıtma bölümü 19. tüp sıcaklığını,
RTHTubeTemp20, radyan tüplü ısıtma bölümü 20. tüp sıcaklığını,
CenterSpeed, merkez hat hızını göstermektedir.

Ana bilgisayardan bilgi alan ve veri tabanına kaydeden program yazılmıştır.

3.4 Uygulama ve Sonuçlar

1. problemin verileri hedef programlama denklemleri ve kısıtları, üretim hatlarında en çok üretilen ebatlar için Microsoft Solver Foundation yazılımı ile denenmiştir. Çeşitli denemeler yapılmış ve sistemin çalışma şartlarına uygunluğu sonuç bölümünde değerlendirilmiştir. Programın sözde kodu aşağıda verilmiştir.

Giriş konveyöründeki bobinlerin optimize tavlama sıcaklıklarını veren sözde kod

```
initialize failures to zero
assign static values to variables
connect to server for taking data
take data to array for entry conveyor
add decision variables (x1,x2,x3,x4,x5,x6,x7,x8,x9,x0) for coils
assign CP = 0.1170 kilo kalori /kg °C
assign specific weight SP = 7800 kg/m3
assign furnace length L = 570 m
assign furnace capacity FC = 34430000 kKal * m / saat
add constraints
    ( CP*SP*L *60* Width(i) * Thick(i) *speed(i)*(x(i)-x(i-1))) ≤ FC)
add constraints for temperature limits
add goal ( minimizing totals of decision variables)
print report every new coil enter to the conveyor
```

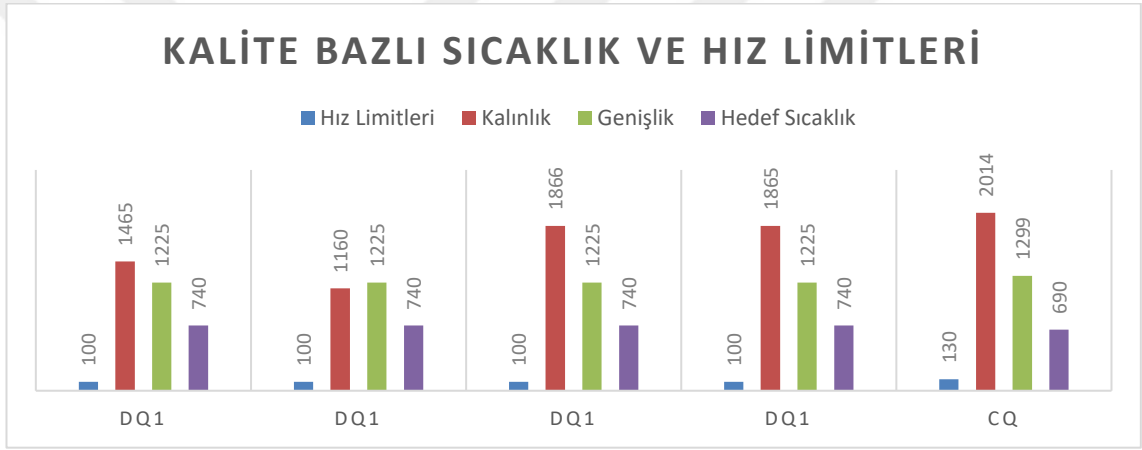
Programın kodu Ek-5’de verilmiştir. Programın çalıştırılması sonucu üretilen sonuçlar grafik olarak aşağıdaki gibi bulunmuştur. Bu arada hattın kapasitesine bağlı olarak çıkartılan maksimum hız tabloları Ek-3’de verilmiştir.

Uygulama sonucu elde edilen en uygun sıcaklık değerleri Tablo 25’den Tablo 35’e kadar olan tablolarda verilmiştir. Bu sıcaklıklara bağlı olarak elde edilen hız değerleri ise Şekil 36’dan Şekil 46’ya kadar olan şekillerde gösterilmiştir.

Tablo 25

1. Deneme için en uygun sıcaklık grafiği

1. Deneme					
Bobin Numarası	Hedef Hız	Kalite Kodu	Kalınlık	Genişlik	Hedef Sıcaklık
36228108000	100	DQ1	1465	1225	740
36228609000	100	DQ1	1160	1225	740
26228154100	100	DQ1	1866	1225	740
36228107000	100	DQ1	1865	1225	740
36229072000	130	CQ	2014	1299	690

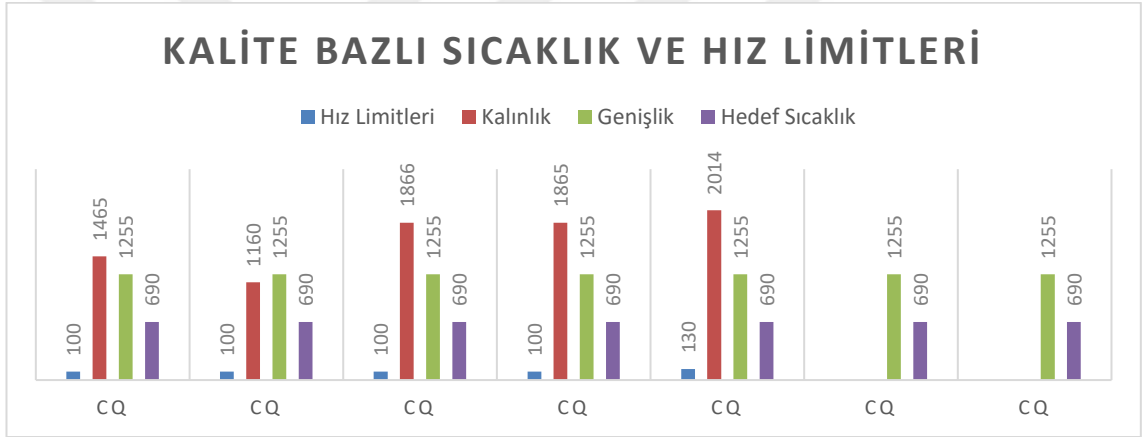


Şekil 36. 1. Deneme Kalınlık, Genişlik Ve Hıza Bağlı Optimum Hedef Sıcaklıkları

Tablo 26

2. Deneme için en uygun sıcaklık grafiği

2. Deneme					
Bobin Numarası	Hedef Hız	Kalite Kodu	Kalınlık	Genişlik	Hedef Sıcaklık
36228291000	130	CQ	2030	1255	690
36228100000	130	CQ	2030	1255	690
36228293000	130	CQ	2030	1255	690
36228399000	130	CQ	2030	1255	690
36228996000	130	CQ	2030	1255	690
36228501000	130	CQ	2030	1255	690
36229156000	130	CQ	2030	1255	690

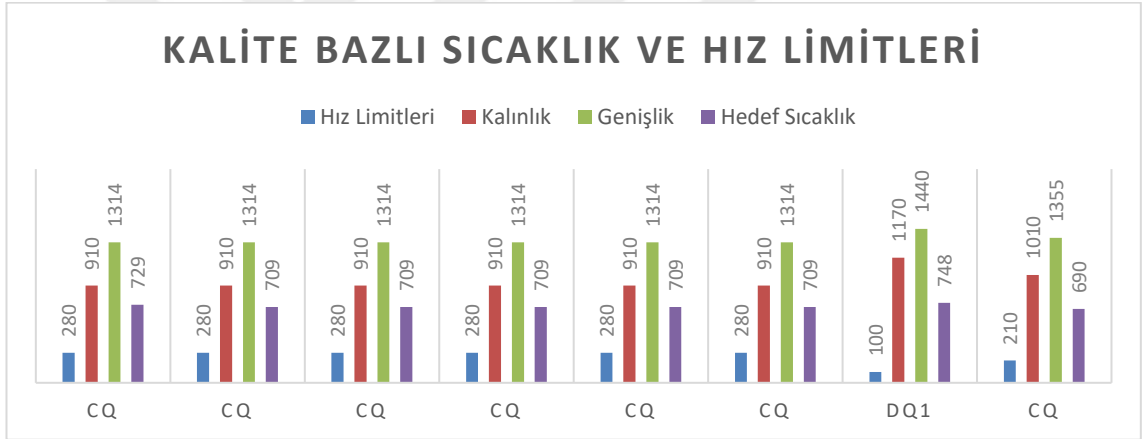


Şekil 37. 2. Deneme Kalınlık, Genişlik, Hıza Bağlı En Uygun Hedef Sıcaklıkları

Tablo 27

3. Deneme için en uygun sıcaklık grafiği

3. Deneme					
Bobin Numarası	Hedef Hız	Kalite Kodu	Kalınlık	Genişlik	Hedef Sıcaklık
26223129000	280	CQ	910	1314	729
36223128000	280	CQ	910	1314	709
36216797000	280	CQ	910	1314	709
26216796000	280	CQ	910	1314	709
36223126000	280	CQ	910	1314	709
36223034000	280	CQ	910	1314	709
26226887000	100	DQ1	1170	1440	748
36213845000	210	CQ	1010	1355	690

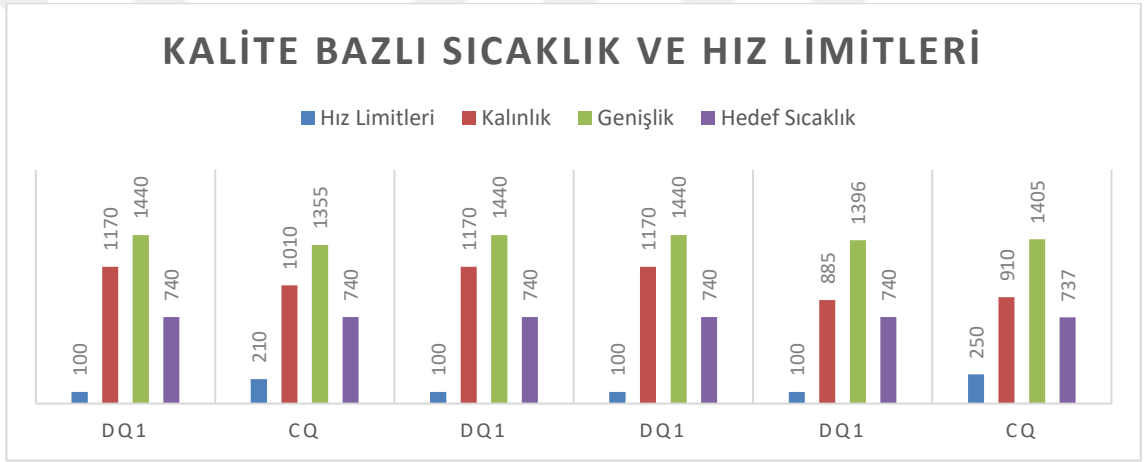


Şekil 38. 3. Deneme Kalınlık, Genişlik, Hıza Bağlı En Uygun Hedef Sıcaklıkları

Tablo 28

4. Deneme için en uygun sıcaklık grafiği

4. Deneme					
Bobin Numarası	Hedef Hız	Kalite Kodu	Kalınlık	Genişlik	Hedef Sıcaklık
26226887000	100	DQ1	1170	1440	740
36213845000	210	CQ	1010	1355	740
26221809000	100	DQ1	1170	1440	740
36226886000	100	DQ1	1170	1440	740
36226888000	100	DQ1	885	1396	740
26211113000	250	CQ	910	1405	737

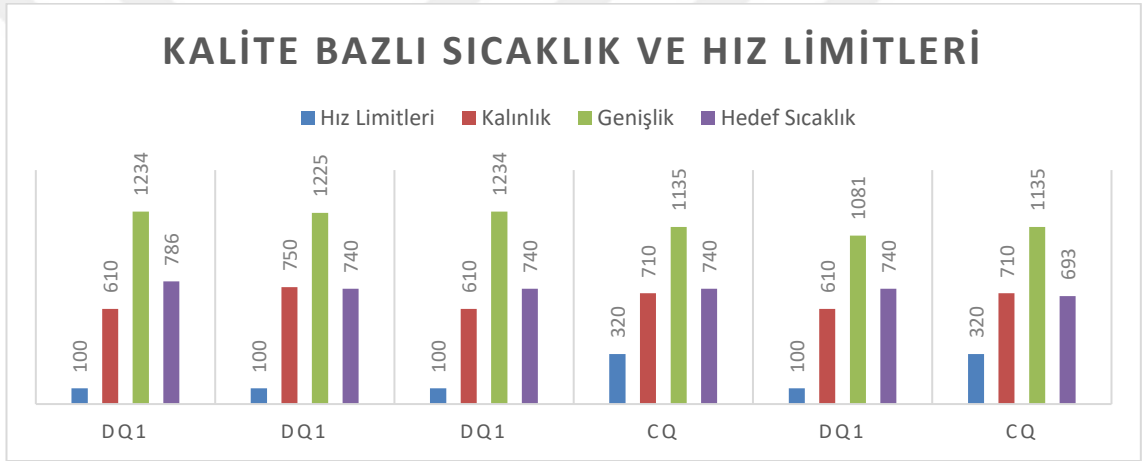


Şekil 39. Deneme Kalınlık, Genişlik, Hıza Bağlı En Uygun Hedef Sıcaklıkları

Tablo 29

5. Deneme için en uygun sıcaklık grafiği

5. Deneme					
Bobin Numarası	Hedef Hız	Kalite Kodu	Kalınlık	Genişlik	Hedef Sıcaklık
26227676000	100	DQ1	610	1234	786
36228106000	100	DQ1	750	1225	740
36227675000	100	DQ1	610	1234	740
26230755000	320	CQ	710	1135	740
26230238000	100	DQ1	610	1081	740
26230757000	320	CQ	710	1135	693

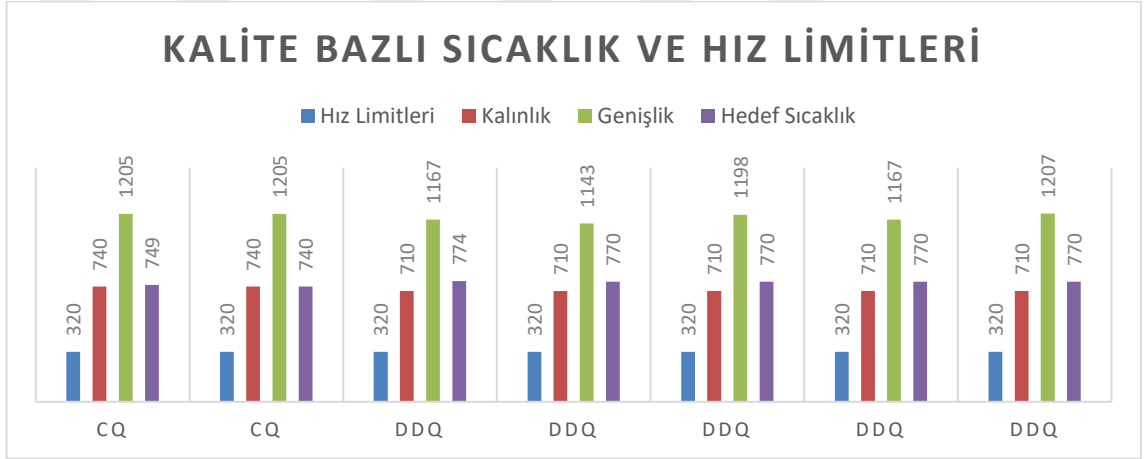


Şekil 40. 5. Deneme Kalınlık, Genişlik, Hıza Bağlı En Uygun Hedef Sıcaklıkları

Tablo 30

6. Deneme için en uygun sıcaklık grafiği

6. Deneme					
Bobin Numarası	Hedef Hız	Kalite Kodu	Kalınlık	Genişlik	Hedef Sıcaklık
26228183000	320	CQ	740	1205	749
26227543000	320	CQ	740	1205	740
26229598000	320	DDQ	710	1167	774
26229725000	320	DDQ	710	1143	770
36221598000	320	DDQ	710	1198	770
26229600000	320	DDQ	710	1167	770
36221468000	320	DDQ	710	1207	770

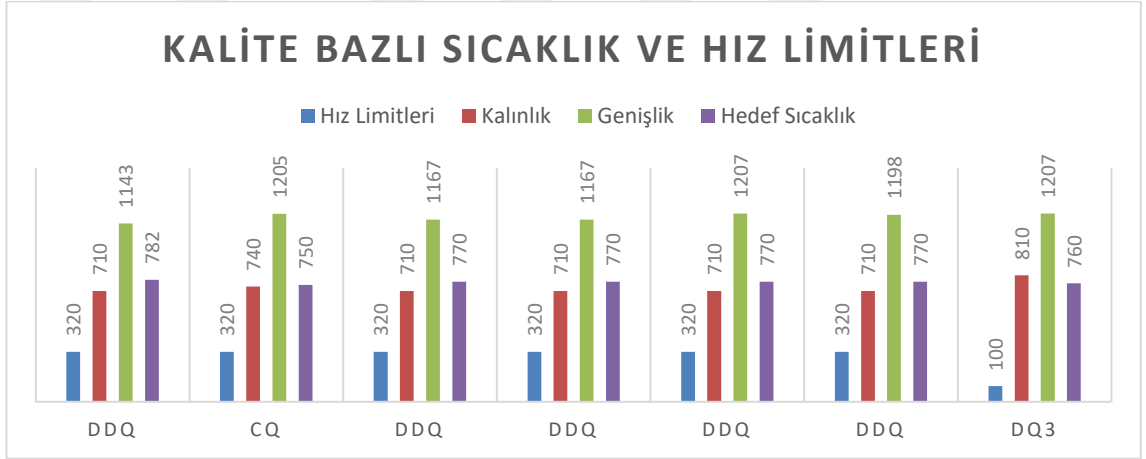


Şekil 41. 6. Deneme Kalınlık, Genişlik, Hıza Bağlı En Uygun Hedef Sıcaklıkları

Tablo 31

7. Deneme için en uygun sıcaklık grafiği

7. Deneme					
Bobin Numarası	Hedef Hız	Kalite Kodu	Kalınlık	Genişlik	Hedef Sıcaklık
26229725000	320	DDQ	710	1143	782
26228183000	320	CQ	740	1205	750
26229600000	320	DDQ	710	1167	770
26229598000	320	DDQ	710	1167	770
36221468000	320	DDQ	710	1207	770
36221598000	320	DDQ	710	1198	770
26229111000	100	DQ3	810	1207	760

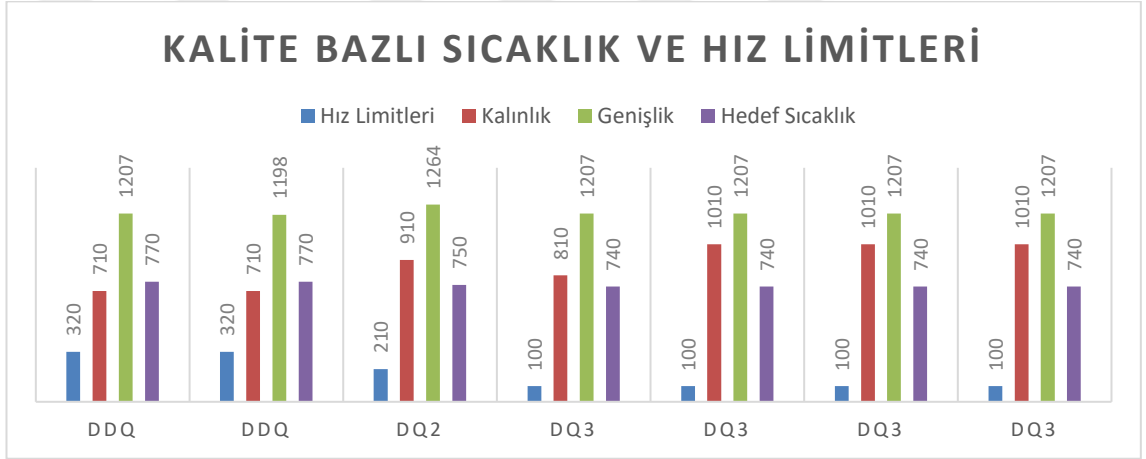


Şekil 42. 7. Deneme Kalınlık, Genişlik, Hıza Bağlı En Uygun Hedef Sıcaklıkları

Tablo 32

8. Deneme için en uygun sıcaklık grafiği

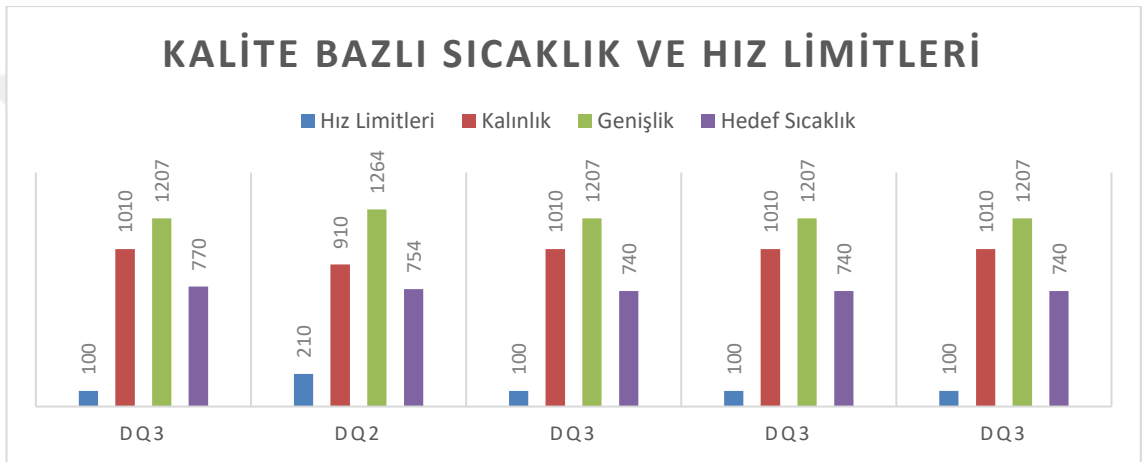
8. Deneme					
Bobin Numarası	Hedef Hız	Kalite Kodu	Kalınlık	Genişlik	Hedef Sıcaklık
36221468000	320	DDQ	710	1207	770
36221598000	320	DDQ	710	1198	770
36223950100	210	DQ2	910	1264	750
26229111000	100	DQ3	810	1207	740
36227244000	100	DQ3	1010	1207	740
26231321000	100	DQ3	1010	1207	740
36229511000	100	DQ3	1010	1207	740



Şekil 43. 8. Deneme Kalınlık, Genişlik, Hıza Bağlı En Uygun Hedef Sıcaklıkları

Tablo 33
9. Deneme için en uygun sıcaklık grafiği

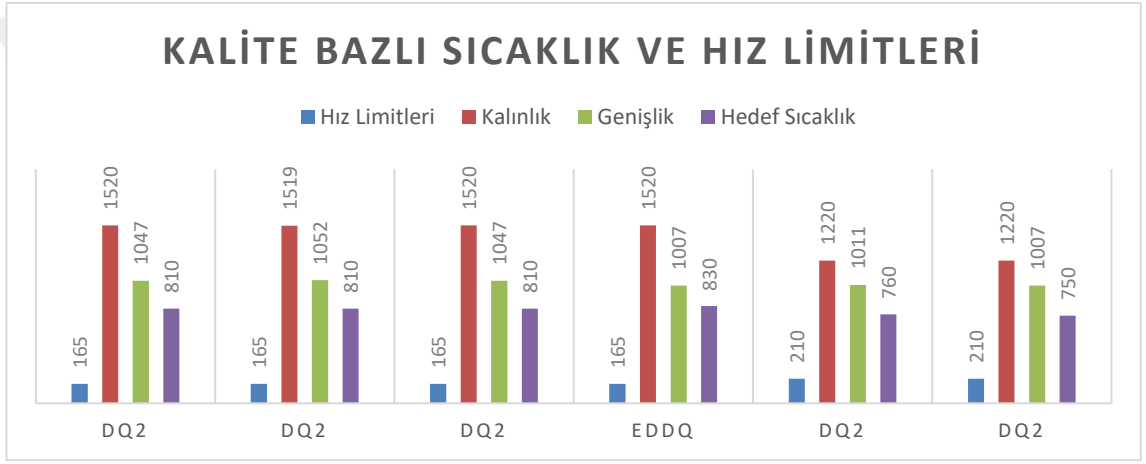
9. Deneme					
Bobin Numarası	Hedef Hız	Kalite Kodu	Kalınlık	Genişlik	Hedef Sıcaklık
26231321000	100	DQ3	1010	1207	770
36223950100	210	DQ2	910	1264	754
36227244000	100	DQ3	1010	1207	740
36229511000	100	DQ3	1010	1207	740
36229513000	100	DQ3	1010	1207	740



Şekil 44. 9. Deneme Kalınlık, Genişlik, Hıza Bağlı En Uygun Hedef Sıcaklıkları

Tablo 34
10. Deneme için en uygun sıcaklık grafiği

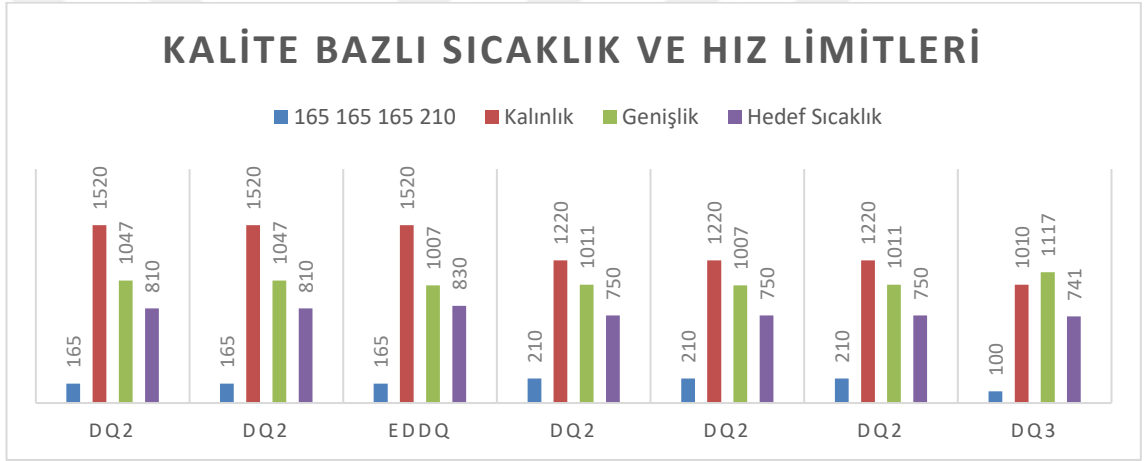
10. Deneme					
Bobin Numarası	Hedef Hız	Kalite Kodu	Kalınlık	Genişlik	Hedef Sıcaklık
26226701000	165	DQ2	1520	1047	810
26220585000	165	DQ2	1519	1052	810
26229970000	165	DQ2	1520	1047	810
26230259000	165	EDDQ	1520	1007	830
26226473000	210	DQ2	1220	1011	760
36226708000	210	DQ2	1220	1007	750



Şekil 45. 10. Deneme Kalınlık, Genişlik, Hıza Bağlı En Uygun Hedef Sıcaklıkları

Tablo 35
11. Deneme için en uygun sıcaklık grafiği

11. Deneme					
Bobin Numarası	Hedef Hız	Kalite Kodu	Kalınlık	Genişlik	Hedef Sıcaklık
26226701000	165	DQ2	1520	1047	810
26229970000	165	DQ2	1520	1047	810
26230259000	165	EDDQ	1520	1007	830
26226473000	210	DQ2	1220	1011	750
36226708000	210	DQ2	1220	1007	750
36226474000	210	DQ2	1220	1011	750
26229635000	100	DQ3	1010	1117	741



Şekil 46. 11. Deneme Kalınlık, Genişlik, Hıza Bağlı En Uygun Hedef Sıcaklıkları

İkinci problemimiz için 3 aylık çalışma sonucunda veri tabanında 411066 adet kayıt biriktirilmiştir (Şekil 47).

PHSCoNo	RTH1CoNo	RTH2CoNo	RTH3CoNo	PHSShipTempAct	RTH2ShipTemp2Act	RTH3ShipTemp3Act	PHSFurnaceTempAct	PHSFurnaceTempAct	RTH1FurnaceTemp1Act	RTH1FurnaceTemp2Act	RTH1FurnaceTemp3Act
33100476000	33100476000	33100476000	33100476000	168	783	781	290	152	728	767	780
33100476000	33100476000	33100476000	33100476000	167	783	781	290	152	728	768	780
33100476000	33100476000	33100476000	33100476000	168	783	781	290	152	728	768	780
33100476000	33100476000	33100476000	33100476000	168	783	781	290	152	728	767	780
33100476000	33100476000	33100476000	33100476000	168	783	781	290	152	728	767	780
33100476000	33100476000	33100476000	33100476000	167	783	781	290	152	729	767	780
33100476000	33100476000	33100476000	33100476000	167	783	781	290	152	728	767	779
33100476000	33100476000	33100476000	33100476000	167	783	781	290	152	729	767	779
33100476000	33100476000	33100476000	33100476000	168	783	781	290	152	729	767	779
33100476000	33100476000	33100476000	33100476000	167	783	781	290	152	729	767	779
33100476000	33100476000	33100476000	33100476000	168	784	781	290	152	729	767	779
33100476000	33100476000	33100476000	33100476000	168	783	781	290	152	729	768	779
33100476000	33100476000	33100476000	33100476000	167	784	781	290	152	729	768	779
33100476000	33100476000	33100476000	33100476000	168	784	781	291	152	729	768	779
33100476000	33100476000	33100476000	33100476000	167	785	781	291	151	729	767	779
33100476000	33100476000	33100476000	33100476000	168	785	781	291	151	728	767	779

Şekil 47. Veri tabanı kayıt sayısı

Bu kayıtların incelenmesi sonucunda 8965 bobine tavlama işlemi uygulandığı ve bu uygulama esnasında 981 adet kalite geçişi yaşandığı tespit edilmiştir. İlk iş olarak değerlerin normalize işlemi için program yazılmıştır. Hız değerleri maksimum hız değeri olan 320 m/dk. ile genişlik değerleri maksimum genişlik değeri olan 1525 cm ile, kalınlık değerleri maksimum kalınlık değeri olan 2 mm ile, sıcaklık değerleri maksimum sıcaklık değeri olan 950 ile değerlerinin maksimum değerlerine göre veri tabanındaki değerleri 0 - 1 aralığına getirecek veri değişiklikleri excel ortamında gerçekleştirilmiştir. Uzunluk boyutunda normalize programı EK-6'de sunulmuştur.

Bu arada kalite değerleri de sayısal forma çevrilmiştir. Bu amaçla;

CQ yerine 1 değeri;

DQ1 yerine 2 değeri;

DQ2 yerine 3 değeri;

DDQ yerine 4 değeri;

EDDQ yerine ise 5 değeri konulmuş ve buna göre normalize rakamlara dönüştürülmüştür.

3. aşamada tez konumuzda önemli olan geçişler sırasındaki operatör hareketlerini modellemek olduğu için önde giden ve arkadan gelen bobinlerin baştan ve sondan %50'lik kısımlarını değerlendirmemiz gerekmektedir. Bu nedenle de bobinlerin sadece %50 lik kısımlarını ele alacak ve geriye kalan kısımları silecek program geliştirilmiştir. Bu sırada aşağıda sözde kodu verilen algoritma kullanılmıştır. Bu sözde kodun programı EK-7'de verilmiştir.

Uzunluklarının baştan veya sondan %50'lik kısmını alan sözde kod

initialize failures to zero

assign static values to variables

connect to database for taking data

take data to array

do while end of data array

if data is between %50 and %100 of length

write to new excel sheet

end if

end while

4. aşamada peş peşe çekilen aynı kalite koduna sahip bobinlerin ayrılması sağlanmıştır. Bu amaçla aşağıda sözde kodu verilen algoritma kullanılmıştır.

Pespeşe devam eden aynı tip bobinleri silen sözde kodu

```
initialize failures to zero
assign static values to variables
connect to excel for taking data
take data to array
do while end of data array
    if width value and next value data is same and
    length value and next length d value is same and
    anneal code value and next anneal code value is same and
    else
        do while coil number not equal next coil number
            increase loop counter
        end
    end if
    increase loop counter
end while
```

Yazılan kod EK-8’de verilmiştir. Bu aşamadan sonra toplam veri satırı sayımız 59310’a düşürülmüştür (Şekil 48).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
59287	3,31E+10	0,448	0,319	0,422	0	0,898	0,843	0,888	0,882	0,917	0,892	0,921	0,923	0,919	0,87
59288	3,31E+10	0,448	0,319	0,422	0	0,914	0,843	0,887	0,882	0,916	0,892	0,921	0,923	0,918	0,871
59289	3,31E+10	0,472	0,319	0,422	0	0,931	0,842	0,886	0,881	0,916	0,891	0,92	0,922	0,918	0,871
59290	3,31E+10	0,483	0,319	0,422	0	0,948	0,842	0,886	0,881	0,914	0,891	0,92	0,922	0,918	0,87
59291	3,31E+10	0,483	0,319	0,422	0	0,966	0,841	0,884	0,88	0,913	0,891	0,92	0,922	0,918	0,87
59292	3,31E+10	0,483	0,319	0,422	0	0,983	0,84	0,883	0,879	0,913	0,891	0,92	0,921	0,917	0,87
59293	3,31E+10	1	0,131	0,422	0	0,87	0,849	0,897	0,878	0,91	0,89	0,918	0,917	0,91	0,838
59294	3,31E+10	1	0,131	0,422	0	0,891	0,849	0,897	0,878	0,91	0,89	0,917	0,917	0,91	0,838
59295	3,31E+10	1	0,131	0,422	0	0,913	0,848	0,897	0,878	0,91	0,889	0,918	0,917	0,91	0,838
59296	3,31E+10	1	0,131	0,422	0	0,935	0,848	0,897	0,877	0,91	0,889	0,917	0,917	0,91	0,838
59297	3,31E+10	1	0,131	0,422	0	0,957	0,847	0,896	0,878	0,909	0,889	0,917	0,917	0,91	0,837
59298	3,31E+10	1	0,131	0,422	0	0,978	0,847	0,896	0,877	0,909	0,889	0,917	0,916	0,91	0,837
59299	6,37E+10	0,448	0,45	0,567	0	0,87	0,861	0,894	0,894	0,912	0,909	0,927	0,931	0,929	0,857
59300	6,37E+10	0,448	0,45	0,567	0	0,913	0,86	0,894	0,893	0,912	0,908	0,927	0,931	0,928	0,857
59301	6,37E+10	0,448	0,45	0,567	0	0,957	0,86	0,893	0,893	0,911	0,908	0,926	0,931	0,928	0,857
59302	6,37E+10	0,414	0,45	0,446	0	0,862	0,808	0,853	0,853	0,871	0,852	0,892	0,894	0,889	0,817
59303	6,37E+10	0,414	0,45	0,446	0	0,897	0,808	0,853	0,853	0,871	0,852	0,892	0,894	0,888	0,817
59304	6,37E+10	0,414	0,45	0,446	0	0,931	0,808	0,853	0,853	0,871	0,852	0,891	0,894	0,888	0,817
59305	6,37E+10	0,414	0,45	0,446	0	0,966	0,809	0,854	0,853	0,871	0,852	0,891	0,894	0,888	0,817
59306	6,37E+10	0,366	0,925	0,754	0	0,882	0,864	0,886	0,9	0,929	0,921	0,948	0,948	0,94	0,836
59307	6,37E+10	0,366	0,925	0,754	0	0,941	0,864	0,886	0,9	0,929	0,921	0,948	0,948	0,94	0,834
59308	6,37E+10	0,366	0,925	0,754	0	1	0,864	0,884	0,899	0,929	0,921	0,948	0,947	0,94	0,834
59309	6,37E+10	0,366	0,925	0,754	0	0,889	0,874	0,896	0,907	0,934	0,921	0,947	0,944	0,938	0,828
59310	6,37E+10	0,366	0,925	0,754	0	0,944	0,873	0,896	0,906	0,934	0,921	0,947	0,944	0,938	0,828
59311															

Şekil 48. Düzenlemelerden sonra kalan veri satırı sayısı

3.5 Analiz

Yukarıdaki grafiklerden görüleceği üzere en yüksek hız performansını bulmada Microsoft firmasının Solver Foundation 3.0 çerçevesinin içinde tanımlandığı “HybridLocalSearchDirective” en iyileme algoritması kullanılmıştır. Bu algoritma hibrit yerel araştırma çözücü stokastik tepe tırmanma algoritmasını kullanmaktadır. Bu yöntem daha önceki bölümlerde anlatılmaktadır. Rasgele bir karar değerinden başlamakta ve tekrarlı olarak ufak değişiklikler ile en uygun değere yakınlasmaya çalışmaktadır. İyi sonuca giden değerleri saklamakta, kötü sonuca doğru giden değerleri ise saklamamaktadır. Bu sonuca ulaşmak için tablolarda belirtilen maksimum hız değerleri kullanılmaktadır. Uygun kalınlık ve genişliğe göre maksimum hız değerleri tablolarda belirlenmiştir. Program en uygun sıcaklık set değerleri arasında en uygununu bulmaya çalışmaktadır. Bu sayede enerji verimliliği de sağlanmış olacaktır. Enerji verimliliği sağlamanın bir yolu da sıcaklık geçişleri arasındaki farkların az olmasında yatmaktadır.

Matlab yazılımı kullanılarak yapay sinir ağıları ile mevcut veriler üzerinde birçok deneyler yapılmıştır. Örnek matlab programı Ek-9’de verilmiştir. Bu deneylerden bazıları Tablo 36’da gösterilmiştir.

Tablo 36
YSA Denemeler Tablosu

<u>Uygulanan Transfer Fonksiyonları ve nod sayısı</u>	<u>MSE</u>
[453] – tansig - trainlm	2,39E-04
[453] – tansig – trainscg	2,05E-04
[504] – tansig – trainlm	2,18E-04
[504] – tansig – trainscg	3,05E-04
[504] – tansig – trainrp	3,63 E-04
[570] – tansig – trainscg	2,12E-04
[650] – tansig – trainscg	2,17E-04
[760] – tansig – trainscg	2,72E-04
[910] – tansig – trainscg	2,29E-04
[1150] – tansig – trainscg	2,44E-04
[1512] – tansig – trainscg	2,79E-04
[2269] – tansig – trainscg	3,25E-04

Aşağıda en uygun mse değerini veren [453] – tansig – trainscg yapay sinir ağının sonuçları aşağıda verilmiştir. Yukarıdaki tabloda uygulanan yapay sinir ağları ve uygulama sonuçlarından bazılarının sonuçlarının çok fazla olması nedeniyle gerçekleşme verilerinden en iyi olanların bazıları Ek-10’de gösterilmektedir.

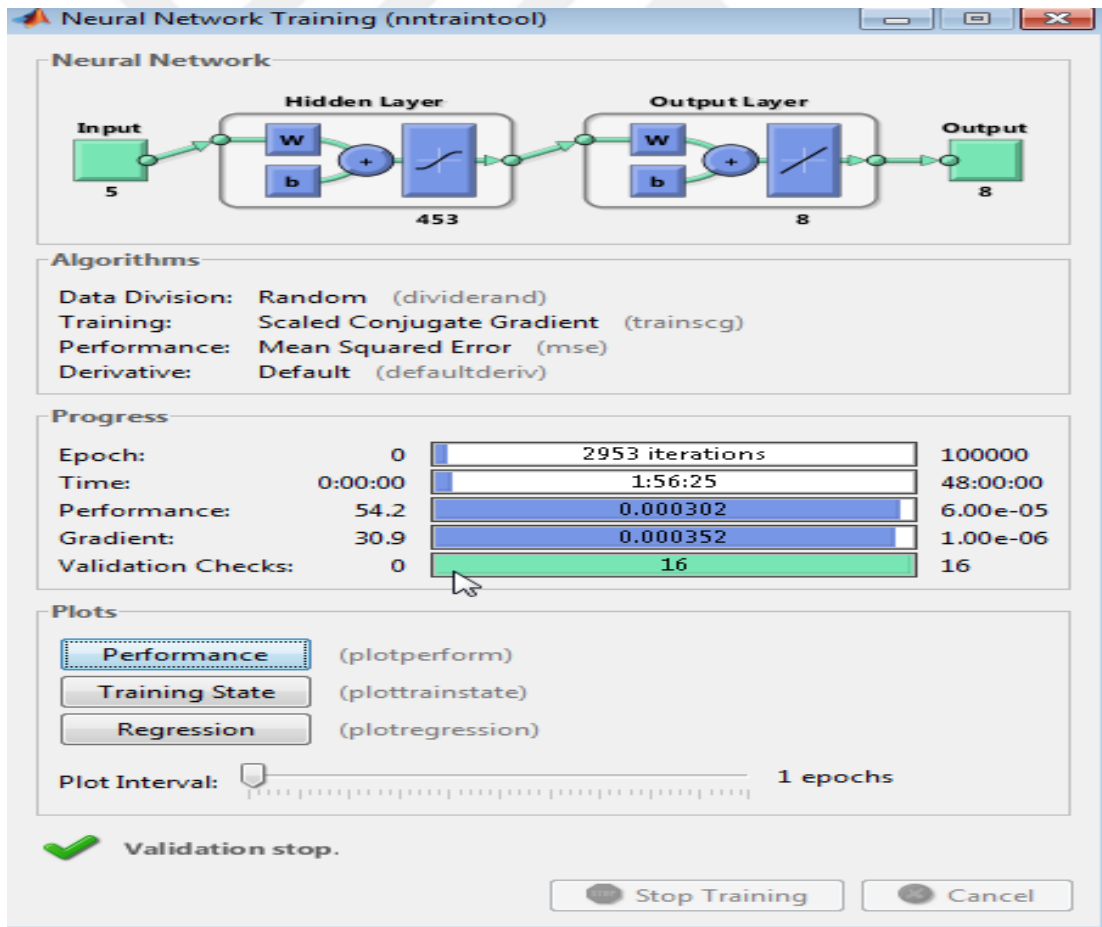
Uygulanan matlab program aşağıdaki gibidir. Nöron hesabı formülüne göre $\alpha = 10$ seçilerek nöron sayısı 453 bulunmuştur. Bir üstteki program ile aynı yapıdadır. Sadece nöron sayısı farklıdır.

Performans değerimiz ise;

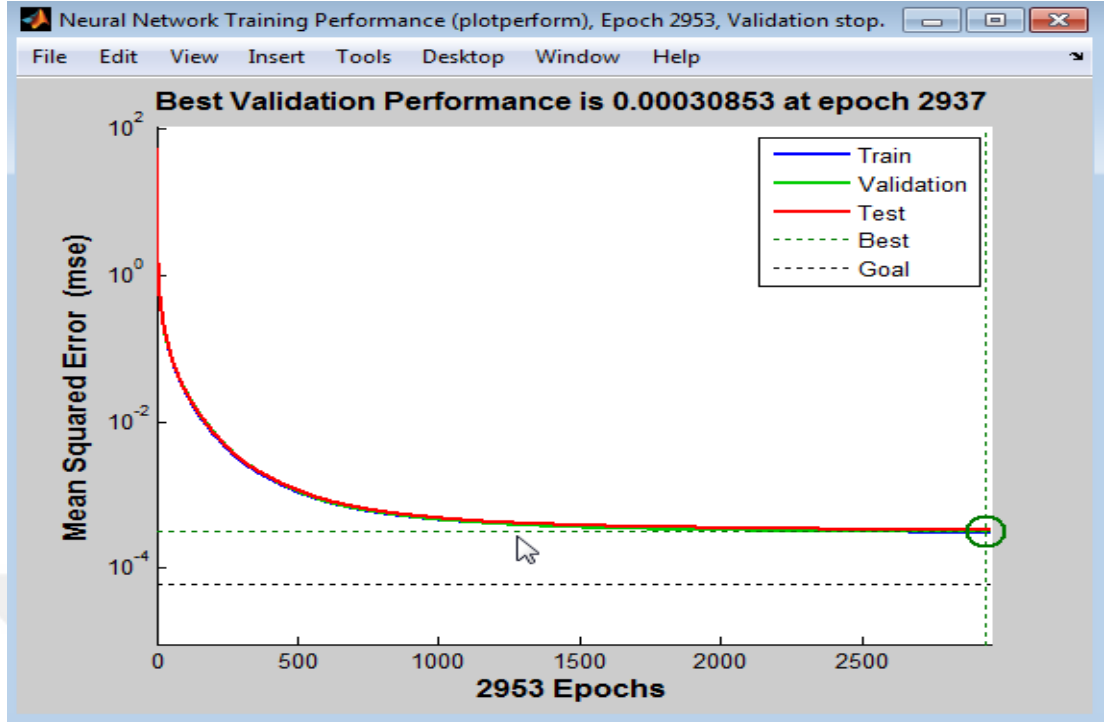
$$\text{Performans (mse)} = 2,05 \cdot 10^{-4}$$

olarak bulunmuştur.

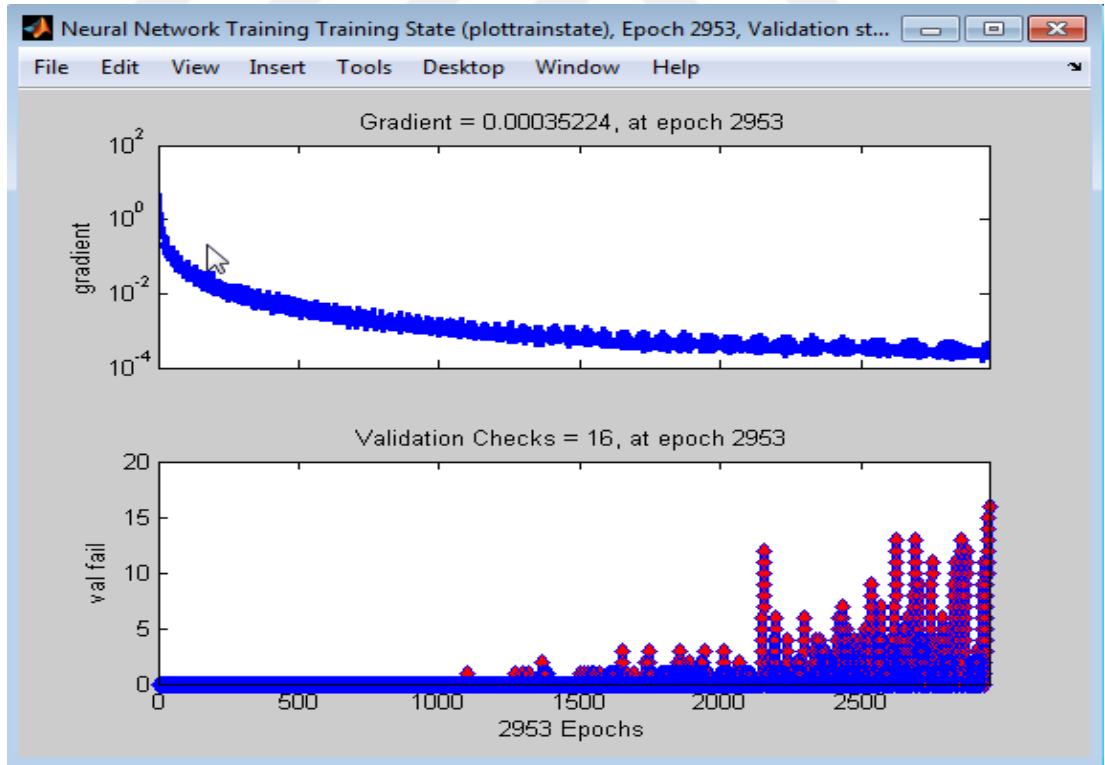
Matlab’den elde edilen sonuçlar Şekil 49, Şekil 50, Şekil 51 ve Şekil 52’de gösterilmiştir.



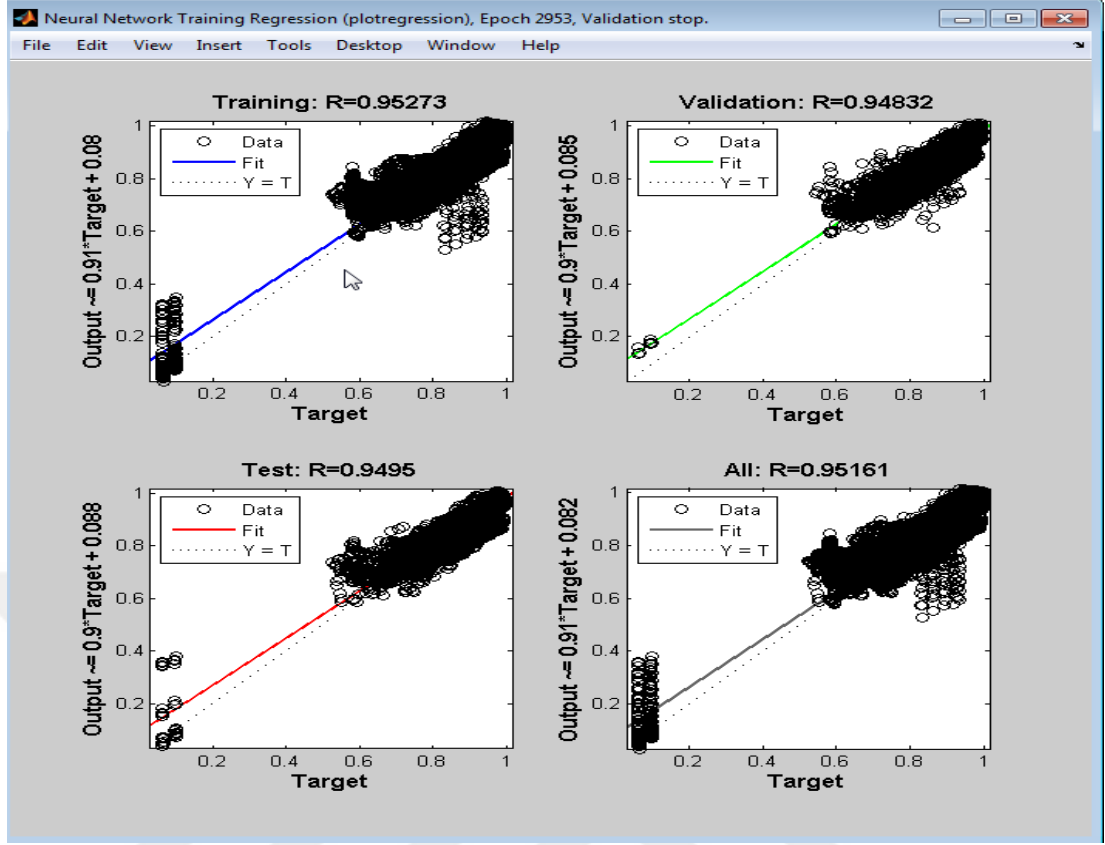
Şekil 49: 453-tansig-trainscg YSA için nntraintool Görünümü



Şekil 50: 453-tansig-trainscg YSA için En İyi Performans Değeri



Şekil 51: 453-tansig-trainscg YSA için Eğitim Durumu



Şekil 52: 453-tansig-trainscg YSA için Regresyon Değerleri

Tablo 37, Tablo 38 ve Tablo 39’da gerçek fırın sıcaklıkları, hesaplanan fırın sıcaklıkları ve arasındaki fark sıcaklıklar sunulmuştur.

Tablo 37
453-tansig-trainscg Gerçek Fırın Sıcaklıkları

[453] tansig –trainscg Gerçek Fırın Sıcaklıkları							
768,6	814,2	795,2	823,7	802,8	828,4	827,5	817,0
772,4	815,1	796,1	823,7	801,8	827,5	824,6	816,1
772,4	816,1	796,1	823,7	801,8	827,5	824,6	816,1
772,4	816,1	796,1	823,7	801,8	827,5	824,6	816,1
772,4	816,1	796,1	823,7	801,8	827,5	824,6	816,1
771,4	816,1	796,1	823,7	801,8	827,5	824,6	816,1
771,4	816,1	796,1	823,7	801,8	827,5	824,6	816,1
771,4	816,1	796,1	823,7	801,8	827,5	824,6	816,1
771,4	816,1	796,1	823,7	801,8	827,5	824,6	816,1
726,8	766,7	756,2	781,9	770,5	794,2	797,1	793,3
726,8	765,7	755,3	780,9	768,6	793,3	796,1	793,3
725,8	764,8	755,3	780,0	768,6	792,3	795,2	792,3
724,9	764,8	754,3	779,0	767,6	792,3	795,2	792,3
723,9	763,8	753,4	778,1	766,7	791,4	794,2	791,4
723,0	762,9	753,4	778,1	766,7	791,4	794,2	790,4
723,0	761,9	752,4	777,1	765,7	790,4	793,3	790,4
722,0	761,0	751,5	776,2	765,7	789,5	793,3	790,4
801,8	842,7	825,6	859,8	830,3	859,8	859,8	856,0
801,8	842,7	825,6	858,8	830,3	858,8	859,8	855,0
801,8	842,7	825,6	858,8	830,3	858,8	859,8	855,0
800,9	843,6	825,6	858,8	829,4	858,8	858,8	855,0
800,9	843,6	825,6	858,8	829,4	857,9	858,8	854,1
761,9	801,8	797,1	825,6	803,7	830,3	832,2	828,4
761,0	800,9	796,1	825,6	803,7	830,3	832,2	828,4
760,0	799,0	795,2	824,6	802,8	830,3	831,3	827,5
759,1	799,0	794,2	824,6	802,8	829,4	831,3	827,5
759,1	798,0	794,2	823,7	802,8	829,4	831,3	825,6
758,1	797,1	793,3	823,7	801,8	828,4	830,3	825,6
758,1	797,1	793,3	822,7	801,8	828,4	830,3	825,6
757,2	796,1	792,3	821,8	801,8	828,4	830,3	825,6
756,2	795,2	791,4	821,8	801,8	828,4	829,4	824,6
763,8	806,6	790,4	818,9	800,9	825,6	824,6	819,9
763,8	806,6	790,4	818,9	800,9	825,6	824,6	818,9
763,8	806,6	790,4	818,9	800,9	824,6	824,6	818,9
762,9	806,6	790,4	818,9	799,9	825,6	824,6	818,9
762,9	806,6	789,5	818,9	799,9	824,6	824,6	818,9
761,9	805,6	790,4	818,0	799,9	824,6	824,6	818,9

Tablo 38
453-tansig-trainscg Hesaplanan Fırın Sıcaklıkları

[453] Tansig – trainscg Hesaplanan Fırın Sıcaklıkları							
766,9	815,7	793,7	815,8	808,5	827,1	814,5	815,3
771,3	816,3	800,4	822,3	817,2	837,7	834,9	825,3
774,7	814,4	798,2	815,9	810,7	829,5	826,5	819,1
775,7	812,7	800,0	812,5	811,7	830,5	827,9	818,7
775,7	809,1	798,8	814,2	816,2	830,6	830,1	818,0
775,0	809,2	800,2	814,5	815,4	830,2	832,0	817,7
775,2	810,8	803,9	815,5	811,3	831,2	831,2	817,4
772,3	810,7	798,7	814,1	807,3	827,8	821,7	815,5
766,9	815,7	793,7	815,8	808,5	827,1	814,5	815,3
729,2	760,1	755,8	773,3	785,8	797,2	801,8	797,0
729,7	762,2	760,5	777,1	784,2	799,5	803,3	797,7
727,7	762,1	760,5	777,6	783,2	801,5	803,5	799,0
725,6	760,6	756,8	775,3	783,4	799,1	801,0	799,9
721,9	760,1	754,0	773,0	781,9	795,8	798,8	797,9
719,8	760,8	754,1	774,3	781,3	795,8	799,5	794,6
723,4	761,3	758,4	775,9	782,6	798,0	803,7	793,1
729,2	762,4	762,3	772,5	781,5	799,9	805,8	795,0
795,8	833,5	828,4	847,6	826,9	849,0	846,5	833,3
792,2	833,6	826,0	843,7	832,4	850,0	846,2	834,6
795,7	830,2	827,2	844,0	835,2	852,2	851,9	837,0
794,1	827,9	824,2	840,7	831,9	849,1	846,9	836,0
794,3	837,8	830,7	842,9	831,4	852,2	844,6	836,6
730,5	766,4	772,5	782,9	791,7	809,7	806,8	804,0
733,7	777,6	782,9	790,3	802,3	820,0	816,8	814,2
731,6	779,9	784,7	792,6	805,6	823,3	818,8	818,8
727,5	776,1	781,2	788,8	803,9	819,5	815,5	817,0
724,8	770,8	777,6	783,5	801,7	814,0	812,2	810,7
732,1	774,8	783,2	791,4	801,6	815,3	815,6	806,8
738,2	778,9	789,4	798,8	803,0	818,6	820,0	807,7
742,9	782,5	793,7	804,0	807,6	822,4	822,8	811,0
744,5	784,1	792,6	803,0	811,2	823,5	822,0	812,7
771,3	816,3	800,4	822,3	817,2	837,7	834,9	825,3
774,7	814,4	798,2	815,9	810,7	829,5	826,5	819,1
775,7	812,7	800,0	812,5	811,7	830,5	827,9	818,7
775,7	809,1	798,8	814,2	816,2	830,6	830,1	818,0
775,0	809,2	800,2	814,5	815,4	830,2	832,0	817,7
775,2	810,8	803,9	815,5	811,3	831,2	831,2	817,4

Tablo 39
453-tansig-trainscg Hesaplanan ile Gerçek Fırın Sıcaklıkları Arasındaki Fark

[453] Tansig – trainscg Fark = Hesaplanan - Gerçek							
1,7	-1,6	1,5	7,8	-5,7	1,3	12,9	1,7
1,0	-1,2	-4,3	1,3	-15,4	-10,2	-10,3	-9,3
-2,3	1,6	-2,1	7,7	-8,9	-2,1	-1,9	-3,1
-3,4	3,4	-3,9	11,2	-9,9	-3,1	-3,3	-2,6
-3,4	6,9	-2,7	9,4	-14,4	-3,2	-5,5	-1,9
-3,6	6,8	-4,1	9,1	-13,6	-2,7	-7,4	-1,6
-3,8	5,3	-7,8	8,2	-9,5	-3,8	-6,6	-1,3
-0,9	5,4	-2,6	9,6	-5,5	-0,4	2,9	0,5
4,5	0,3	2,4	7,8	-6,7	0,4	10,1	0,8
-2,5	6,5	0,4	8,6	-15,4	-3,0	-4,8	-3,7
-2,9	3,5	-5,3	3,8	-15,6	-6,3	-7,2	-4,4
-1,9	2,7	-5,2	2,4	-14,7	-9,2	-8,4	-6,7
-0,7	4,1	-2,5	3,7	-15,8	-6,8	-5,9	-7,6
2,0	3,7	-0,6	5,1	-15,3	-4,4	-4,6	-6,6
3,1	2,0	-0,8	3,8	-14,6	-4,4	-5,3	-4,2
-0,5	0,6	-6,0	1,2	-16,9	-7,6	-10,5	-2,7
-7,2	-1,5	-10,9	3,6	-15,8	-10,4	-12,5	-4,6
6,0	9,2	-2,9	12,1	3,4	10,7	13,3	22,7
9,6	9,1	-0,4	15,1	-2,1	8,8	13,6	20,4
6,1	12,5	-1,7	14,8	-4,9	6,6	7,8	18,0
6,8	15,7	1,4	18,1	-2,5	9,7	11,9	19,0
6,6	5,8	-5,1	15,9	-2,0	5,6	14,2	17,5
31,4	35,4	24,6	42,6	12,0	20,6	25,4	24,4
27,2	23,2	13,2	35,3	1,4	10,3	15,4	14,2
28,4	19,1	10,4	32,0	-2,8	7,0	12,5	8,7
31,6	22,9	13,0	35,8	-1,1	9,8	15,8	10,4
34,3	27,2	16,6	40,2	1,1	15,4	19,0	14,9
26,0	22,3	10,1	32,3	0,2	13,1	14,7	18,7
19,9	18,2	3,9	23,9	-1,2	9,8	10,3	17,8
14,3	13,6	-1,4	17,8	-5,8	6,0	7,5	14,6
11,7	11,1	-1,3	18,8	-9,4	4,9	7,4	11,9
-7,5	-9,7	-10,0	-3,4	-16,4	-12,1	-10,3	-5,5
-10,9	-7,9	-7,8	3,0	-9,8	-4,0	-1,9	-0,2
-11,9	-6,1	-9,6	6,4	-10,9	-5,9	-3,3	0,2
-12,9	-2,6	-8,4	4,7	-16,3	-5,1	-5,5	0,9
-12,1	-2,7	-10,7	4,4	-15,5	-5,6	-7,4	1,2
-13,3	-5,2	-13,5	2,5	-11,4	-6,6	-6,6	1,5

SONUÇ VE ÖNERİLER

Değerlendirme

Çalışma iki kısımdan oluşmuştur. Birinci kısımda hatta giren bobinlerin uygun sırada ve sıcaklıklarda tavlınması ve ikinci kısımda ise kullanılamayan fırın modellerinin yerine kullanılacak bir sistem geliştirilmesi sağlanmıştır.

Birinci kısımda ele alınan konu bir optimizasyon problemi idi. Yaklaşık giriş stok sahasında bulunan bobinler arasında en uygun sıcaklık ve boyut geçişlerini sağlayacak üretim sırası oluşturacak algoritma geliştirmeye çalışılmıştır. Bu çalışmada amaçlar ve kısıtlar belirlenmiş ve Microsoft C# yazılımını kullanarak program geliştirilmiştir.

Yapılan denemeler sistemden alınan gerçek değerler ile yapılmıştır. Hız değerleri daha önce bulunmuş kalite ve genişlik temelli hız değerlerine sabitlenmiştir. 1.deneme’de DQ1’den CQ’ya inecek sıcaklık değeri en az değer olan 690 °C kadar soğutulmaktadır. 2. deneme tek bir kalite için yapılmış ve en düşük sıcaklıkta şerit tutulmaya çalışılmıştır. 3.deneme’de sistem tek bir bobin için sıcaklık değerini yükseltmek zorunda kalmış ve tekrar en düşük sıcaklık olan 690 °C değerine sistemi taşımıştır. 4.deneme’de kalite DQ1’den CQ değerine inmiş daha sonra tekrar DQ1’e geri dönmüştür. Burada da en uygun sıcaklık değeri olarak DQ1’in olması gereken en düşük sıcaklık değeri CQ için olması gereken en yüksek sıcaklık değeri bulunmuştur. 5. Deneme de ise DQ1’den CQ’ya geçiş vardır ve CQ maksimum sıcaklıkta fırın set değeri elde edilmiştir. 6.denemede fırın sıcaklıkları 740 °C’den 770 °C’ye çıkarılmıştır. Maksimum hız uygulanmıştır. Geri kalan sıcaklık ve hız değerlerine baktığımızda en uygun hız ve sıcaklık değerlerinin yakalandığı görülmüştür.

İkinci kısımda ise kullanılamayan fırın modeli yerine operatör hareketlerini modelleyen bir sistem geliştirilmiştir. Fırın modellerinin kullanılamama nedeni üretim planlamanın gönderdiği üretim sisteminin fırın üreticisinin önerdiği sisteme uygun olmamasıdır. Bu nedenle işletme kendi pratiklerini oluşturmuştur. Bu pratiklere göre satış değeri düşük olan ürünün son kısmı tolerans harici yapılarak, ara bobin kullanılmadan ve üretim kaybına neden olmadan üretim sağlamaktır. Bu sistemi geliştirirken yapay sinir ağlarından yararlanılmıştır. Yapay sinir ağı kullanılma nedeni matematiksel olarak modellenemeyen

sistemleri oluşturmada başarılı olmasıdır ve daha önce görmediği örneklerle karşıda sonuç üretebilmesidir.

Yapay sinir ağı oluşturmak için mathworks oluşumu tarafından yapılan performans testleri sonucunda en iyi sonuçları veren üç algoritma kullanılmıştır. Bu algortimalar trainlm - Levenberg-Marquardt, trainrp - Resilient Backpropagation ve trainscg - Scaled Conjugate Gradient algortimalarıdır. Ara katman nöron sayısı olarak da stackexchange web sayfasından bulunan formül kullanılmıştır. Üç algortma ile ayrı ayrı 9 deneme yapılmıştır. Bu denemeler sonucu en iyi sonuç teze yansıtılmıştır. Diğer sonuçlardan bazıları ise ekler bölümünde sunulmuştur.

YSA $2,05 \cdot 10^{-4}$ hata değerine ulaşmasına rağmen hesaplanan sıcaklık değerleri ile gerçek sıcaklık değerleri arasında $[-23,34,5]$ kadar fark olabilmektedir. Ağın eğitilmesi için çok zaman alan işlemdir. Ağın eğitim performansı % 95,27 , doğrulama performansı % 94,83 ve test performansı ise % 94,95 olarak bulunmuştur. Bütün performans oranı ise % 95,16'dır. Bu performans oranını %99 çıkartmak için uygun bir öğrenme algoritması bulunana kadar çalışmalar devam edecektir.

Bulgular

Birinci çalışmada hem verimliliğe hem de üretim artışına neden olacak bir çözüm önerilmiştir. Denemelerde görülmüştür ki en uygun sıcaklıkları ve hızları veren program uygun bir şekilde çalışmıştır. Operatörlerin hesabını yapamayacağı ve müdahalesine gerek olmadan en uygun sıcaklıklara fırının getirilmesi sağlanmıştır. Bu sayede hem ebat ve kaliteye bağlı olarak maksimum hızda bobinlerin tavlanması sağlanmış hem de aynı zamanda enerji tasarrufu da yapılmıştır. Bu çalışmada maliyet açısından bir değer vermek şu an için mümkün değildir. Teorik uygulama gerçeğe dönüştüğünde bu olayı daha iyi yorumlama fırsatımız olacaktır.

İkinci çalışmada hattın matematiksel modelinin yerine boyut ve kalite geçişleri için operatör hareketlerini referans alan bir model geliştirilmiştir. Üretim şartları altında boyut ve kalite geçişleri çok fazla miktarda olduğu için ekonomik olarak fazla kayıplara neden olmaktadır. Üretim kaybının yanı sıra aynı zamanda enerji kaybının da bulunduğu yapılan gözlemlerden anlaşılmıştır. Bu konuda benzer hatta yapılan raporlama çalışması sırasında görülmüştür ki bu kayıp yıllık bazda **%2'leri** bulmaktadır. Bu çalışmanın ilk 10 aylık

sonuçları EK-11’de gösterilmiştir. Benzer tavlama hattı çok ince bobinlerin çekildiği ve geçişlerin STH hattına göre daha az olduğu bir hattır. Bu rağmen 498 bobin yılbaşından bu yana ara bobin olarak çekilmiştir. Bu bobinlerin ağırlığı toplam 2305 ton dur. Toplam kapasite 240 000 ton olduğuna göre ara bobinlerin toplam üretime oranı % 0,96 olarak bulunur.

Çalışma yapılan hatta 2016 yılında 936,3 ton malzeme tav fırın sıcaklık hatalı kusurundan tolerans harici olmuştur. Bu miktarın toplam kapasiteye oranı % 0,10 dur. Yapılan bu çalışma ile kalite ve boyut geçişlerinin operatör uygulamalarının standartlaştırılması sağlanmıştır. Bu sayede bu tolerans harici malzemenin azalacağı düşünülmektedir.

Öneri

Veri sayısı çok fazla olursa ağın öğrenme süresi çok uzamaktadır. Bu nedenle yapılabilirse veri alt bölümlere ayrılmalı ve bölüm bölüm öğrenme sağlanmalıdır. Veri sayısının fazla olması ayrıca daha kuvvetli ve hızlı bilgisayar sistemlerinin de kullanılmasını gerektirmektedir. Bu sistemlerde pahalı sistemler oldukları için özellikle küçük işletmeler için yapılan projelerde uygun görülmeyebilir.

YSA çalışmalarında veri doğruluğunu sağlamak amacıyla istatistiksel veri analizlerinin de yapılması (Regresyon Analizi) yapılan işlemlerin doğruluğunu ve hatalı ve etkisiz verilerin ayıklanması sağlanabilir.

Ayrıca bu çalışma sadece ısıtma fırınları için yapılmıştır. Soğutma tarafına da böyle bir sistemin kurulması kalite ve hız açısından önem arz etmektedir. Sadece ısıtma fırınlarında yapılan bu çalışma tek başına bir ifade etmeyecektir. Hattın hem ısıtma hem de soğutma tarafına yayılması gerekmektedir.

Günden güne yeni öğrenme ve optimizasyon algoritmaları çıkmaktadır. Geleneksel deterministik algoritmaların yanında sezgisel algoritmaların da kullanılması öğrenme performansının iyileştirilmesi bakımından önem arz etmektedir.

Tezin Kısıtlılıkları

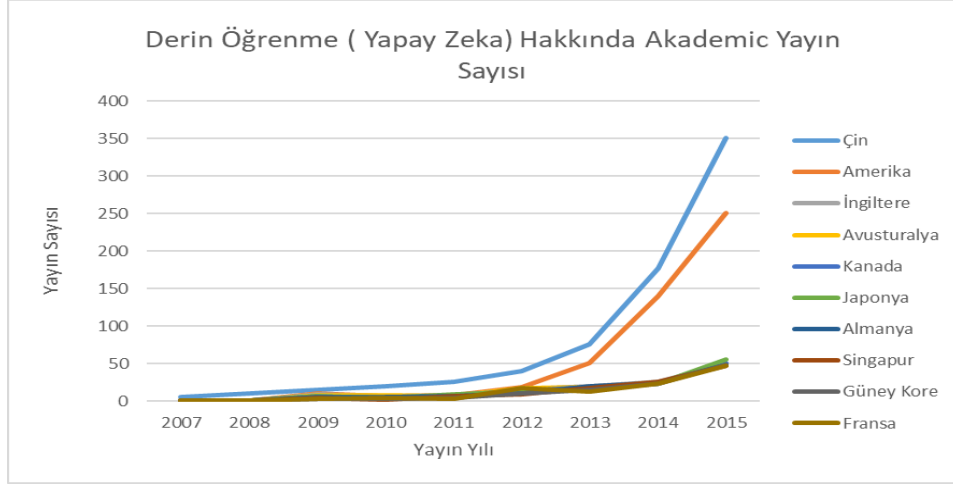
Bu çalışma toplanan 3 aylık örneklerle yapılmıştır. Bütün ürün gamına çalışmamızı yaymak için veri toplama ve bu verileri eğitim aşamasında kullanma çalışmalarımız devam etmektedir. Yapay sinir ağını eğitim ve gerçek üretimde kullanma aşamaları aynı zamanda olacaktır. Gelen veri yeni bir veri ise operatör uyarılacak ve kontrol operatörün olacaktır. Fakat eski veri ise ve daha önce eğitimde kullanıldı ise ağ set değeri üretmeye devam edecektir.

Yapay Sinir Ağları geleneksel programlama göre bir çok avantaj ve dezavantajları vardır. Bilgi ağına tamamına yayıldığı için birkaç bilginin kaybolması veya eksikliği ağına çalışmasını engellemez. Makine öğrenmesi yapar. Verileri verdikten sonra benzer veriler karşısında aynı özellikleri gösterebilme yetenekleri vardır. Eksik bilgi ile çalışabilir. Ağına öğrenme süreci sonlanmamakta yeni veriler eğitime devam edilebilmektedir. Bu avantajlarının yanı sıra dezavantajları da vardır. Dezavantajları ise ağına davranışlarının açıklanamamasıdır. Uygun ağ yapısının bulunması ise en önemli sorunlarından biridir. Ağ topolojini bulmak için çok fazla sayıda deneme yapmak zorunda kalınmaktadır. Aynı zamanda ağ eğitim süresinin bilinmemesi de ayrı bir problemdir. Verdiğiniz hata sınırın altına bile indiğinde test aşamasında hatalı sonuçlar verebilmektedir.

Hata oranlarının yanı sıra ezberleme yapma durumları da oluşmaktadır. Bu durumda tekrar ağı öğretmek gerekebilir. Eğer veri çok fazla ise hafıza sorunu ortaya çıkabilir ve öğrenme süresinin çok uzun olmasına neden olabilir.

Tavsiyeler

Yapay sinir ağları sürekli gelişen bir yapıdadır. Bu alanda yapılan çalışmalar devam etmekte ve akademik yayınlar yıldan yıla artmaktadır (Şekil 53). Bu nedenle bu alanda yapılan çalışmalar sürekli olarak takip edilmeli ve bu çalışmalar doğrultusunda yapılan uygulamaların iyileştirilmesi sağlanmalıdır.

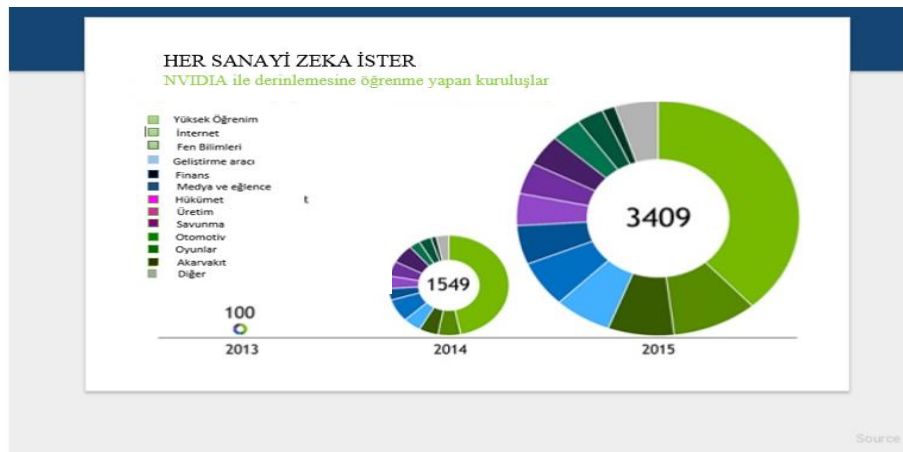


Şekil 53: Akademik Yayınların Gelişimi

Kaynak: Slideshare, 2017, <https://www.slideshare.net/LuMa921/deep-learning-the-past-present-and-future-of-artificial-intelligence>

Bu arada akademisyenlerin yanısıra şirketlerin de zeki sistemleri ile ilgileri de gitgide artmaktadır (Şekil 54). Zeki sistemlerin iş hayatına girmesi ve üretim hatlarında kullanılması firmaların iş yapış biçimlerini de değiştirecektir. İnsana bağlı yapıdan makineye bağlı yapıya geçiş sağlanacak, bu sayede de kaliteli ve ucuz iş ortaya çıkacaktır.

Zeki sistemler alt yapısını kuran firmalar Endüstri 4.0 uygulamasına kolaylıkla geçebileceklerdir. Bu sayede de robotik iş gücü yaygınlaşacaktır. İnsan bağımlılığından kurtarılacak hata yapma olasılıkları en az düzeye indirilmiş olacaktır.



Şekil 54. Organizasyonların Zeki Sistemlere olan İlgileri

Kaynak: Slideshare, 2017, <https://www.slideshare.net/LuMa921/deep-learning-the-past-present-and-future-of-artificial-intelligence>

Günümüz enerji çağıdır. Enerjiyi etkin kullanan şirketler bir adım önde rekabet avantajı sağlamış olurlar. Bu nedenle birçok şirket üretkenliğini arttıracak araştırmalar yapmakta ve ödüller vermektedir. Avantaj sağlayacak unsurlardan biri enerjiyi etkin kullanmak, diğeri ise üretim miktarını arttırmaktır. Bu nedenle firmalar deneysel tasarımlar (DOE - Design of Experiment) yapmak için ekipler toplamakta enerji tasarrufu, çevresel gelişimler ve üretim arttırıcı çalışmalar yapmaktadır. Hükümetler AR-GE çalışmaları için teşvikler vermekte ve hatta finansman desteği sağlamaktadırlar. Bu teşviklerin nedeni sürdürülebilir enerji kaynaklarının gün geçtikçe azalması ve enerji gereksiniminin artmasıdır. Bundan sonra yapılacak çalışmalarda enerji tasarrufuna önem verilmelidir. Enerji tasarrufu aynı zamanda şirketleri maliyetlerin azaltılması konusunda diğer firmalardan bir adım öne geçirecektir.

KAYNAKÇA

Kitaplar

- ANDERSON D.R. , SWEENEY D.J. , WILLIAMS T.A. , WISNIEWSKI M. (2009), “An Introduction to Management Science”, C & C Offset, Çin
- ARORA J.S. (2004), “Intoduction to Optimum Design Second Edition”, Elsevier Academic Press
- AYANOĞLU M. (2006), “Yönetim Bilimi(Yöneylem Araştırması)”, Sakarya Kitapevi, Adapazarı
- HANS J. ZIMMERMANN (1991), “Fuzzy Set Theory and its Application”, Kluwer Academic Publisher, Boston
- HILLIER F.S., LIEBERMAN G.J., Introduction to Mathematical Programming, McGraw-Hill , Inc.U.S.A., 1990
- KARABOĞA D. (2011), “Yapay Zeka Optimizasyon Algoritmaları”, Nobel Kitap Dağıtım, İstanbul
- MALCZEWSKI J. (1999), “GIS and Multicriteria Decision Analysis”, John Wiley And Sons, Newyork
- ÖZKAN MUSTAFA M. (2003), Bulanık Hedef Programlama, Ekin Kitabevi, Kastamonu
- ÖZTEMEL, E. (2012), “Yapay Sinir Ağları”, Papatya Yayıncılık Eğitim, İstanbul
- RAGSDALE, C.T. (2008), “Spreadsheet Modeling & Decision Analysis: A Practical Introduction to Management Science”, ICC Macmillan Inc. , USA.
- SHANT P. Gupta (2002), “Solid state phase transformations.” Allied Publishers Private Limited. pp. 28–29.
- WINSTON, W. (2003), “Operations Research Applications and Algorithms”, 4th. Edition
- WU NESA ve COPPINS RICHARD (1981), “Linear Programming and Extensions”, McGraw-Hill Inc., New York



Süreli Yayınlar

- ALVARENGA, H. D. , VAN DE PUTTE, T. , VAN STEENBERGE, N., SIETSMA, J., TERRY, H. (2015), "Influence of Carbide Morphology and Microstructure on the Kinetics of Superficial Decarburization of C-Mn Steels", Metallurgical and Materials Transactions, V. 46, s 123-133
- ARUMUGAM M. S. , CHANDRAMOHAN A. VE MURTHY G. R. (2011), "On the optimal control of steel annealing processes via various versions of genetic and particle swarm optimization algorithms", International Journal of Bio-Inspired Computation, Cilt 12, s. 371-392
- CARVALHO S.R., ONG T.H, GUIMARAES G. (2006), "A mathematical and computational model of furnaces for continuous steel strip processing", Journal of Materials Processing Technology , Cilt 178, s. 379 – 387
- GHASEM DINI G. , VAGHEFI S. M. M. , LOTFIANI M. (2008), "Computational and experimental failure analysis of continuous-annealing furnace radiant tubes", Journal of Engineering Failure Analysis, Cilt 15: 445–457.
- CHO Y. C. , CASSANDRAS C. G., WOOK HYUN KWON W. H. (2004), "Optimal control for steel annealing processes as hybrid systems", Journal of Control Engineering Practice", Cilt 12, s. 1319–1328
- CHRISTOPHER D. KELLY, DHANI WATANAPONGSE AND KENNETH M. G. (1998), "Application of Modern Control to a Continuous Anneal Line", IEEE Control Systems Magazine, 1998, 32-37.
- GOSWAMI A., SINGH R.K., PRASAD A., CHOWDHURI S. (2012), "Mathematical model based control system for Silicon Steel Mill of Rourkela Steel Plant", Journal of Procedia Technology, Cilt 4, s. 867-872
- GUO C. , ZHANG Y , YOU X. , CHEN X. AND ZHANG Y. (2009), "Optimal Control of Continuous Annealing Process Using PSO", International Conference on Automation and Logistics, Cilt 10, s. 602-606

- HACIOĞLU, A. (2011), “Doğrusal olmayan optimizasyon problemleri için taşınır algoritmik fonksiyonlar yöntemi”, Havacılık ve Uzay Teknolojileri Dergisi, Cilt 5, s.1-9
- HAMZAÇEBİ, C. VE KUTAY, F. (2004), “Yapay Sinir Ağları ile Türkiye Elektrik Enerji Tüketiminin 2010 Yılına Kadar Tahmini”, Gazi Üniversitesi Müh. Mimarlık Fakültesi Dergisi, Cilt 19, No. 3, s.227-233
- JEONG S. H. , YI J. J. , KIM J. K. AND HA M. Y. (1991), “Computer Modeling of the Continuous Annealing Furnace”, Journal of KSME, Cilt 5,s. 16-21
- LEUNG S. C. H., WU Y. VE K. K. LAI K. K.(2003), “Multi-Site Aggregate Production Planning With Multiple Objectives: A Goal Programming Approach”, Production Planning & Control”, Cilt 14, No. 5, s.427.
- LI S. , CHENA Q. VE HUANG Q. (2006), “A Dynamic temperature modeling of continuous annealing furnace using GGAP-RBF neural network”, Journal of Neurocomputing, Cilt 69, s. 523–536
- MARLOW D. (1996), “Modelling direct-fired annealing furnaces for transient operations”, Journal of Application Mathematics Modelling, Cilt 20, s.34-40
- MING Z. , DATAI Y. VE JIANGANG Z. (2008), “A New Strip Temperature Control Method for the Heating Section of Continuous Annealing Line”, IEEE Conference on Cybernetics and Intelligent Systems, Cilt 1, s.861-864
- ROBERT W. C. VE PETER H., (1996), “Physical Metallurgy”. Cilt 2. Elsevier Science. s. 10–11.
- SEZER Ö.F. VE COŞKUN E. (2016), “Sürekli Tavlama Hatlarında Isıtma Optimizasyonu ile Enerji Tasarrufu Sağlanması”, Alphanumeric journal, Volume 4, Issue 1, s. 73 - 83
- TİMOR, M. , (2001), “Yöneylem Araştırması ve İşletmecilik Uygulamaları”, İşletme Fakültesi Yayın No: 280, İstanbul, s. 195

- YANAGISHIMA F. , SHIMOYAMA Y. , SUZUKI M., SUNAMI H. , HAGA T. , IDA Y. VE IRIE T. (1981), “Characteristics and Operation of Multipurpose Continuous Annealing Line at Chiba Works”, Kawasaki Steel Technical Report, Cilt 2, s. 1-13
- YOSHITANI N. VE AKIHIKO H. (1998), “Model-Based Control of Strip Temperature for the Heating Furnace in Continuous Annealing”, IEEE Transactions on Control Systems Technology, Cilt 6, s. 146-156
- ZHANG Y. , LI S. , TENGA Y. (2012), “Dynamic processes monitoring using recursive kernel principal component analysis”, Journal of Chemical Engineering Science, Cilt 72, s. 78–86
- WAN F. , WANG Y. VE QIN S. (2012), “Modeling of Strip Heating Process in Vertical Continuous Anealing Furnace”, Journal Of Iron And Steel Research, Cilt 19, s. 29-36
- WANG W. , LI J. , HUA F., LIU X. (2010), “Application of Self-Learning to Heating Process Control of Simulated Continuous Anneal”, Journal Of Iron And Steel Research, Cilt 17, s. 27-31

Diğer Yayınlar

- AKÇAYOL, DOÇ.DR. M.A.(2015) “Zeki Optimisasyon Teknikleri Ders Notları”, Gazi Üniversitesi Bilgisayar Mühendisliği Bölümü,
“<http://w3.gazi.edu.tr/~akcayol/files/ZOL1Giris.pdf>”
- AKDOĞAN A. (2008), “Demir – Karbon Denge Diyagramı” , Yıldız Teknik Üniversitesi Makina fakültesi,
www.yildiz.edu.tr/~akdogan/lessons/malzeme2/Demir_Esasli_Malzemeler.pdf
- AKDOĞAN A. (2) (2008), “Çeliğin Isıl İşlemleri” , Yıldız Teknik Üniversitesi Makina fakültesi,
http://www.yildiz.edu.tr/~akdogan/lessons/malzeme1/Celigin_isil_islemleri.pdf
- ARAS N. (2011) , “Doğrusal Olmayan Programlama”, Anadolu Üniversitesi Endüstri Mühendisliği Bölümü,
<http://www.endustri.anadolu.edu.tr/nila/IST328/duyuru/DogrusalOlmayan1.pdf>
- BARIŞKAN B. (2012) , “Yapay Zeka Teknolojine Genel Bir Bakış”,
<http://ahmetkacan.blogspot.com.tr/>,
- BAYRAKTAR D. VE ÇEBİ F. (2003), “Yöneylem Araştırması Ders Notları”, İtÜ İşletme Fakültesi İşletme Mühendisliği Bölümü”, Erişim Tarihi:01.06.2017
- BİYOLOJİ DERS NOTLARI, “Denetleme ve Düzenleme Sistemi, Sinir Sistemi ve Endokrin Sistem”, <http://www.biyolojidersnotlari.com/denetleme-ve-duzenleme-sistemi-sinir-sistemi-ve-endokrin-sistem.html>, Erişim Tarihi:13.06.2017
- ÇALIŞKAN A. (2017),” Isıl İşlemler”, makine2.kocaeli.edu.tr/malzeme/isilislemler.pdf
- CANAN,S. (2015), “Merkezi Sinir Sisteminin Genel yapısı”,
<http://www.sinancanan.net/?p=382>, Erişim Tarihi:19.05.2015
- ÇAYIROĞLU, İ. (2015), İleri Algoritma Analizi – 5 Yapay Sinir Ağları,
<http://www.ibrahimcayiroglu.com/Dokumanlar/IleriAlgoritmaAnalizi/IleriAlgoritmaAnalizi-5.Hafta-YapaySinirAglari.pdf>, Karabük Üniversitesi

- ÇELİK, B. (2008). YSA Metodolojisi ile Zaman Serisi Analizi: Teori ve Uygulama, Yüksek Lisans Tezi, Marmara Üniversitesi Sosyal Bilimler Enstitüsü, İstanbul.
- CURA, T. (2004) , Karar Verme Aracı Olarak Yapay Sinir Ağları ile Portföy Analizi, Yayınlanmamış Doktora Tezi, İstanbul Üniversitesi Sosyal Bilimler Enstitüsü
- DAĞ, İ. (2006), “Hopfield Ağı”, Eskisehir Osmangazi Üniversitesi, www.ahmetselman.com/document/hopfield.ppt
- DEMİRHAN, H. , “Malzemenin Tanımı ve Seçimi”, <http://w3.balikesir.edu.tr/~demirhan/MalzemeDersNotlar.pdf>, Balıkesir Üniversitesi Edremit Yüksek Okulu, Erişim Tarihi: 28.03.2016
- DERİN, C. (2007), Bulanık Hedef Programlama ve İşletme Yönetiminde bir Uygulama, Yayınlanmamış Doktora Tezi, Sakarya Üniversitesi
- EBRD, “EBRD Türkiye’nin en büyük demir-çelik üreticisi Erdemir’i verimli kaynak kullanımı için destekliyor.”, <http://www.ebrd.com/news/2016/erdemir-turkish.html>
- ELEKTRİKPORT (2013), “Enerji Yönetimi”, <http://www.elektrikport.com/teknik-kutuphane/enerji-yonetimi/10058#ad-image-0>
- ENERJİ DERGİSİ, “Yüksek enerji maliyetleri demir çelik sektörünü yatırıma zorluyor“, <http://www.enerjidergisi.com.tr/haber/2012/12/yuksek-enerji-maliyetleri-demir-celik-sektorunu-yatirim-zorluyor>, Erişim Tarihi: 21.11.2016
- ERDEMİR (CPT) (2017), “ 2. Soğuk Haddehane Sürekli Asitleme ve Tandem Hattı Üretim Akış Şeması”
- ERDEMİR (CAL) (2017), “ 2. Soğuk Haddehane Sürekli Tavlama Hattı Üretim Akış Şeması”,
- ERİŞİR E. (2013), “Çeliklerde Malzeme Bilimi ve Son Gelişmeler”, Kocaeli Üniversitesi Metalurji ve Malzeme Mühendisliği Bölümü, metalurji.kocaeli.edu.tr/files/DersNotlari/mmt209-13.pdf

KABALCI E. (2007), “Yapay Sinir Ağları”, Nevşehir Üniversitesi, Mühendislik-Mimarlık Fakültesi Elektrik Elektronik Müh.,
<https://ekblc.files.wordpress.com/2014/02/ysa.pdf>

KAKICI, A. (2009), ”Yapay Sinir Ağlarına Giriş”,
<http://ahmetkakici.github.io/yazilim/yapay-sinir-aglarina-giris/>, Erişim Tarihi: 17.05.2015

KARAALİ F.Ç. (2006), Use Of Artificial Neural Networks And Cognitive Mapping Methodology In Predicting Unemployment Rates And Employment Index Level, İstanbul Technical University Institute Of Science And Technology

KARAHAN M. (2011), İstatistiksel Tahmin Yöntemleri: Yapay Sinir Ağları Metodu ile Ürün Talep Tahmini Uygulaması, Selçuk Üni. Sosyal Bilimler Enstitüsü

KHAIRA, H.K. (2013), Annealing, Dept. Of Mat. Sci. And Met. Eng. , Indian Institute of Technology, Bombay

KOÇAK H., Doğrusal Programlama , Karar Verme ve Modelleme,
<http://mimoza.marmara.edu.tr/~hkocak/1.DP%20Karar%20verme.ppt>, Erişim Tarihi: 23.02.2015

KOZ M., İstirahat Membran Potansiyeli ve Aksiyon Potansiyeli, Ankara Üniversitesi Açık Ders Malzemeleri,
http://acikders.ankara.edu.tr/pluginfile.php/8344/mod_resource/content/1/4.1.AksiyonPotansiyeli.pdf, Erişim Tarihi: 23.05.2017

MATHWORKS DOCUMENTATION, “newff train toolbox”,
<http://radio.feld.cvut.cz/matlab/toolbox/nnet/newff.html>

MATHWORKS DOCUMENTATION (1), “Algorithms”,
<https://www.mathworks.com/help/nnet/ref/trainscg.html#f8-2931204>

MATHWORKS DOCUMENTATION (2), “Choose a Multilayer Neural Networks Training Function”, <https://www.mathworks.com/help/nnet/ug/choose-a-multilayer-neural-network-training-function.html>

MATHWORKS DOCUMENTATION (3), “trainlm”,
<https://www.mathworks.com/help/nnet/ref/trainlm.html>

MATHWORKS DOCUMENTATION (4), “trainscg”,
<https://www.mathworks.com/help/nnet/ref/trainscg.html>

MATHWORKS DOCUMENTATION (5), “trainrp”,
<https://www.mathworks.com/help/nnet/ref/trainrp.html>

MERİÇ O.A. (2004), Veri Madenciliği Aracı Olarak Genetik Algoritmalar ile Yapay Sinir Ağları ve Genetik Algoritma-Yapay Sinir Ağı Melez Modelinin Müşteri Değerlendirilmesinde Kullanılması, Yayınlanmamış Doktora Tezi, İstanbul Üniversitesi Sosyal Bilimler Enstitüsü

METALURJİ VE MALZEME PLATFORMU (2013), “Haddeleme İşlemi”,
<http://www.metalurjik.net/tag/haddeleme>, Erişim Tarihi: 01.02.2017

METALURJİ VE MALZEME PLATFORMU (2013), “Isıl işlem nedir? Neden Yapılır?”, <http://www.metalurjik.net/genel/isil-islem-nedir-neden-yapilir.html>, Erişim Tarihi: 01.02.2017

METALUZMANI (2015), ”Prof Dr. İrfan AY Balıkesir Üniversitesi Ders Notları “,
<http://www.metaluzmani.com/allotropi-ve-demirin-allotropisi/>

MMP (Metalurji Malzeme Platformu), “Soğuk Haddeleme”,
<http://www.metalurjik.net/genel/soguk-haddeleme-islemi.html>, Erişim Tarihi: 04.06.2017

MÜHENDİSHANE, “Katı hal faz dönüşümlerine genel bakış”,
<https://muhendishane.org/kutuphane/malzeme-biliminin-fiziksel-temelleri/kati-hal-faz-donusumlerine-genel-bakis/>, Erişim Tarihi: 04.03.2017

NEURAL NETWORK TOOLBX, Levenberg-Marquardt (trainlm)
<http://matlab.izmiran.ru/help/toolbox/nnet/backpr12.html>

NEURAL NETWORK TOOLBX, Scaled Conjugate Gradient(trainscg),
<http://matlab.izmiran.ru/help/toolbox/nnet/backpr12.html>

NKK CORPORATION (1995), “ CAL Level-2 Model Document”

- ÖZKAN F. (2017), Fe-C Faz Diyagramı ve Özellikleri,
<https://furkanozkan92.wordpress.com/2013/03/12/fe-c-faz-diyagrami-ve-ozellikleri/>, Erişim Tarihi: 10.3.2017
- ÖZTÜRK C. (2011), Yapay Sinir Ağlarının Yapay Arı Kolonisi Algoritması ile Eğitilmesi, Erciyes Üniversitesi Fen Bilimleri Enstitüsü
- ÖZTÜRK S. (2016), “Faz Diyagramları Ders Notları”, KTÜ Metalurji ve Malzeme Mühendisliği Bölümü,
aves.ktu.edu.tr/ImageOfByte.aspx?Resim=8&SSNO=4&USER=7811
- RESEARCH HUBS (2017) , “Hessian Matrix”,
<http://researchhubs.com/post/maths/fundamentals/hessian-matrix.html>, Erişim Tarihi: 9.6.2017
- SANCAKDAR T. (2002), “Yapay Sinir Ağı Destekli Kalman Filtresi Yardımıyla Hedef İzleme”, İTÜ Fen Bilimleri Enstitüsü, Y. Lisans tezi.
- SARIAY S. N. (2012), Hedef Programlama ve bir İşletme Uygulaması, Yayınlanmamış Doktora Tezi, Aksaray Üniversitesi
- SERT, Ü. (2015), Haddeleme Hakkında. <http://www.arsivbelge.com/yaz.php?sc=2418>
- SIDEPLAYER, Kobilerde stratejik yönetim, <http://slideplayer.biz.tr/slide/5583812/>
- SİNİRSİSTEMİ.GEN.TR, Sinir Sistemi, <http://www.sinirsistemi.gen.tr/periferik-sinir-sistemi.html>, Erişim Tarihi: 17.11.2016
- STACK EXCHANGE, How to choose the number of hidden layers and nodes in a feedforward neural network?, <http://stats.stackexchange.com/questions/181/how-to-choose-the-number-of-hidden-layers-and-nodes-in-a-feedforward-neural-netw>, Erişim Tarihi: 17.11.2016
- ŞAHİN, A. (2012), Çelikler ve Isıl İşlem Ders Notu, Bozok Üniversitesi
- TAKÇI, H. (2017), “ (LVQ) Linear Vector Quantization”, Gebze Teknik Üniversitesi, sucati.org/YSA/index_dosyalar/bolum-5.pdf

TOSUN, K., Katı Cisimlerin İç Yapısı, Dokuz Eylül Üniversitesi,
kisi.deu.edu.tr/userweb/kamile.tosun/02_Kati_cisimlerin_ic_yapisi.pdf, Erişim
Tarihi: 27.03.2016

TÜRKİYE SANAYİ VE KALKINMA BANKASI (2011),”Türkiye’deki Enerji
Verimliliği Çalışmaları”, www.tskb.com.tr/i/content/486_1_Turkiyede-Enerji-
Verimliliği-Calismalari.pdf

YAZGAN Y. (2015), “Öğrenmenin Norolojik Temelleri”,
<http://www.yankiyazgan.com/ogrenmenin-norolojik-temelleri-2/>, Erişim Tarihi:
5.9.2015

YAVUZ O. (2013), Tedarik Zinciri Performansının Değerlendirilmesinde Yapay Sinir
Ağlarının Kullanımı Ve Bir Model Önerisi, Gazi Üniversitesi Sosyal Bilimler
Enstitüsü

YILMAZ S. (2008) , “Optimizasyon”,
serhatyilmaz.org/dersler/sc/SayCoz_dosyalar/10.Optimizasyon.ppt, Kocaeli
Üniversitesi, Mühendislik Fakültesi

YÜKSEK, A.G. (2007), Hava Kirliliği Tahmininde Çoklu Reresyon Analiz ve YSA
Yönteminin Karşılaştırılması, Cumhuriyet Üniversitesi Sosyal Bilimler Enstitüsü
Doktora Tezi

EKLER

EK-1: STH Hattı Özellikleri

Sürekli Tavlama Hattı

Teknik Özellikler

Üretime Geçiş Tarihi	Temmuz 1996
Üretim Kapasitesi	900.000 ton/yıl
Bobin Ağırlığı (Maks.)	28 ton (Maks.)
Şerit Genişliği	625-1525 mm
Şerit Kalınlığı	0.4 - 2.0 mm
Kapasite	176 ton/saat
Hız (Giriş/Fırın/Çıkış)	480/320/540 m/dk.
Lup Yüksekliği (Giriş/Çıkış)	41/40 m
Lup Kapasitesi (Giriş/Çıkış)	630/710 m
Temperde Maks. Uzama	%2
Toplam Hat Uzunluğu	330 m



Sürekli Tavlama Hattı; maksimum 320 m/dk proses hızına ve 900.000 ton/yıl kapasiteye sahiptir.

NKK (Japonya) • MARUBENI(Japonya) • MITSUBISHI (Japonya) • SELAS (Fransa) • MANNESMAN DEMAG SACK (Almanya)

EK-2: Isıl Modeli Tablosu

Tavlama Kalitesi	Şerit Sıcaklıkları[°C]						Zaman [sec]
	TSPH	TSRH	TSRS	TSRQ	TSOA	TSFC	TMRS
CQ	≤ 200	720 ± 30	720 ± 30	400 ± 40	320 ± 30	≤ 160	≥ 40
DQ-1	≤ 200	770 ± 30	770 ± 30	400 ± 40	320 ± 30	≤ 160	≥ 40
DQ-2	≤ 200	780 ± 30	780 ± 30	450 ± 40	320 ± 50	≤ 160	≥ 40
DDQ	≤ 200	800 ± 30	800 ± 30	450 ± 40	320 ± 50	≤ 160	≥ 40
EDDQ	≤ 200	850 ± 30	850 ± 30	450 ± 40	320 ± 50	≤ 160	≥ 40
HSS-45	≤ 200	800 ± 30	800 ± 30	400 ± 40	320 ± 30	≤ 160	≥ 40
HSS-60	≤ 200	800 ± 30	800 ± 30	400 ± 40	320 ± 30	≤ 160	≥ 40

EK-3: Tavlama Kalitelerine ve Boyutlarına göre Maksimum Hız Değerleri

CQ KALİTELER											
7110K, 6837K, 7001K, 1112I, 2002K, 2005K, 2242K, 4237K, 7106K, 7108K, 6010K, 5009K, 7112K, 7123K, 6182K, 7512K, 7608K											
		<650	650-749	750-849	850-949	950-1049	1050-1149	1150-1249	1250-1349	1350-1449	≥1450
		625	700	800	900	1000	1100	1200	1300	1400	1500
<0,45	0.40	320	320	320	320	320	250	250			
0,45-0,54	0.50	320	320	320	320	320	280	280	250		
0,55-0,64	0.60	280	280	280	320	320	320	320	280	260	
0,65-0,74	0.70	260	260	260	320	320	320	320	280	260	
0,75-0,84	0.80	260	260	260	320	320	320	320	280	260	
0,85-0,94	0.90	260	260	260	320	320	320	320	280	250	125
0,95-1,09	1.00	260	260	260	310	310	310	257	236	210	125
1,10-1,29	1.20	260	260	260	260	260	260	251	232	200	125
1,30-1,69	1.50	208	208	208	208	208	208	208	174	155	100
1,70-1,89	1.80	174	174	174	173	173	174	173	145	129	86
≥1,90	2.00	156	156	156	156	156	156	156	130	116	80
HSS KALİTELER											
7132K, 7136K, 1113I, 6012K, 4237K, 7106K, 6137K, 7111K, 7125K, 7128K, 7142K, 7140K, 7612K, 7659K											
		<650	650-749	750-849	850-949	950-1049	1050-1149	1150-1249	1250-1349	1350-1449	≥1450
		625	700	800	900	1000	1100	1200	1300	1400	1500
<0,45	0.40										
0,45-0,54	0.50										
0,55-0,64	0.60										
0,65-0,74	0.70	110	110	110	100	100	100	100			
0,75-0,84	0.80	110	110	110	100	100	100	100			
0,85-0,94	0.90	110	110	110	100	100	100	100	100		
0,95-1,09	1.00	110	110	110	100	100	100	100	100		
1,10-1,29	1.20	110	110	110	100	100	100	100	100		
1,30-1,69	1.50	110	110	110	100	100	100	100	100		
1,70-1,89	1.80	110	110	110	100	100	100	100	100		
≥1,90	2.00	100	100	100	100	100	100	100	100		
DQ2 KALİTELER											
7314K, 7114K, 5105K, 7113K, 7123K, 7122K, 7124K, 7514K, 7513K, 7516K											
		<650	650-749	750-849	850-949	950-1049	1050-1149	1150-1249	1250-1349	1350-1449	≥1450
		625	700	800	900	1000	1100	1200	1300	1400	1500
<0,45	0.40	320	320	320	320	320	320	290			
0,45-0,54	0.50	320	320	320	320	320	320	290			
0,55-0,64	0.60	280	280	280	310	310	310	310	260		
0,65-0,74	0.70	260	260	260	310	310	310	285	265		
0,75-0,84	0.80	275	275	275	275	275	275	255	235	220	
0,85-0,94	0.90	250	250	250	250	250	250	230	210	200	
0,95-1,09	1.00	210	210	210	210	210	210	190	175	165	100
1,10-1,29	1.20	210	210	210	210	210	210	190	175	165	100
1,30-1,69	1.50	165	165	165	165	165	165	165	150	150	100
1,70-1,89	1.80	143	143	143	143	143	143	130	123	118	86
≥1,90	2.00	135	135	135	135	135	135	110	110	100	80

EDDQ KALİTELER											
7114K, 7115K, 7001K, 7116K, 7124K, 7315K, 7316K, 7514K, 7516K											
		<650	650-749	750-849	850-949	950-1049	1050-1149	1150-1249	1250-1349	1350-1449	≥1450
		625	700	800	900	1000	1100	1200	1300	1400	1500
<0,45	0.40	320	320	320	320	320	320	290			
0,45-0,54	0.50	320	320	320	320	320	320	290			
0,55-0,64	0.60	280	280	280	310	310	310	310	260		
0,65-0,74	0.70	260	260	260	310	310	310	285	265		
0,75-0,84	0.80	275	275	275	275	275	275	255	235	220	
0,85-0,94	0.90	250	250	250	250	250	250	230	210	200	
0,95-1,09	1.00	210	210	210	210	210	210	190	175	165	100
1,10-1,29	1.20	210	210	210	210	210	210	190	175	165	100
1,30-1,69	1.50	165	165	165	165	165	165	165	150	150	100
1,70-1,89	1.80	143	143	143	143	143	143	130	123	118	86
≥1,90	2.00	135	135	135	135	135	135	110	110	100	80
DDQ KALİTELER											
7313K, 7114K, 1122K, 7113K, 7123K, 1522K, 1526K											
		<650	650-749	750-849	850-949	950-1049	1050-1149	1150-1249	1250-1349	1350-1449	≥1450
		625	700	800	900	1000	1100	1200	1300	1400	1500
<0,45	0.40	320	320	320	320	310	250	250			
0,45-0,54	0.50	320	320	320	320	310	280	280			
0,55-0,64	0.60	280	280	280	320	320	320	320	280		
0,65-0,74	0.70	260	260	260	320	320	320	320	280		
0,75-0,84	0.80	260	260	260	320	320	320	320	280	250	
0,85-0,94	0.90	260	260	260	291	300	300	300	268	250	
0,95-1,09	1.00	245	245	245	262	262	262	262	242	224	125
1,10-1,29	1.20	218	218	218	218	218	218	218	201	187	125
1,30-1,69	1.50	174	174	174	174	174	174	174	161	150	100
1,70-1,89	1.80	145	145	146	145	145	146	145	134	125	86
≥1,90	2.00	131	131	131	131	131	131	131	121	112	80

EK-4: Isı Transferi Teorisine Dayanan Matematiksel Modeller

Bölüm	Matematiksel Model		
RTH	$RHO * H * \frac{dQ_{ts}}{dt} = 2 * U * U_{ts} * \sigma * [(TF + 273)^4 - (TS + 273)^4]$		
	Q_{ts}	[kcal/kg]	Şerit ısı kapasitesi
	TS	[°C]	Şerit sıcaklığı
	TF	[°C]	Fırın sıcaklığı
	t	[h]	Zaman
	H	[m]	Şerit kalınlığı
	RHO	[kg/m ³]	Şerit yoğunluğu
	U	[-]	Model güncelleme parametresi
	U_{ts}	[-]	Isı transfer katsayısı
	σ	[kcal/ h m ² K ⁴]	Stefan-Boltzman sabiti
RTS	TS = U * TF		
	TS	[°C]	Şerit sıcaklığı
	TF	[°C]	Fırın sıcaklığı
	U	[-]	Model güncelleme parametresi
GJS	$RHO * H * \frac{dQ_{ts}}{dt} = 2 * U * \left[U_1 * C_1 * P^{C_2} * \left(\frac{C_3}{(TG + 273)} \right)^{C_4} * (TG - TS) + U_2 * C_5 * \sigma * ((TF + 273)^4 - (TS + 273)^4) + U_3 * C_6 * (TF - TS) \right]$		
	Q_{ts}	[kcal/kg]	Şerit ısı kapasitesi
	TS	[°C]	Şerit sıcaklığı
	TF	[°C]	Fırın sıcaklığı
	TG	[°C]	Soğutma gazı sıcaklığı
	t	[h]	Zaman
	H	[m]	Şerit kalınlığı
	RHO	[kg/m ³]	Şerit yoğunluğu
	U	[-]	Model güncelleme parametresi
	$U_1 - U_3$	[-]	Isı transfer katsayısı
	$C_1 - C_6$	[-]	Sabit katsayı
	σ	[kcal/ h m ² K ⁴]	Stefan-Boltzman sabiti
RQS	$RHO * H * \frac{dQ_{ts}}{dt} = U * [C_1 * (THETA)^{C_2} + C_3] * (TC - TS)$		
	Q_{ts}	[kcal/kg]	Şerit ısı kapasitesi
	TS	[°C]	Şerit sıcaklığı
	TC	[°C]	Soğutma suyu sıcaklığı
	t	[h]	Zaman
	H	[m]	Şerit kalınlığı
	RHO	[kg/m ³]	Şerit yoğunluğu
	THETA	[deg]	Şerit ile merdanenin değme açısı
	U	[-]	Model güncelleme parametresi
	$C_1 - C_3$	[-]	Sabit katsayı

EK-5: STH Konveyör Optimizasyon Programı

Program.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Linq;
using System.Text;
using System.Net;
using System.Net.Sockets;
using System.Runtime.InteropServices;
//Add MySql Library
using MySql.Data.MySqlClient;

namespace ConveyorOptimizationConsole
{
    class Program
    {
        public static Optimization MyOptimize = new Optimization();

        public struct DataRecord_Opt
        {
            public Int16[] Por;
            public string[] COILNO;
            public string[] ANNCODE;
            public string[] ACTTHICK;
            public string[] TARWID;
            public string[] ACTLEN;
            public string[] ACTWEI;
        }

        //DataRecord_Opt SelectedData = new DataRecord_Opt();

        private static System.Timers.Timer aTimer;

        static void Main(string[] args)
        {
            StartGettingData();

            // Create a timer and set a two second interval.

            aTimer = new System.Timers.Timer();
            // Start the timer
```

```

aTimer.Enabled = true;
aTimer.Interval = 10000; // 600000;

// Hook up the Elapsed event for the timer.
aTimer.Elapsed += OnTimedEvent;

// Have the timer fire repeated events (true is the default)
aTimer.AutoReset = true;

}

public static void StartGettingData()
{
    DataRecord_Opt SelectedData = new DataRecord_Opt();

    try
    {
        MySqlConnection connection;
        string server;
        string database;
        string uid;
        string password;
        server = "IP STRING";
        database = "Database Name";
        uid = "Database Kullanıcı Adı";
        password = "Şifre";

        string connectionString;
        connectionString = "SERVER=" + server + ";" + "DATABASE=" + database
+ ";" + "UID=" + uid + ";" + "PASSWORD=" + password + ";";

        connection = new MySqlConnection(connectionString);

        connection.Open();

        string SelectQuery = "select * from cal_conveyor";

        //Create Command
        MySqlCommand cmd = new MySqlCommand(SelectQuery, connection);
        //Create a data reader and Execute the command
        MySqlDataReader dataReader = cmd.ExecuteReader();

        //Read the data and store them in the list
        int i = 0;
        Array.Resize(ref SelectedData.Por, 10);
        Array.Resize(ref SelectedData.COILNO, 10);
        Array.Resize(ref SelectedData.ANNCODE, 10);
    }
    catch { }
}

```

```

Array.Resize(ref SelectedData.ACTTHICK, 10);
Array.Resize(ref SelectedData.TARWID, 10);
Array.Resize(ref SelectedData.ACTLEN, 10);
Array.Resize(ref SelectedData.ACTWEI, 10);

while (dataReader.Read())
{
    if (dataReader.GetString(0) != null)
        SelectedData.Por[i] = Convert.ToInt16( dataReader.GetString(0));
    else SelectedData.Por[i] = 0;
    if (dataReader.GetString(1) != null)
        SelectedData.COILNO[i] = dataReader.GetString(1);
    else SelectedData.COILNO[i] = " ";
    if (dataReader.GetString(2) != null)
        SelectedData.ANNCODE[i] = dataReader.GetString(2);
    else SelectedData.ANNCODE[i] = " ";
    if (dataReader.GetString(3) != null)
        SelectedData.ACTTHICK[i] = dataReader.GetString(3);
    else SelectedData.ACTTHICK[i] = " ";
    if (dataReader.GetString(4) != null)
        SelectedData.TARWID[i] = dataReader.GetString(4);
    else SelectedData.TARWID[i] = " ";
    if (dataReader.GetString(5) != null)
        SelectedData.ACTLEN[i] = dataReader.GetString(5);
    else SelectedData.ACTLEN[i] = " ";
    if (dataReader.GetString(6) != null)
        SelectedData.ACTWEI[i] = dataReader.GetString(6);
    else SelectedData.ACTWEI[i] = " ";

    i++;
}

//close Data Reader
dataReader.Close();

//close Connection
connection.Close();

Optimization.DataRecord_Opt OptStr = new Optimization.DataRecord_Opt();

Array.Resize(ref OptStr.Por, 10);
Array.Resize(ref OptStr.COILNO, 10);
Array.Resize(ref OptStr.ANNCODE, 10);
Array.Resize(ref OptStr.ACTTHICK, 10);
Array.Resize(ref OptStr.TARWID, 10);
Array.Resize(ref OptStr.ACTLEN, 10);
Array.Resize(ref OptStr.ACTWEI, 10);

```

```

        for ( i = 0; i < 10; i++)
        {
            OptStr.COILNO[i] = SelectedData.COILNO[i];
            OptStr.ANNCODE[i] = SelectedData.ANNCODE[i];
            OptStr.ACTTHICK[i] = SelectedData.ACTTHICK[i];
            OptStr.TARWID[i] = SelectedData.TARWID[i];
            OptStr.ACTLEN[i] = SelectedData.ACTLEN[i];
            OptStr.ACTWEI[i] = SelectedData.ACTWEI[i];
            OptStr.Por[i] = SelectedData.Por[i];
        }

        MyOptimize.Optimize_Schedule(OptStr);

    }
    catch (MySQL.Data.MySqlClient.MySqlException e)
    {
        Console.WriteLine(e.Message);
    }
}

private static void OnTimedEvent(Object source, System.Timers.ElapsedEventArgs
e)
{
    StartGettingData();
    Console.WriteLine("Timer fired....");
}

}
}

```

STH KONVEYÖR PROGRAMI - OPTIMIZATION.CS

```

using System;
using System.Data;
using System.Data.OleDb;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.IO;
using Microsoft.SolverFoundation.Services;
using Microsoft.SolverFoundation.Solvers;
using Excel = Microsoft.Office.Interop.Excel;

namespace ConveyorOptimizationConsole
{
    public class Optimization

```

```

{
    public Excel.Application excelApp ;
    public Excel.Workbook wb;
    public Excel.Worksheet workSheet;
    public string Filename = @"c:\WriteSolverValues.txt";

    public struct DataRecord_Opt
    {
        public Int16[] Por;
        public string[] COILNO;
        public string[] ANNCODE;
        public string[] ACTTHICK;
        public string[] TARWID;
        public string[] ACTLEN;
        public string[] ACTWEI;
    }

    public Optimization()
    {
    }

    public void Optimize_Schedule(DataRecord_Opt OptData )
    {
        DataRecord_Opt strClassData = new DataRecord_Opt();

        // Define and resize class
        Array.Resize(ref strClassData.Por, 10);
        Array.Resize(ref strClassData.COILNO, 10);
        Array.Resize(ref strClassData.ANNCODE, 10);
        Array.Resize(ref strClassData.ACTTHICK, 10);
        Array.Resize(ref strClassData.TARWID, 10);
        Array.Resize(ref strClassData.ACTLEN, 10);
        Array.Resize(ref strClassData.ACTWEI, 10);

        int Count = 0;
        if (OptData.Por[0] == 1)
        {
            if (OptData.COILNO[9] != null && OptData.COILNO[9] != "")
            {
                strClassData.COILNO[Count] = OptData.COILNO[9];
                strClassData.ANNCODE[Count] = OptData.ANNCODE[9];
                strClassData.ACTTHICK[Count] = OptData.ACTTHICK[9];
                strClassData.TARWID[Count] = OptData.TARWID[9];
                strClassData.ACTLEN[Count] = OptData.ACTLEN[9];
                strClassData.ACTWEI[Count] = OptData.ACTWEI[9];
                Count++;
            }

            if (OptData.COILNO[4] != null && OptData.COILNO[4] != "")
            {
                strClassData.COILNO[Count] = OptData.COILNO[4];
                strClassData.ANNCODE[Count] = OptData.ANNCODE[4];
                strClassData.ACTTHICK[Count] = OptData.ACTTHICK[4];
                strClassData.TARWID[Count] = OptData.TARWID[4];
                strClassData.ACTLEN[Count] = OptData.ACTLEN[4];
                strClassData.ACTWEI[Count] = OptData.ACTWEI[4];
                Count++;
            }

            if (OptData.COILNO[8] != null && OptData.COILNO[8] != "")

```

```

{
    strClassData.COILNO[Count] = OptData.COILNO[8];
    strClassData.ANNCODE[Count] = OptData.ANNCODE[8];
    strClassData.ACTTHICK[Count] = OptData.ACTTHICK[8];
    strClassData.TARWID[Count] = OptData.TARWID[8];
    strClassData.ACTLEN[Count] = OptData.ACTLEN[8];
    strClassData.ACTWEI[Count] = OptData.ACTWEI[8];
    Count++;
}
if (OptData.COILNO[3] != null && OptData.COILNO[3] != "")
{
    strClassData.COILNO[Count] = OptData.COILNO[3];
    strClassData.ANNCODE[Count] = OptData.ANNCODE[3];
    strClassData.ACTTHICK[Count] = OptData.ACTTHICK[3];
    strClassData.TARWID[Count] = OptData.TARWID[3];
    strClassData.ACTLEN[Count] = OptData.ACTLEN[3];
    strClassData.ACTWEI[Count] = OptData.ACTWEI[3];
    Count++;
}
if (OptData.COILNO[7] != null && OptData.COILNO[7] != "")
{
    strClassData.COILNO[Count] = OptData.COILNO[7];
    strClassData.ANNCODE[Count] = OptData.ANNCODE[7];
    strClassData.ACTTHICK[Count] = OptData.ACTTHICK[7];
    strClassData.TARWID[Count] = OptData.TARWID[7];
    strClassData.ACTLEN[Count] = OptData.ACTLEN[7];
    strClassData.ACTWEI[Count] = OptData.ACTWEI[7];
    Count++;
}
if (OptData.COILNO[2] != null && OptData.COILNO[2] != "")
{
    strClassData.COILNO[Count] = OptData.COILNO[2];
    strClassData.ANNCODE[Count] = OptData.ANNCODE[2];
    strClassData.ACTTHICK[Count] = OptData.ACTTHICK[2];
    strClassData.TARWID[Count] = OptData.TARWID[2];
    strClassData.ACTLEN[Count] = OptData.ACTLEN[2];
    strClassData.ACTWEI[Count] = OptData.ACTWEI[2];
    Count++;
}
if (OptData.COILNO[6] != null && OptData.COILNO[6] != "")
{
    strClassData.COILNO[Count] = OptData.COILNO[6];
    strClassData.ANNCODE[Count] = OptData.ANNCODE[6];
    strClassData.ACTTHICK[Count] = OptData.ACTTHICK[6];
    strClassData.TARWID[Count] = OptData.TARWID[6];
    strClassData.ACTLEN[Count] = OptData.ACTLEN[6];
    strClassData.ACTWEI[Count] = OptData.ACTWEI[6];
    Count++;
}
if (OptData.COILNO[1] != null && OptData.COILNO[1] != "")
{
    strClassData.COILNO[Count] = OptData.COILNO[1];
    strClassData.ANNCODE[Count] = OptData.ANNCODE[1];
    strClassData.ACTTHICK[Count] = OptData.ACTTHICK[1];
    strClassData.TARWID[Count] = OptData.TARWID[1];
    strClassData.ACTLEN[Count] = OptData.ACTLEN[1];
    strClassData.ACTWEI[Count] = OptData.ACTWEI[1];
}

```

```

        Count++;
    }

    if (OptData.COILNO[5] != null && OptData.COILNO[5] != "")
    {
        strClassData.COILNO[Count] = OptData.COILNO[5];
        strClassData.ANNCODE[Count] = OptData.ANNCODE[5];
        strClassData.ACTTHICK[Count] = OptData.ACTTHICK[5];
        strClassData.TARWID[Count] = OptData.TARWID[5];
        strClassData.ACTLEN[Count] = OptData.ACTLEN[5];
        strClassData.ACTWEI[Count] = OptData.ACTWEI[5];
        Count++;
    }

    if (OptData.COILNO[0] != null && OptData.COILNO[0] != "")
    {
        strClassData.COILNO[Count] = OptData.COILNO[0];
        strClassData.ANNCODE[Count] = OptData.ANNCODE[0];
        strClassData.ACTTHICK[Count] = OptData.ACTTHICK[0];
        strClassData.TARWID[Count] = OptData.TARWID[0];
        strClassData.ACTLEN[Count] = OptData.ACTLEN[0];
        strClassData.ACTWEI[Count] = OptData.ACTWEI[0];
        Count++;
    }
}
else if (OptData.Por[0] == 2)
{
    if (OptData.COILNO[4] != null && OptData.COILNO[4] != "")
    {
        strClassData.COILNO[Count] = OptData.COILNO[4];
        strClassData.ANNCODE[Count] = OptData.ANNCODE[4];
        strClassData.ACTTHICK[Count] = OptData.ACTTHICK[4];
        strClassData.TARWID[Count] = OptData.TARWID[4];
        strClassData.ACTLEN[Count] = OptData.ACTLEN[4];
        strClassData.ACTWEI[Count] = OptData.ACTWEI[4];
        Count++;
    }

    if (OptData.COILNO[9] != null && OptData.COILNO[9] != "")
    {
        strClassData.COILNO[Count] = OptData.COILNO[9];
        strClassData.ANNCODE[Count] = OptData.ANNCODE[9];
        strClassData.ACTTHICK[Count] = OptData.ACTTHICK[9];
        strClassData.TARWID[Count] = OptData.TARWID[9];
        strClassData.ACTLEN[Count] = OptData.ACTLEN[9];
        strClassData.ACTWEI[Count] = OptData.ACTWEI[9];
        Count++;
    }

    if (OptData.COILNO[3] != null && OptData.COILNO[3] != "")
    {
        strClassData.COILNO[Count] = OptData.COILNO[3];
        strClassData.ANNCODE[Count] = OptData.ANNCODE[3];
        strClassData.ACTTHICK[Count] = OptData.ACTTHICK[3];
        strClassData.TARWID[Count] = OptData.TARWID[3];
        strClassData.ACTLEN[Count] = OptData.ACTLEN[3];
        strClassData.ACTWEI[Count] = OptData.ACTWEI[3];
        Count++;
    }
}

```

```

if (OptData.COILNO[8] != null && OptData.COILNO[8] != "")
{
    strClassData.COILNO[Count] = OptData.COILNO[8];
    strClassData.ANNCODE[Count] = OptData.ANNCODE[8];
    strClassData.ACTTHICK[Count] = OptData.ACTTHICK[8];
    strClassData.TARWID[Count] = OptData.TARWID[8];
    strClassData.ACTLEN[Count] = OptData.ACTLEN[8];
    strClassData.ACTWEI[Count] = OptData.ACTWEI[8];
    Count++;
}

if (OptData.COILNO[2] != null && OptData.COILNO[2] != "")
{
    strClassData.COILNO[Count] = OptData.COILNO[2];
    strClassData.ANNCODE[Count] = OptData.ANNCODE[2];
    strClassData.ACTTHICK[Count] = OptData.ACTTHICK[2];
    strClassData.TARWID[Count] = OptData.TARWID[2];
    strClassData.ACTLEN[Count] = OptData.ACTLEN[2];
    strClassData.ACTWEI[Count] = OptData.ACTWEI[2];
    Count++;
}

if (OptData.COILNO[7] != null && OptData.COILNO[7] != "")
{
    strClassData.COILNO[Count] = OptData.COILNO[7];
    strClassData.ANNCODE[Count] = OptData.ANNCODE[7];
    strClassData.ACTTHICK[Count] = OptData.ACTTHICK[7];
    strClassData.TARWID[Count] = OptData.TARWID[7];
    strClassData.ACTLEN[Count] = OptData.ACTLEN[7];
    strClassData.ACTWEI[Count] = OptData.ACTWEI[7];
    Count++;
}

if (OptData.COILNO[1] != null && OptData.COILNO[1] != "")
{
    strClassData.COILNO[Count] = OptData.COILNO[1];
    strClassData.ANNCODE[Count] = OptData.ANNCODE[1];
    strClassData.ACTTHICK[Count] = OptData.ACTTHICK[1];
    strClassData.TARWID[Count] = OptData.TARWID[1];
    strClassData.ACTLEN[Count] = OptData.ACTLEN[1];
    strClassData.ACTWEI[Count] = OptData.ACTWEI[1];
    Count++;
}

if (OptData.COILNO[6] != null && OptData.COILNO[6] != "")
{
    strClassData.COILNO[Count] = OptData.COILNO[6];
    strClassData.ANNCODE[Count] = OptData.ANNCODE[6];
    strClassData.ACTTHICK[Count] = OptData.ACTTHICK[6];
    strClassData.TARWID[Count] = OptData.TARWID[6];
    strClassData.ACTLEN[Count] = OptData.ACTLEN[6];
    strClassData.ACTWEI[Count] = OptData.ACTWEI[6];
    Count++;
}

if (OptData.COILNO[0] != null && OptData.COILNO[0] != "")
{
    strClassData.COILNO[Count] = OptData.COILNO[0];
    strClassData.ANNCODE[Count] = OptData.ANNCODE[0];
    strClassData.ACTTHICK[Count] = OptData.ACTTHICK[0];
    strClassData.TARWID[Count] = OptData.TARWID[0];

```

```

        strClassData.ACTLEN[Count] = OptData.ACTLEN[0];
        strClassData.ACTWEI[Count] = OptData.ACTWEI[0];
        Count++;
    }

    if (OptData.COILNO[5] != null && OptData.COILNO[5] != "")
    {
        strClassData.COILNO[Count] = OptData.COILNO[5];
        strClassData.ANNCODE[Count] = OptData.ANNCODE[5];
        strClassData.ACTTHICK[Count] = OptData.ACTTHICK[5];
        strClassData.TARWID[Count] = OptData.TARWID[5];
        strClassData.ACTLEN[Count] = OptData.ACTLEN[5];
        strClassData.ACTWEI[Count] = OptData.ACTWEI[5];
        Count++;
    }
}

// Create min max temperatures variables
double x0min = 0.0, x0max = 0.0;
double x1min = 0.0, x1max = 0.0;
double x2min = 0.0, x2max = 0.0;
double x3min = 0.0, x3max = 0.0;
double x4min = 0.0, x4max = 0.0;
double x5min = 0.0, x5max = 0.0;
double x6min = 0.0, x6max = 0.0;
double x7min = 0.0, x7max = 0.0;
double x8min = 0.0, x8max = 0.0;
double x9min = 0.0, x9max = 0.0;
Int16[] Speed = new Int16[10];

for (int i = 0; i < Count; i++)
{
    if (!string.IsNullOrEmpty(strClassData.ANNCODE[i]) &&
        strClassData.ANNCODE[i] != "\\0\\0\\0\\0" )
        Speed[i] = ReadSpeedFromExcel(strClassData.ANNCODE[i], Convert.ToInt16(strClassData.ACTTHICK[i]), Convert.ToInt16(strClassData.TARWID[i]));
    else Speed[i] = 0;
}

HybridLocalSearchDirective directive = new HybridLocalSearchDirective();

SolverContext context = SolverContext.GetContext();
directive.TimeLimit = 10000;

double SpecificWeight = 7800; // kg/m^3

double FurnCapacity;
FurnCapacity = 34430000; // kKal * m / saat

double Cp;
Cp = 0.1170; // kilo Kalori / kg °C

int FurnLength;
FurnLength = 570;

Model model = context.CreateModel();

if (Count == 2 )
{
    // Create min max temperatures variables

```

```

        d0Max = 60;
        d1Max = 60;
        if (strClassData.ANNCODE[0].Trim(' ') == "CQ") { x0min = 690;
x0max = 750; }
        if (strClassData.ANNCODE[0].Trim(' ') == "DQ1") { x0min = 740;
x0max = 800; }
        if (strClassData.ANNCODE[0].Trim(' ') == "DQ2") { x0min = 750;
x0max = 810; }
        if (strClassData.ANNCODE[0].Trim(' ') == "DDQ") { x0min = 770;
x0max = 830; }
        if (strClassData.ANNCODE[0].Trim(' ') == "EDDQ") { x0min = 830;
x0max = 870; doMax = 40}

        if (strClassData.ANNCODE[1].Trim(' ') == "CQ") { x1min = 690;
x1max = 750; }
        if (strClassData.ANNCODE[1].Trim(' ') == "DQ1") { x1min = 740;
x1max = 800; }
        if (strClassData.ANNCODE[1].Trim(' ') == "DQ2") { x1min = 750;
x1max = 810; }
        if (strClassData.ANNCODE[1].Trim(' ') == "DDQ") { x1min = 770;
x1max = 830; }
        if (strClassData.ANNCODE[1].Trim(' ') == "EDDQ") { x1min = 830;
x1max = 870; d1Max = 40 }

        Decision d0 = new Decision(Domain.IntegerRange(0, d0Max), "d0");
        Decision d1 = new Decision(Domain.IntegerRange(0, d1Max), "d1");
        model.AddDecisions(d0, d1 );

        model.AddConstraint("Row0", Cp * SpecificWeight * Convert.ToDouble(strClassData.TARWID[0]) * Convert.ToDouble(strClassData.ACTTHICK[0]) *
Speed[0] * (x1 - x0) * FurnLength <= FurnCapacity);

        model.AddConstraint("Row10", x0min <= x0);
        model.AddConstraint("Row11", x0 <= x0max);
        model.AddConstraint("Row12", x1min <= x1);
        model.AddConstraint("Row13", x1 <= x1max);
        model.AddConstraint("Row14", 0 <= d0);
        model.AddConstraint("Row15", d0 <= d0Max);
        model.AddConstraint("Row14", 0 <= d1);
        model.AddConstraint("Row15", d1 <= d1Max);
        model.AddConstraint("Row16", x0 + d0 = x0min);
        model.AddConstraint("Row17", x1 + d1 = x1min);

        Goal goal = model.AddGoal("Goal0", GoalKind.Minimize, Model.Sum(d0, d1));

        Solution solution = context.Solve(directive);

        string[] LogText = new string[14];

        LogText[0] = " Bobin Numarası " + " Hedef Hız " + " Kalite Kodu "
+ " Kalınlık " + " Genişlik " + " Hedef Sıcaklık ";

        foreach (object[] value in d0.GetValues())
        {

```

```

        LogText[1] = " " + String.Format("{0:#####}",
strClassData.COILNO[0]) + " " + Speed[0] + " " + strClassData.ANN-
CODE[0] + " " + strClassData.ACTTHICK[0] + " " + strClass-
Data.TARWID[0] + " " + d0 + value[0].ToString();
    }

    foreach (object[] value in d1.GetValues())
    {
        LogText[2] = " " + strClassData.COILNO[1] + " " +
Speed[1] + " " + strClassData.ANNCODE[1] + " " + strClass-
Data.ACTTHICK[1] + " " + d1 + strClassData.TARWID[1] + " " + va-
lue[0].ToString();
    }

    LogText[Count + 1] = goal.ToInt32().ToString();
    LogText[Count + 2] = " ";
    LogText[Count + 3] = " ";

    if (File.Exists(Filename)) File.Delete(Filename);

    System.IO.File.WriteAllLines(Filename, LogText);

    Console.WriteLine("Döngü bitti");
    Console.ReadKey();
}
else if (Count == 3)
{
    .....
}
else if (Count == 4)
{
    .....
}
else if (Count == 5)
{
    .....
}
else if (Count == 6)
{
    .....
}
else if (Count == 7)
{
    .....
}
else if (Count == 8)
{
    .....
}
else if (Count == 9)
{
    .....
}
else if (Count == 10)
{
    .....
}
}
private DataSet ReadExcelFile(string AnnealCode)
{
    DataSet ds = new DataSet();

```

```

        string connectionString = GetConnectionString();

        using (OleDbConnection conn = new OleDbConnection(connectionString))
        {
            conn.Open();
            OleDbCommand cmd = new OleDbCommand();
            cmd.Connection = conn;

            // Get all Sheets in Excel File
            DataTable dtSheet = conn.GetOleDbSchemaTable(OleDbSchemaGuid.Tab-
les, null);

            AnnealCode = AnnealCode.TrimEnd(' ') + @"$" ;

            // Get all rows from the Sheet
            cmd.CommandText = "SELECT * FROM [" + AnnealCode + "];";

            DataTable dt = new DataTable();
            dt.TableName = AnnealCode;

            OleDbDataAdapter da = new OleDbDataAdapter(cmd);
            da.Fill(dt);

            ds.Tables.Add(dt);
            //}

            cmd = null;
            conn.Close();
        }

        return ds;
    }

    private string GetConnectionString()
    {
        Dictionary<string, string> props = new Dictionary<string, string>();

        // XLSX - Excel 2007, 2010, 2012, 2013
        props["Provider"] = "Microsoft.ACE.OLEDB.12.0;";
        props["Extended Properties"] = "Excel 12.0 XML";
        props["Data Source"] = "E:\\SAU_DR\\TargetLineSpeeds.xlsx";

        StringBuilder sb = new StringBuilder();

        foreach (KeyValuePair<string, string> prop in props)
        {
            sb.Append(prop.Key);
            sb.Append('=');
            sb.Append(prop.Value);
            sb.Append(';');
        }

        return sb.ToString();
    }

    private Int16 ReadSpeedFromExcel(string AnnealCode, Int16 Thickness,
Int16 Width)
    {
        Int16 Speed = 0 ;
        System.Data.DataSet DtSet;
        DtSet = ReadExcelFile(AnnealCode);
    }

```

```

int ColumnCount = 0, RowCount = 0;
int Temp = 0;

for (int i = 1; i < DtSet.Tables[0].Columns.Count; i++)
{
    Temp = Convert.ToInt16(DtSet.Tables[0].Rows[0][i]);
    if (Width < Temp)
    { ColumnCount = i; break; }
    else ColumnCount = DtSet.Tables[0].Columns.Count - 1;
}

for (int j = 1; j < DtSet.Tables[0].Rows.Count; j++)
{
    if (Thickness < Convert.ToInt16(DtSet.Tables[0].Rows[j][0]))
    { RowCount = j; break; }
    else RowCount = DtSet.Tables[0].Rows.Count - 1;
}

Speed = Convert.ToInt16(DtSet.Tables[0].Rows[RowCount][ColumnCount]);
return Speed;
}
}

```

EK-6: Uzunluk normalize programı (Visual Basic Macro)

```
Sub Length_Calc()  
    Dim LastRow As Long  
    Dim MaxValue As Long  
    Dim j As Long  
  
    LastRow = Range("A1").End(xlDown).Row  
    For i = 3 To LastRow - 1  
        If (StrComp(Cells(i, 1), Cells(i + 1, 1), vbBinaryCompare) = 0) Then  
            j = j + 1  
            If (Cells(i, 17) < Cells(i + 1, 17)) Then MaxValue = Cells(i + 1, 17)  
        Else  
            For k = i - j To i  
                Cells(k, 19) = Cells(k, 17) / MaxValue  
            Next  
            j = 0  
        End If  
    Next  
End Sub
```

EK-7: Şerit uzunluklarının baştan veya sondan %50'lik kısmını alan program

```
using System;
using System.IO;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Data;
using System.Data.OleDb;
using Excel = Microsoft.Office.Interop.Excel;

namespace ConsoleApplication4
{
    class Program
    {
        static Excel.Application ExcelApp = null;
        static Excel.Workbook ExcelWorkBook = null;

        static Excel.Worksheet MySheet = null;
        static string crFile = "C:\\csharp-Excel.xls";

        static object misValue = System.Reflection.Missing.Value;

        static OleDbCommand cmd = new OleDbCommand();

        static OleDbConnection conn;

        static void Main(string[] args)
        {
            ExcelApp = new Excel.Application();
            ExcelApp.Visible = false;

            ExcelWorkBook = ExcelApp.Workbooks.Add(misValue);
            MySheet = (Excel.Worksheet)ExcelWorkBook.Worksheets.get_Item(1);

            DataSet DtSet = new DataSet();

            string connectionString = GetConnectionString();
            //try
            //{
                using (conn = new OleDbConnection(connectionString))
                {
                    conn.Open();

                    cmd.Connection = conn;

                    // Get all Sheets in Excel File
                    DataTable dtSheet = conn.GetOleDbSchemaTable(OleDbSchema-
Guid.Tables, null);
```

```

string AnnealCode = @"Sayfa1$";

cmd.CommandText = "SELECT * FROM [" + AnnealCode + "]";

DataTable dt = new DataTable();
dt.TableName = AnnealCode;
OleDbDataAdapter da = new OleDbDataAdapter(cmd);
da.Fill(dt);
DtSet.Tables.Add(dt);
double Temp1;
int RowCount = 1;

for (int j = 1; j < DtSet.Tables[0].Rows.Count; j++)
{
    if (!string.IsNullOrEmpty(DtSet.Tables[0].Rows[j][5].ToString()))
        Temp1 = Convert.ToDouble(DtSet.Tables[0].Rows[j][5]);
    else
        Temp1 = 0.0;

    if (Temp1 >= 0.50 && Temp1 <= 1.0)
    {
        for (int i = 0; i < DtSet.Tables[0].Columns.Count; i++)
        {
            if (!string.IsNullOrEmpty(DtSet.Tables[0].Rows[j][i].ToString()))
            {
                if (i == 0)
                    MySheet.Cells[RowCount, i + 1] = DtSet.Tables[0].Rows[j][i].ToString();
                else
                    MySheet.Cells[RowCount, i + 1] = Math.Round(Convert.ToDouble(DtSet.Tables[0].Rows[j][i]), 3);
            }
            else
            {
                Console.WriteLine("Hatalı satır = " + i.ToString() + " " + j.ToString());
                MySheet.Cells[RowCount, i] = "0.000";
            }
        }
        ++RowCount;
    }
}

if (File.Exists(crFile)) File.Delete(crFile);
ExcelWorkBook.SaveAs(crFile, Excel.XlFileFormat.xlWorkbookNormal,
misValue, misValue, misValue, misValue, Excel.XlSaveAsAccessMode.xlExclusive,
misValue, misValue, misValue, misValue, misValue);
ExcelWorkBook.Close(true, misValue, misValue);

```

```

        ExcelApp.Quit();

        releaseObject(ExcelApp);
        releaseObject(ExcelWorkBook);
        releaseObject(ExcelApp);
        cmd = null;
        conn.Close();
    }
}
static void releaseObject(object obj)
{
    try
    {
        System.Runtime.InteropServices.Marshal.ReleaseComObject(obj);
        obj = null;
    }
    catch (Exception ex)
    {
        obj = null;
        Console.WriteLine("Exception Occured while releasing object " + ex.ToString());
    }
    finally
    {
        GC.Collect();
    }
}
static string GetConnectionString()
{
    Dictionary<string, string> props = new Dictionary<string, string>();

    props["Provider"] = "Microsoft.ACE.OLEDB.12.0;";
    props["Extended Properties"] = "Excel 12.0 XML";
    props["Data Source"] = @"E:\SAU_DR\SAU-DR\RTHNormalized.xlsm";

    StringBuilder sb = new StringBuilder();
    foreach (KeyValuePair<string, string> prop in props)
    {
        sb.Append(prop.Key);
        sb.Append('=');
        sb.Append(prop.Value);
        sb.Append(';');
    }

    return sb.ToString();
}
}
}

```

EK-8: Peş peşe devam eden aynı tip bobinleri silen program

```
using System;
using System.IO;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Data;
using System.Data.OleDb;
using Excel = Microsoft.Office.Interop.Excel;

namespace ConsoleApplication4
{
    class Program
    {
        static Excel.Application ExcelApp = null;
        static Excel.Workbook ExcelWorkBook = null;

        static Excel.Worksheet MySheet = null;
        static string crFile = "C:\\csharp-Excel.xls";

        static object misValue = System.Reflection.Missing.Value;

        static OleDbCommand cmd = new OleDbCommand();

        static OleDbConnection conn;

        static void Main(string[] args)
        {
            ExcelApp = new Excel.Application();
            ExcelApp.Visible = false;

            ExcelWorkBook = ExcelApp.Workbooks.Add(misValue);
            MySheet = (Excel.Worksheet)ExcelWorkBook.Worksheets.get_Item(1);

            DataSet DtSet = new DataSet();

            string connectionString = GetConnectionString();

            using (conn = new OleDbConnection(connectionString))
            {
                conn.Open();

                cmd.Connection = conn;

                // Get all Sheets in Excel File
                DataTable dtSheet = conn.GetOleDbSchemaTable(OleDbSchema-
                    Guid.Tables, null);
            }
        }
    }
}
```

```

string AnnealCode = @"Sayfa1$";

// Get all rows from the Sheet
cmd.CommandText = "SELECT * FROM [" + AnnealCode + "]";

DataTable dt = new DataTable();
dt.TableName = AnnealCode;

OleDbDataAdapter da = new OleDbDataAdapter(cmd);
da.Fill(dt);

DtSet.Tables.Add(dt);

double Temp1;
int RowCount = 1;

for (int j = 0 ; j < DtSet.Tables[0].Rows.Count-1; j++)
{
    //if (DtSet.Tables[0].Rows[j][0].ToString() == DtSet.Tables[0].Rows[j +
1][0].ToString() &&
        if( Convert.ToDouble(DtSet.Tables[0].Rows[j][2]) == Convert.ToDo-
uble(DtSet.Tables[0].Rows[j + 1][2]) &&
            Convert.ToDouble(DtSet.Tables[0].Rows[j][3]) == Convert.ToDo-
uble(DtSet.Tables[0].Rows[j + 1][3]) &&
                Convert.ToDouble(DtSet.Tables[0].Rows[j][4]) == Convert.ToDo-
uble(DtSet.Tables[0].Rows[j + 1][4]) )
            {
                for (int i = 0; i < DtSet.Tables[0].Columns.Count; i++)
                {
                    MySheet.Cells[RowCount, i+1] = Math.Round
(Convert.ToDouble(DtSet.Tables[0].Rows[j][i]),3);
                }
                ++RowCount;
            }
        else
        {
            //Console.WriteLine("Döngüye girdi = " + DtSet.Tab-
les[0].Rows[j][0]);
            do
            {
                j++;
                if (j == 64168 ) break; //DtSet.Tables[0].Columns.Count - 2)
            } while (DtSet.Tables[0].Rows[j][0] == DtSet.Tables[0].Rows[j
+ 1][0]);
            j--;
        }
    }
}

```

```

        if (File.Exists(crFile)) File.Delete(crFile);

        ExcelWorkBook.SaveAs(crFile, Excel.XlFileFormat.xlWorkbookNormal,
misValue, misValue, misValue, misValue, Excel.XlSaveAsAccessMode.xlExclusive,
misValue, misValue, misValue, misValue, misValue);
        ExcelWorkBook.Close(true, misValue, misValue);
        ExcelApp.Quit();

        releaseObject(ExcelApp);
        releaseObject(ExcelWorkBook);
        releaseObject(ExcelApp);
        cmd = null;
        conn.Close();
    }
}
static void releaseObject(object obj)
{
    try
    {
        System.Runtime.InteropServices.Marshal.ReleaseComObject(obj);
        obj = null;
    }
    catch (Exception ex)
    {
        obj = null;
        Console.WriteLine("Exception Occured while releasing object " + ex.ToSt-
ring());
    }
    finally
    {
        GC.Collect();
    }
}

static string GetConnectionString()
{
    Dictionary<string, string> props = new Dictionary<string, string>();

    // XLSX - Excel 2007, 2010, 2012, 2013
    props["Provider"] = "Microsoft.ACE.OLEDB.12.0;";
    props["Extended Properties"] = "Excel 12.0 XML";
    props["Data Source"] = @"E:\SAU_DR\SAU-DR\Tez\Tez Öneri ve İzleme Ra-
por\Programlar\Normalizedplus15percentage.xls";

    StringBuilder sb = new StringBuilder();

    foreach (KeyValuePair<string, string> prop in props)
    {

```

```
        sb.Append(prop.Key);  
        sb.Append('=');  
        sb.Append(prop.Value);  
        sb.Append(';');  
    }  
  
    return sb.ToString();  
}  
  
}  
}
```



EK-9: Matlab YSA Eğitim Programı

```
clear all;
NNinput = xlsread ( 'C:\NormalizedValues. xlsx','Sayfa1','B1:F50000' );
NNoutput = xlsread ( 'C:\ NormalizedValues. xlsx','Sayfa1','G1:N50000' );
NNsampleInp = xlsread ( 'C:\ NormalizedValues. xlsx','Sayfa1','B50001:F50250' );
NNsampleOut = xlsread('C:\ NormalizedValues. xlsx','Sayfa1','G50001:N50250' );
NNoutput=transpose(NNoutput);
NNinput=transpose(NNinput);
NNsampleInp=transpose(NNsampleInp);
NNsampleOut=transpose(NNsampleOut);
netFF. trainParam. goal = 0,00001;
netFF = newff(NNinput, NNoutput,[90],{'tansig'}, 'trainscg');
netFF. performFcn = 'mse';
netFF = train(netFF, NNinput, NNoutput);
save('c:\netFF');
NNtestOut = sim(netFF, NNsampleInp);
perfnetFF = perform(netFF, NNtestOut, NNsampleOut);
```

EK-10: Yapay Sinir Ağı Deneme Sonuçları

[453] – tansig – trainlm

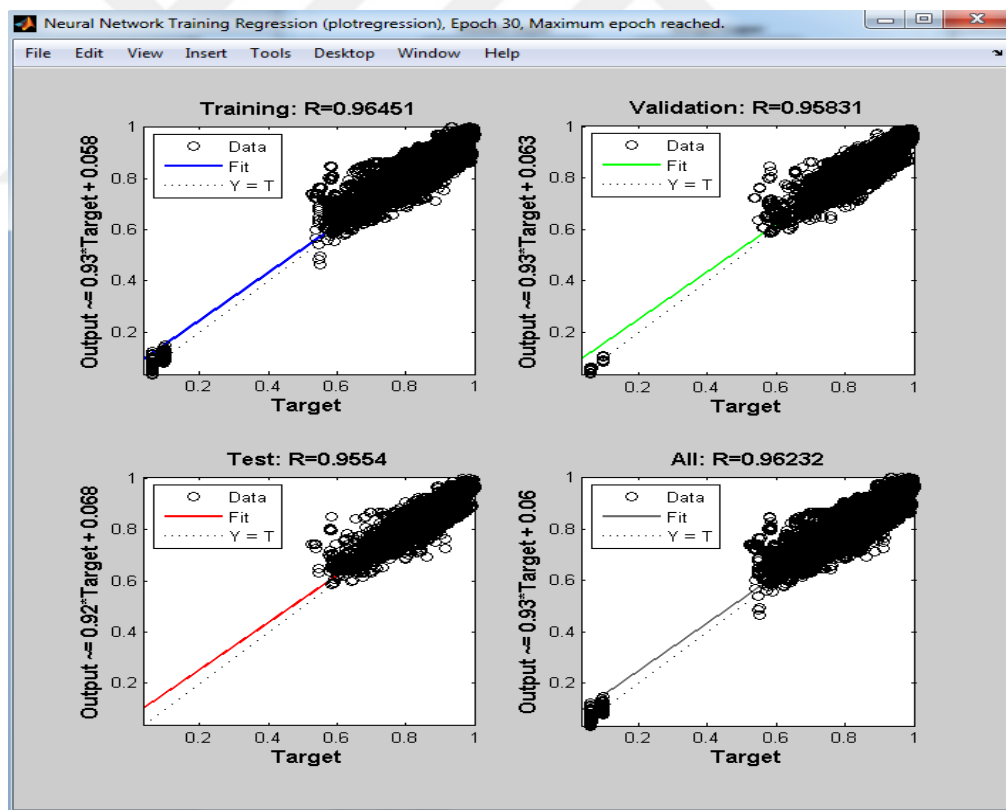
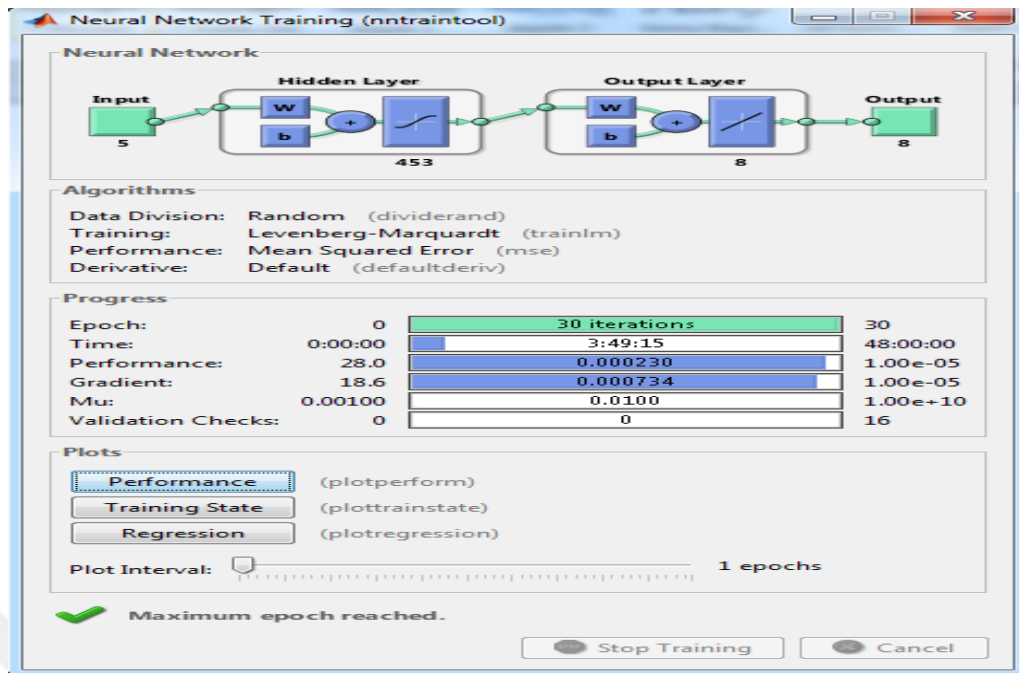
Uygulanan matlab program aşağıdaki gibidir. Nöron hesabı formülüne göre $\alpha = 10$ seçilerek nöron sayısı 453 bulunmuştur.

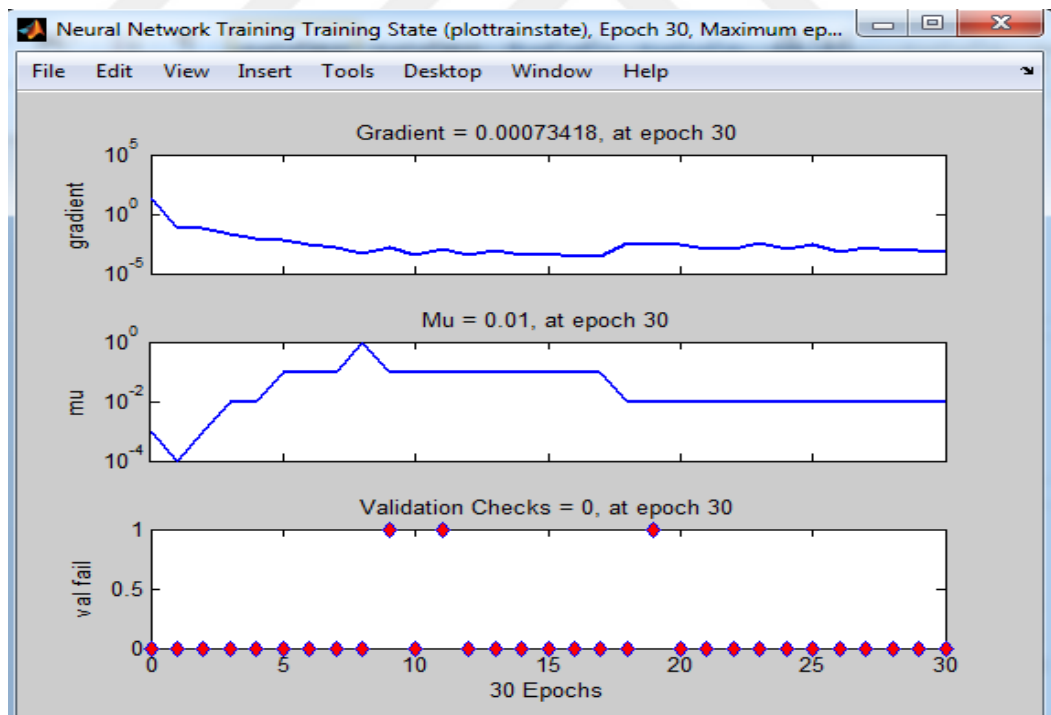
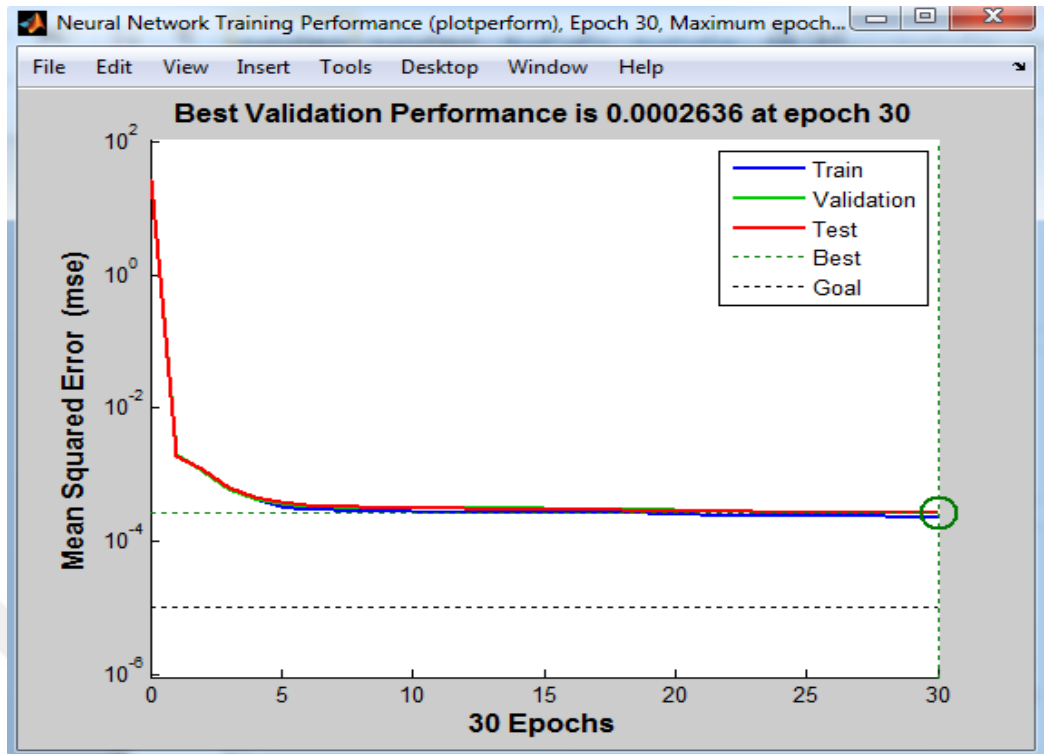
Matlab Program:

```
clear all;
NNinput = xlsread('C:\Normalized15perdoubrecdeleted.xlsx','Sayfa1','B1:F62087');
NNoutput = xlsread('C:\Normalized15perdoubrecdeleted.xlsx','Sayfa1','G1:N62087');
NNsampleInp = xlsread('C:\Normalized15perdoubrecdeleted.xlsx','Sayfa1','B62087:F62133');
NNsampleOut = xlsread('C:\Normalized15perdoubrecdeleted.xlsx','Sayfa1','G62087:N62133');
NNoutput=transpose(NNoutput);
NNinput=transpose(NNinput);
NNsampleInp=transpose(NNsampleInp);
NNsampleOut=transpose(NNsampleOut);
netFF = newff(NNinput,NNoutput,453,{'tansig'},'trainlm');
netFF.performFcn = 'mse';
netFF.trainParam.show = 1;
netFF.trainParam.epochs = 30;
netFF.trainParam.goal = 1e-5;
netFF.trainParam.max_fail=16;
netFF.trainParam.time = 172800;
netFF.encyency.memoryReduction = 6;
netFF = train(netFF,NNinput,NNoutput);
save('c:\netFF-2270');
NNtestOut = sim(netFF,NNsampleInp);
perfnetFF = perform(netFF,NNtestOut,NNsampleOut);
Performans değerimiz ise;
```

$$\textbf{Performans (mse) = } \underline{\underline{2,39 \cdot 10^{-4}}}$$

olarak bulunmuştur.





[453] Tansig – Trainlm Gerçek Fırın Sıcaklıkları							
768,6	814,2	795,2	823,7	802,8	828,4	827,5	817,0
772,4	815,1	796,1	823,7	801,8	827,5	824,6	816,1
772,4	816,1	796,1	823,7	801,8	827,5	824,6	816,1
772,4	816,1	796,1	823,7	801,8	827,5	824,6	816,1
772,4	816,1	796,1	823,7	801,8	827,5	824,6	816,1
771,4	816,1	796,1	823,7	801,8	827,5	824,6	816,1
771,4	816,1	796,1	823,7	801,8	827,5	824,6	816,1
771,4	816,1	796,1	823,7	801,8	827,5	824,6	816,1
771,4	816,1	796,1	823,7	801,8	827,5	824,6	816,1
726,8	766,7	756,2	781,9	770,5	794,2	797,1	793,3
726,8	765,7	755,3	780,9	768,6	793,3	796,1	793,3
725,8	764,8	755,3	780,0	768,6	792,3	795,2	792,3
724,9	764,8	754,3	779,0	767,6	792,3	795,2	792,3
723,9	763,8	753,4	778,1	766,7	791,4	794,2	791,4
723,0	762,9	753,4	778,1	766,7	791,4	794,2	790,4
723,0	761,9	752,4	777,1	765,7	790,4	793,3	790,4
722,0	761,0	751,5	776,2	765,7	789,5	793,3	790,4
801,8	842,7	825,6	859,8	830,3	859,8	859,8	856,0
801,8	842,7	825,6	858,8	830,3	858,8	859,8	855,0
801,8	842,7	825,6	858,8	830,3	858,8	859,8	855,0
800,9	843,6	825,6	858,8	829,4	858,8	858,8	855,0
800,9	843,6	825,6	858,8	829,4	857,9	858,8	854,1
761,9	801,8	797,1	825,6	803,7	830,3	832,2	828,4
761,0	800,9	796,1	825,6	803,7	830,3	832,2	828,4
760,0	799,0	795,2	824,6	802,8	830,3	831,3	827,5
759,1	799,0	794,2	824,6	802,8	829,4	831,3	827,5
759,1	798,0	794,2	823,7	802,8	829,4	831,3	825,6
758,1	797,1	793,3	823,7	801,8	828,4	830,3	825,6
758,1	797,1	793,3	822,7	801,8	828,4	830,3	825,6
757,2	796,1	792,3	821,8	801,8	828,4	830,3	825,6
756,2	795,2	791,4	821,8	801,8	828,4	829,4	824,6
763,8	806,6	790,4	818,9	800,9	825,6	824,6	819,9
763,8	806,6	790,4	818,9	800,9	825,6	824,6	818,9
763,8	806,6	790,4	818,9	800,9	824,6	824,6	818,9
762,9	806,6	790,4	818,9	799,9	825,6	824,6	818,9
762,9	806,6	789,5	818,9	799,9	824,6	824,6	818,9
761,9	805,6	790,4	818,0	799,9	824,6	824,6	818,9
761,9	805,6	789,5	818,0	799,9	824,6	823,7	818,9
775,2	804,7	804,7	820,8	818,0	834,1	837,9	836,0

[453] Tansig – Trainlm Hesaplanan Fırın Sıcaklıkları							
768,1	810,4	796,3	816,6	807,9	826,5	825,6	815,2
772,6	810,9	798,2	812,4	811,2	830,4	829,5	818,6
770,0	810,8	796,9	813,9	808,8	828,4	827,1	817,1
771,2	810,9	797,1	814,1	810,5	829,8	828,3	818,0
772,5	812,1	798,0	813,7	811,2	831,5	830,0	819,5
770,9	812,0	797,0	813,1	810,1	829,7	828,6	818,7
770,5	810,9	795,0	811,7	809,1	828,1	827,1	817,1
773,0	813,4	797,1	814,6	809,0	829,3	827,7	816,7
768,1	810,4	796,3	816,6	807,9	826,5	825,6	815,2
729,8	765,8	764,6	778,3	786,4	801,9	807,3	800,3
733,2	769,3	766,7	780,2	787,2	802,6	808,0	800,6
733,5	769,5	767,1	779,6	787,1	801,6	807,0	799,7
734,4	768,5	767,3	779,6	787,8	802,1	807,7	800,3
735,9	769,4	768,6	781,5	789,1	803,4	808,9	801,1
736,7	769,8	768,7	782,0	788,2	803,5	808,9	800,5
736,4	768,6	768,3	779,8	787,2	802,7	808,7	800,4
733,3	765,9	767,0	776,8	787,2	801,4	807,2	800,3
799,8	836,2	830,4	847,6	832,7	851,9	850,7	839,6
796,5	834,7	828,6	844,2	831,6	849,8	848,8	838,0
795,8	833,8	828,5	844,3	833,2	849,4	848,2	837,8
798,5	836,8	828,8	845,8	832,7	850,6	849,0	837,7
796,5	835,1	831,4	844,1	834,7	850,4	848,6	837,9
742,0	778,7	785,0	805,0	807,8	822,7	823,2	818,0
749,1	787,5	791,1	811,3	813,6	830,4	830,6	826,5
752,6	791,6	793,9	815,5	815,0	834,4	834,6	830,3
753,2	790,1	794,6	816,4	814,2	834,7	835,6	831,0
750,2	785,0	792,2	813,2	811,7	831,9	834,0	828,9
755,2	790,3	796,4	817,3	813,3	832,4	834,6	828,4
756,6	794,4	798,7	818,6	813,0	832,6	833,4	827,2
756,5	796,1	799,7	821,0	813,2	833,8	833,7	828,3
757,9	796,7	800,0	823,6	814,3	835,6	835,7	830,1
772,6	810,9	798,2	812,4	811,2	830,4	829,5	818,6
770,0	810,8	796,9	813,9	808,8	828,4	827,1	817,1
771,2	810,9	797,1	814,1	810,5	829,8	828,3	818,0
772,5	812,1	798,0	813,7	811,2	831,5	830,0	819,5
770,9	812,0	797,0	813,1	810,1	829,7	828,6	818,7
770,5	810,9	795,0	811,7	809,1	828,1	827,1	817,1
773,0	813,4	797,1	814,6	809,0	829,3	827,7	816,7
744,2	771,7	776,7	786,1	793,3	812,6	814,4	809,2

[453] Tansig – Trainlm Fark = Hesaplanan - Gerçek							
0,4	3,8	-1,2	7,1	-5,2	1,9	1,8	1,8
-0,2	4,2	-2,1	11,3	-9,4	-2,9	-4,9	-2,6
2,4	5,3	-0,8	9,8	-7,0	-0,9	-2,5	-1,0
1,1	5,2	-1,0	9,5	-8,7	-2,3	-3,7	-2,0
-0,1	4,0	-1,9	9,9	-9,4	-4,1	-5,4	-3,5
0,5	4,1	-0,9	10,5	-8,3	-2,2	-4,0	-2,7
0,9	5,2	1,1	12,0	-7,3	-0,6	-2,5	-1,0
-1,6	2,6	-1,0	9,0	-7,2	-1,9	-3,1	-0,6
3,3	5,7	-0,2	7,1	-6,1	1,0	-1,0	0,9
-3,1	0,8	-8,4	3,6	-15,9	-7,7	-10,2	-7,1
-6,4	-3,6	-11,4	0,7	-18,7	-9,4	-11,9	-7,3
-7,7	-4,7	-11,8	0,3	-18,5	-9,3	-11,9	-7,4
-9,5	-3,7	-13,0	-0,6	-20,2	-9,8	-12,5	-8,0
-12,0	-5,6	-15,3	-3,4	-22,4	-12,1	-14,7	-9,8
-13,8	-7,0	-15,3	-4,0	-21,6	-12,2	-14,7	10,1
-13,5	-6,7	-15,9	-2,7	-21,5	-12,3	-15,4	-10,0
-11,3	-4,9	-15,6	-0,6	-21,5	-11,9	-14,0	-9,9
2,0	6,4	-4,9	12,2	-2,4	7,9	9,0	16,3
5,3	8,0	-3,1	14,6	-1,3	9,0	11,0	17,0
6,0	8,9	-3,0	14,5	-2,9	9,4	11,6	17,2
2,3	6,8	-3,2	13,0	-3,4	8,2	9,8	17,3
4,3	8,5	-5,9	14,7	-5,4	7,5	10,2	16,1
19,9	23,1	12,1	20,5	-4,1	7,6	9,0	10,4
11,8	13,3	5,0	14,3	-9,9	-0,1	1,6	1,9
7,4	7,4	1,3	9,1	-12,3	-4,1	-3,3	-2,9
5,9	8,9	-0,4	8,2	-11,5	-5,3	-4,4	-3,5
8,8	13,0	2,0	10,4	-8,9	-2,5	-2,8	-3,3
2,9	6,8	-3,1	6,3	-11,5	-4,0	-4,3	-2,8
1,5	2,6	-5,4	4,1	-11,2	-4,2	-3,1	-1,7
0,7	0,0	-7,4	0,8	-11,4	-5,4	-3,4	-2,7
-1,7	-1,6	-8,7	-1,9	-12,5	-7,2	-6,3	-5,5
-8,8	-4,4	-7,8	6,5	-10,4	-4,8	-4,9	1,2
-6,2	-4,2	-6,5	5,0	-8,0	-2,8	-2,5	1,8
-7,4	-4,3	-6,7	4,8	-9,7	-5,2	-3,7	0,9
-9,6	-5,5	-7,6	5,2	-11,3	-6,0	-5,4	-0,6
-8,1	-5,4	-7,5	5,8	-10,2	-5,1	-4,0	0,2
-8,6	-5,3	-4,6	6,3	-9,2	-3,5	-2,5	1,8
-11,1	-7,8	-7,6	3,3	-9,1	-4,7	-4,0	2,2
31,0	33,0	27,9	34,7	24,7	21,5	23,5	26,8
28,0	33,9	25,2	33,5	23,8	22,3	21,6	24,7
26,8	34,3	25,1	33,2	22,4	20,9	19,6	22,3
-36,8	-30,2	-36,2	-31,1	-50,6	-30,0	-28,1	-27,0

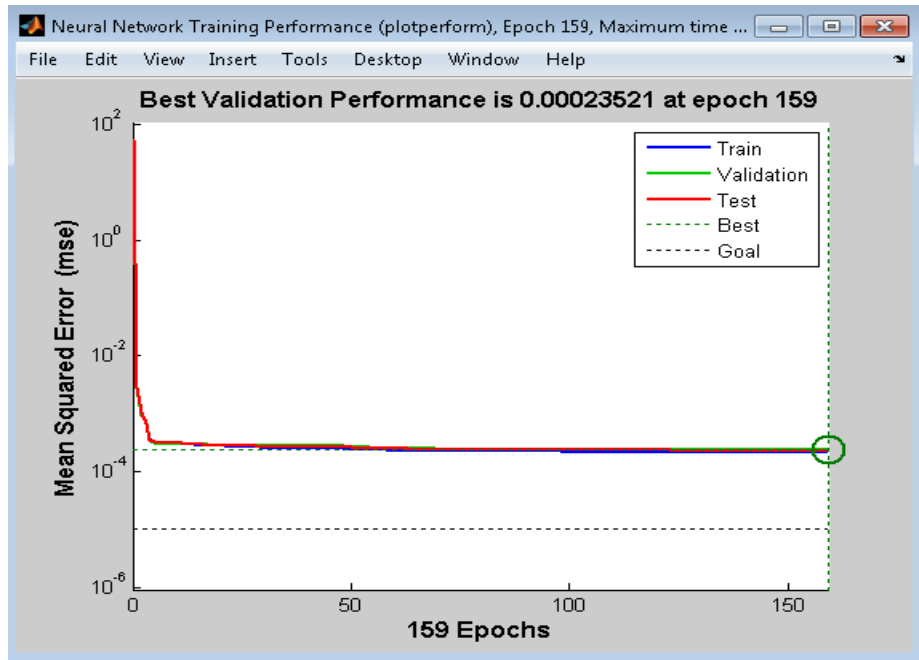
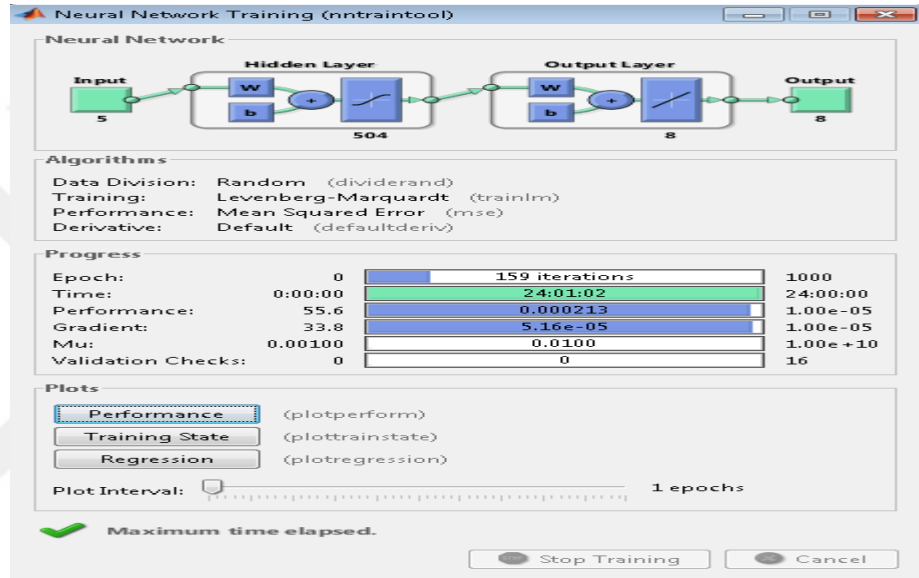
[[504] – tansig – trainlm

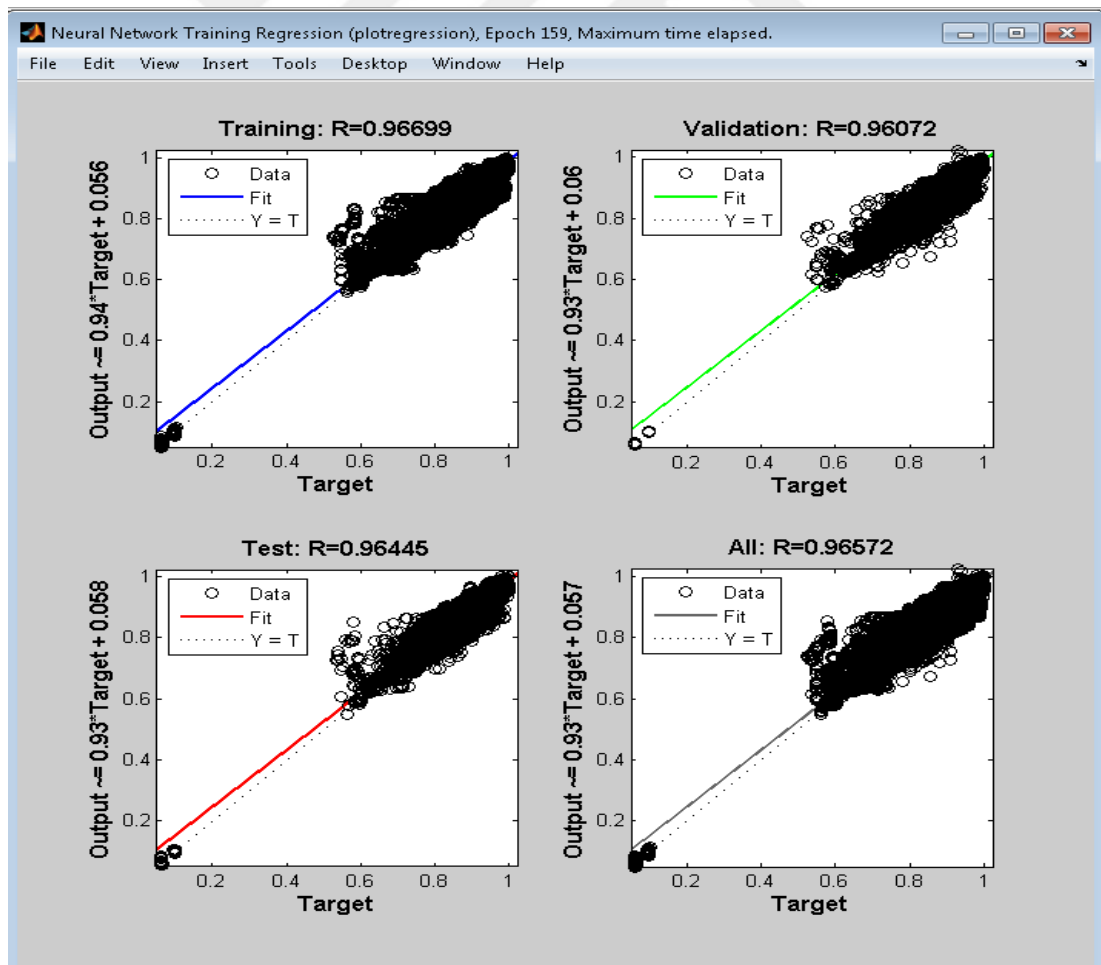
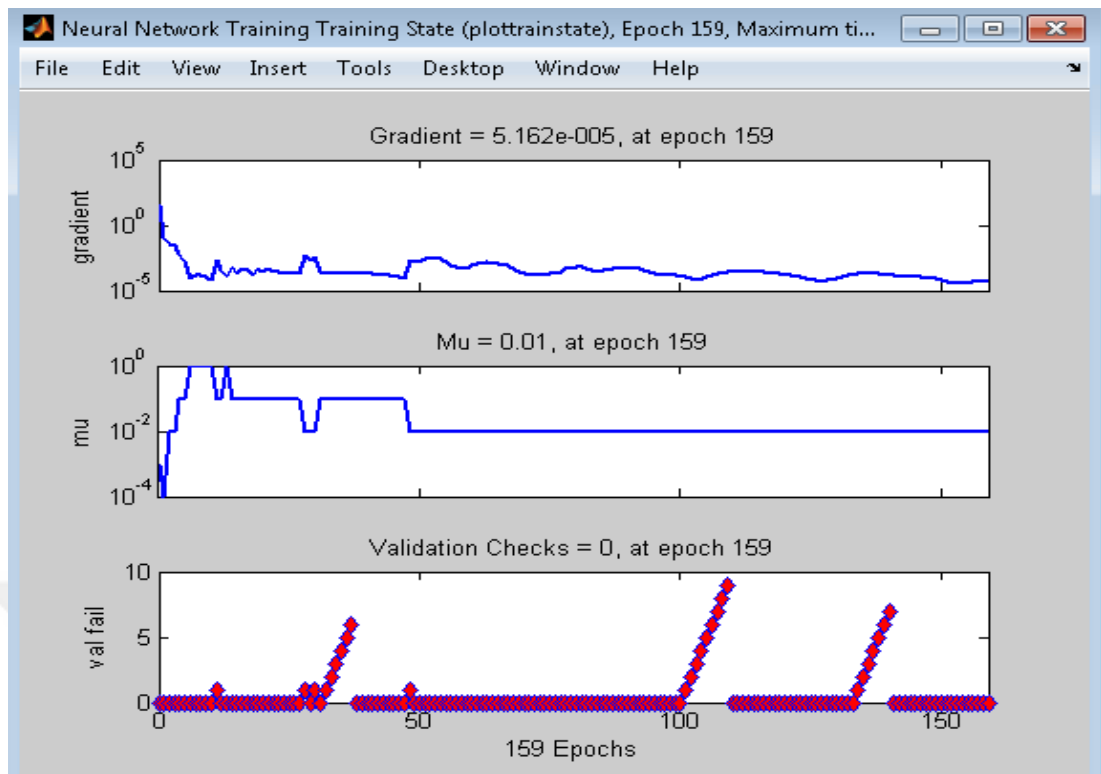
Uygulanan matlab program aşağıdaki gibidir. Nöron hesabı formülüne göre $\alpha = 9$ seçilerek nöron sayısı 504 bulunmuştur. Bir üstteki program ile aynı yapıdadır. Sadece nöron sayısı farklıdır.

Performans değerimiz ise;

$$\text{Performans (mse)} = 2.18 \cdot 10^{-4}$$

olarak bulunmuştur.





[504] tansig – trainlm Gerçek Fırın Sıcaklıkları							
768,6	814,2	795,2	823,7	802,8	828,4	827,5	817,0
772,4	815,1	796,1	823,7	801,8	827,5	824,6	816,1
772,4	816,1	796,1	823,7	801,8	827,5	824,6	816,1
772,4	816,1	796,1	823,7	801,8	827,5	824,6	816,1
772,4	816,1	796,1	823,7	801,8	827,5	824,6	816,1
771,4	816,1	796,1	823,7	801,8	827,5	824,6	816,1
771,4	816,1	796,1	823,7	801,8	827,5	824,6	816,1
771,4	816,1	796,1	823,7	801,8	827,5	824,6	816,1
771,4	816,1	796,1	823,7	801,8	827,5	824,6	816,1
726,8	766,7	756,2	781,9	770,5	794,2	797,1	793,3
726,8	765,7	755,3	780,9	768,6	793,3	796,1	793,3
725,8	764,8	755,3	780,0	768,6	792,3	795,2	792,3
724,9	764,8	754,3	779,0	767,6	792,3	795,2	792,3
723,9	763,8	753,4	778,1	766,7	791,4	794,2	791,4
723,0	762,9	753,4	778,1	766,7	791,4	794,2	790,4
723,0	761,9	752,4	777,1	765,7	790,4	793,3	790,4
722,0	761,0	751,5	776,2	765,7	789,5	793,3	790,4
801,8	842,7	825,6	859,8	830,3	859,8	859,8	856,0
801,8	842,7	825,6	858,8	830,3	858,8	859,8	855,0
801,8	842,7	825,6	858,8	830,3	858,8	859,8	855,0
800,9	843,6	825,6	858,8	829,4	858,8	858,8	855,0
800,9	843,6	825,6	858,8	829,4	857,9	858,8	854,1
761,9	801,8	797,1	825,6	803,7	830,3	832,2	828,4
761,0	800,9	796,1	825,6	803,7	830,3	832,2	828,4
760,0	799,0	795,2	824,6	802,8	830,3	831,3	827,5
759,1	799,0	794,2	824,6	802,8	829,4	831,3	827,5
759,1	798,0	794,2	823,7	802,8	829,4	831,3	825,6
758,1	797,1	793,3	823,7	801,8	828,4	830,3	825,6
758,1	797,1	793,3	822,7	801,8	828,4	830,3	825,6
757,2	796,1	792,3	821,8	801,8	828,4	830,3	825,6
756,2	795,2	791,4	821,8	801,8	828,4	829,4	824,6
763,8	806,6	790,4	818,9	800,9	825,6	824,6	819,9
763,8	806,6	790,4	818,9	800,9	825,6	824,6	818,9
763,8	806,6	790,4	818,9	800,9	824,6	824,6	818,9
762,9	806,6	790,4	818,9	799,9	825,6	824,6	818,9
762,9	806,6	789,5	818,9	799,9	824,6	824,6	818,9
761,9	805,6	790,4	818,0	799,9	824,6	824,6	818,9
761,9	805,6	789,5	818,0	799,9	824,6	823,7	818,9
775,2	804,7	804,7	820,8	818,0	834,1	837,9	836,0

[504] tansig – trainlm Hesaplanan Fırın Sıcaklıkları							
775,0	813,5	793,5	820,4	806,2	831,6	831,2	818,7
777,9	813,9	795,9	822,5	807,2	835,6	832,5	819,7
771,7	811,1	793,3	818,5	805,9	833,5	830,6	818,5
772,1	811,3	792,7	818,4	806,4	833,8	831,3	819,1
775,1	813,8	795,5	819,3	806,6	832,8	830,6	818,0
773,3	811,7	792,7	818,7	805,8	831,9	829,7	817,3
775,9	815,3	796,0	821,4	808,6	833,6	831,9	819,1
773,8	813,1	792,6	818,7	805,6	830,6	829,5	817,0
775,0	813,5	793,5	820,4	806,2	831,6	831,2	818,7
730,2	763,2	765,6	783,8	790,0	805,0	810,1	803,5
729,3	761,9	763,7	782,2	789,2	804,2	809,6	802,8
729,2	761,5	762,1	781,2	788,3	803,4	809,0	801,8
732,5	765,9	766,8	785,3	791,7	807,3	812,6	805,5
731,8	765,5	767,0	785,6	791,8	807,8	812,9	805,7
731,1	764,0	765,2	784,5	790,1	806,1	811,0	803,8
734,8	767,5	768,3	788,2	792,3	808,6	812,9	805,9
736,1	768,6	770,2	789,3	795,3	811,6	814,9	807,4
789,2	834,5	834,2	845,7	837,4	854,6	851,2	842,0
790,8	835,1	833,7	846,1	838,1	854,8	851,6	841,6
782,7	827,4	825,9	838,1	832,3	847,5	844,7	835,3
790,4	836,5	835,8	848,4	840,2	855,6	853,1	844,3
785,7	832,8	830,2	841,4	836,1	851,9	847,9	839,0
737,9	772,4	778,8	794,1	804,5	826,5	833,3	825,0
748,3	782,9	788,6	804,7	813,0	835,9	841,8	834,8
736,3	770,4	774,2	790,1	803,1	825,7	831,2	824,8
733,3	767,5	770,4	785,6	801,3	823,7	829,2	822,8
746,8	782,4	785,6	798,7	813,0	835,0	840,3	833,8
753,3	789,6	791,1	805,1	815,7	836,6	841,6	834,3
745,0	780,7	781,4	796,7	807,7	828,3	832,2	825,8
744,1	779,1	781,0	795,7	807,1	828,3	831,0	825,8
744,0	776,9	779,2	794,9	807,6	829,2	831,1	827,0
777,9	813,9	795,9	822,5	807,2	835,6	832,5	819,7
771,7	811,1	793,3	818,5	805,9	833,5	830,6	818,5
772,1	811,3	792,7	818,4	806,4	833,8	831,3	819,1
775,1	813,8	795,5	819,3	806,6	832,8	830,6	818,0
773,3	811,7	792,7	818,7	805,8	831,9	829,7	817,3
775,9	815,3	796,0	821,4	808,6	833,6	831,9	819,1
773,8	813,1	792,6	818,7	805,6	830,6	829,5	817,0
741,3	766,9	774,0	788,9	799,8	819,7	820,9	813,5

[504] tansig – trainlm Fark = Hesaplanan - Gerçek							
-6,4	0,6	1,6	3,3	-3,4	-3,2	-3,8	-1,7
-5,6	1,2	0,2	1,1	-5,4	-8,2	-7,9	-3,7
0,6	4,9	2,8	5,2	-4,1	-6,1	-6,0	-2,5
0,3	4,8	3,4	5,2	-4,6	-6,3	-6,7	-3,0
-2,8	2,2	0,6	4,3	-4,8	-5,4	-6,0	-1,9
-1,9	4,4	3,4	5,0	-4,0	-4,5	-5,1	-1,3
-4,5	0,8	0,1	2,2	-6,8	-6,1	-7,3	-3,0
-2,4	2,9	3,5	5,0	-3,8	-3,1	-4,9	-1,0
-3,6	2,5	2,6	3,3	-4,4	-4,1	-6,6	-2,6
-3,5	3,4	-9,4	-1,9	-19,6	-10,8	-13,0	-10,2
-2,6	3,8	-8,5	-1,3	-20,6	-10,9	-13,5	-9,5
-3,4	3,2	-6,8	-1,2	-19,7	-11,1	-13,8	-9,5
-7,6	-1,2	-12,5	-6,3	-24,1	-15,0	-17,5	-13,2
-7,9	-1,7	-13,6	-7,5	-25,1	-16,5	-18,7	-14,3
-8,2	-1,1	-11,8	-6,4	-23,4	-14,7	-16,8	-13,4
-11,8	-5,6	-15,9	-11,1	-26,6	-18,2	-19,6	-15,5
-14,1	-7,7	-18,7	-13,1	-29,6	-22,1	-21,6	-17,0
12,6	8,1	-8,7	14,1	-7,1	5,2	8,5	13,9
11,0	7,6	-8,1	12,7	-7,8	4,0	8,2	13,4
19,1	15,2	-0,4	20,7	-2,0	11,3	15,1	19,7
10,5	7,1	-10,3	10,4	-10,8	3,2	5,7	10,7
15,2	10,8	-4,7	17,4	-6,8	5,9	10,9	15,1
24,0	29,4	18,2	31,5	-0,8	3,8	-1,1	3,4
12,7	17,9	7,5	20,9	-9,3	-5,6	-9,6	-6,4
23,7	28,6	20,9	34,5	-0,4	4,6	0,1	2,6
25,8	31,5	23,8	39,0	1,4	5,6	2,0	4,6
12,3	15,6	8,6	25,0	-10,3	-5,7	-9,1	-8,3
4,8	7,5	2,2	18,6	-13,9	-8,2	-11,3	-8,7
13,1	16,4	11,8	26,0	-5,9	0,1	-1,9	-0,2
13,1	17,0	11,3	26,0	-5,3	0,1	-0,7	-0,2
12,2	18,3	12,2	26,9	-5,8	-0,8	-1,8	-2,4
-14,1	-7,3	-5,5	-3,6	-6,4	-10,1	-7,9	0,1
-7,9	-4,6	-2,9	0,4	-5,0	-8,0	-6,0	0,4
-8,3	-4,7	-2,3	0,5	-5,6	-9,2	-6,7	-0,2
-12,3	-7,3	-5,1	-0,4	-6,7	-7,3	-6,0	0,9
-10,5	-5,1	-3,2	0,2	-5,9	-7,3	-5,1	1,6
-14,0	-9,7	-5,6	-3,5	-8,7	-9,0	-7,3	-0,2
-11,9	-7,5	-3,1	-0,7	-5,7	-6,0	-5,8	1,9
33,9	37,7	30,7	31,9	18,1	14,4	17,0	22,5

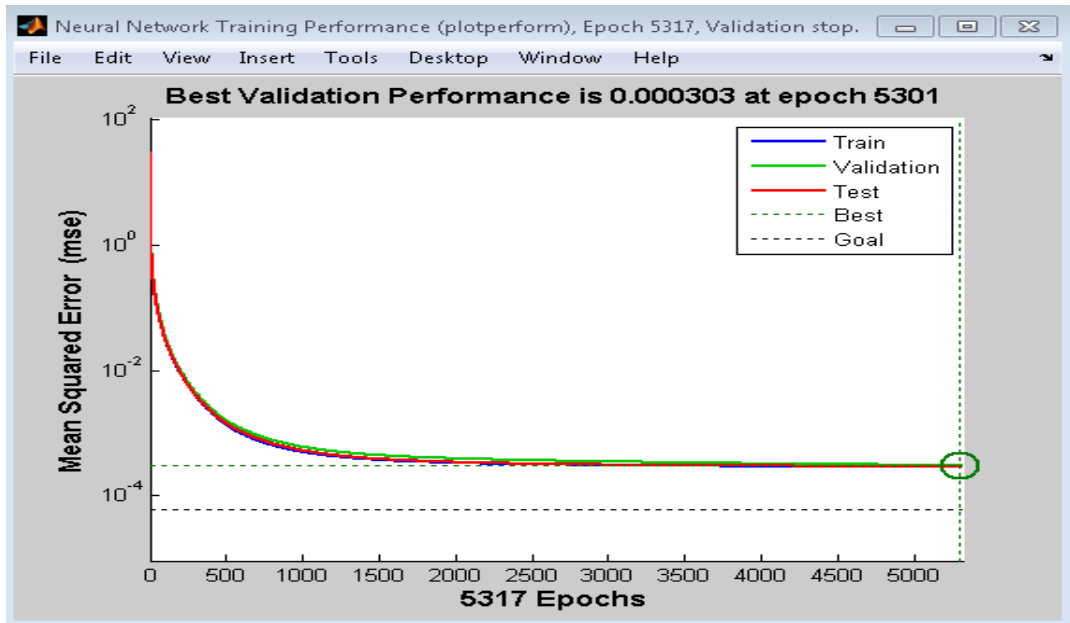
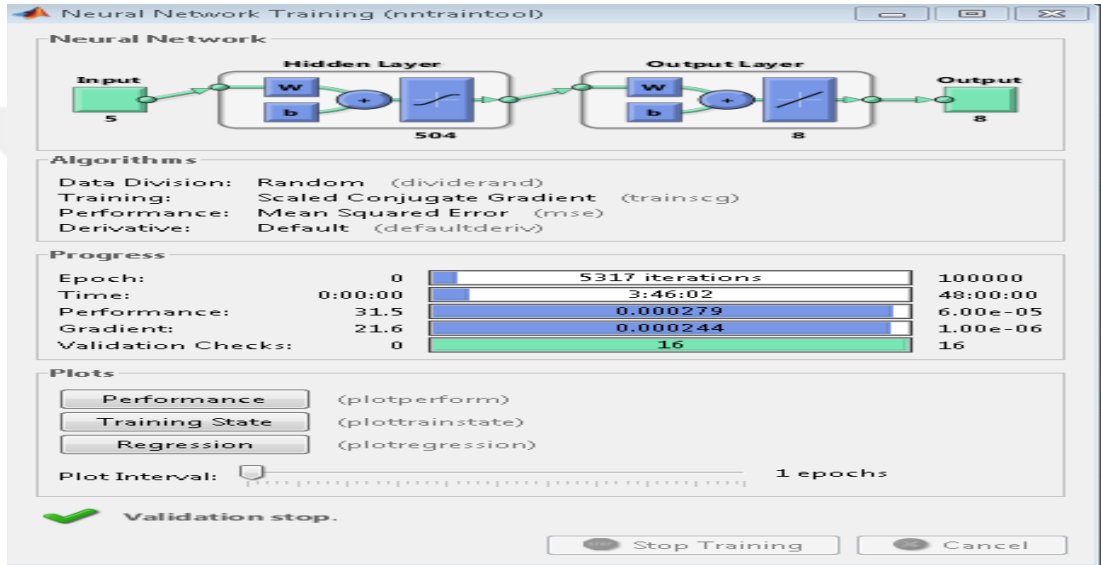
[504] – tansig – trainscg

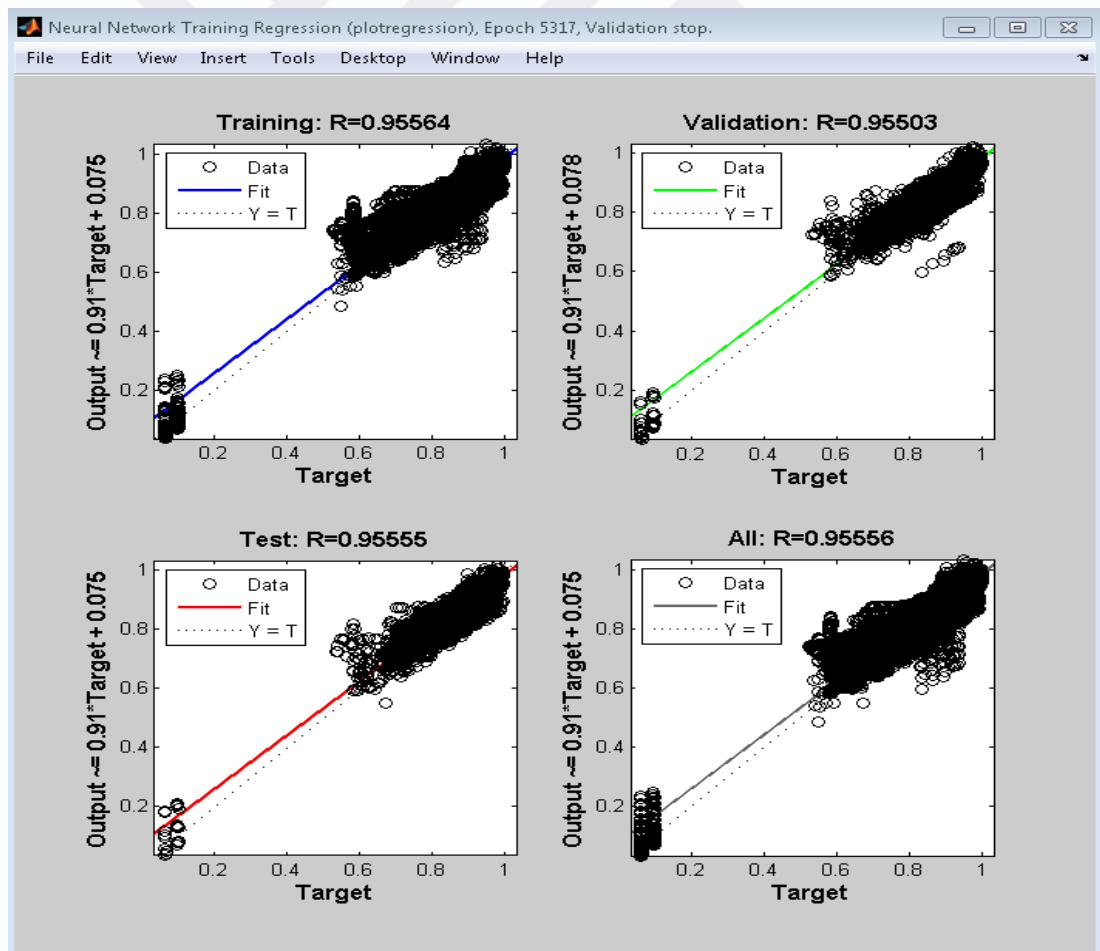
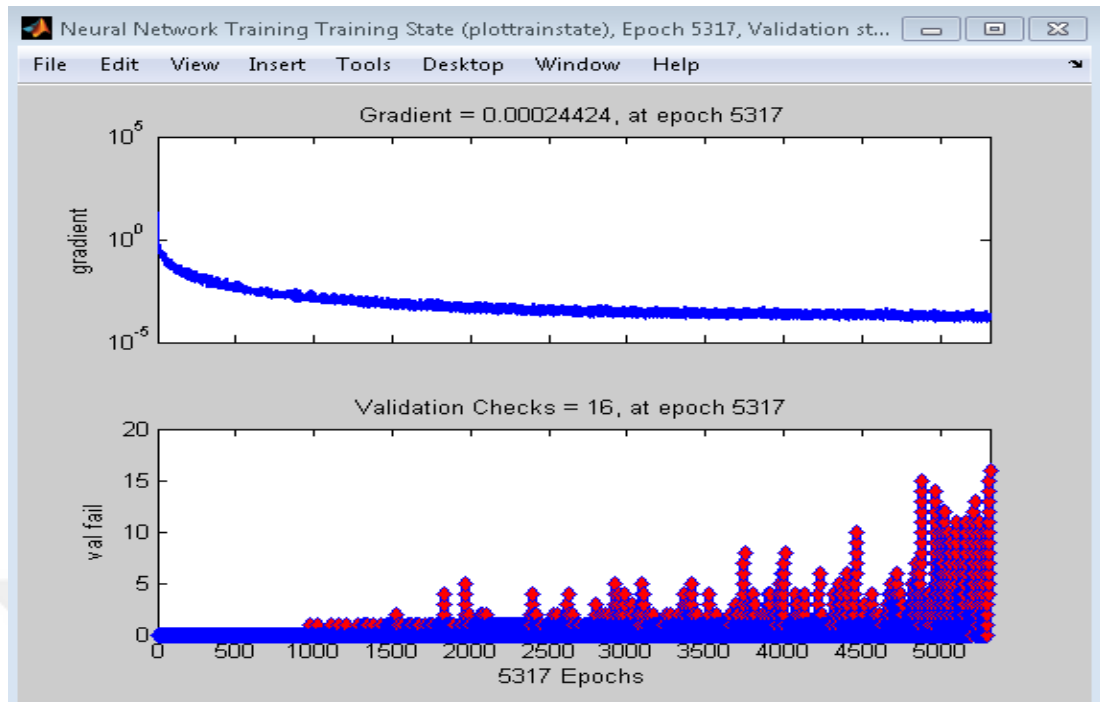
Uygulanan matlab program aşağıdaki gibidir. Nöron hesabı formülüne göre $\alpha = 9$ seçilerek nöron sayısı 504 bulunmuştur. Bir üstteki program ile aynı yapıdadır. Sadece nöron sayısı farklıdır.

Performans değerimiz ise;

$$\text{Performans (mse)} = 3.05 \cdot 10^{-4}$$

olarak bulunmuştur.





[504] Tansig – Trainseg Gerçek Fırın Sıcaklıkları							
768,55	814,15	795,15	823,65	802,75	828,4	827,45	817
772,35	815,1	796,1	823,65	801,8	827,45	824,6	816,05
772,35	816,05	796,1	823,65	801,8	827,45	824,6	816,05
772,35	816,05	796,1	823,65	801,8	827,45	824,6	816,05
772,35	816,05	796,1	823,65	801,8	827,45	824,6	816,05
771,4	816,05	796,1	823,65	801,8	827,45	824,6	816,05
771,4	816,05	796,1	823,65	801,8	827,45	824,6	816,05
771,4	816,05	796,1	823,65	801,8	827,45	824,6	816,05
771,4	816,05	796,1	823,65	801,8	827,45	824,6	816,05
726,75	766,65	756,2	781,85	770,45	794,2	797,05	793,25
726,75	765,7	755,25	780,9	768,55	793,25	796,1	793,25
725,8	764,75	755,25	779,95	768,55	792,3	795,15	792,3
724,85	764,75	754,3	779	767,6	792,3	795,15	792,3
723,9	763,8	753,35	778,05	766,65	791,35	794,2	791,35
722,95	762,85	753,35	778,05	766,65	791,35	794,2	790,4
722,95	761,9	752,4	777,1	765,7	790,4	793,25	790,4
722	760,95	751,45	776,15	765,7	789,45	793,25	790,4
801,8	842,65	825,55	859,75	830,3	859,75	859,75	855,95
801,8	842,65	825,55	858,8	830,3	858,8	859,75	855
801,8	842,65	825,55	858,8	830,3	858,8	859,75	855
800,85	843,6	825,55	858,8	829,35	858,8	858,8	855
800,85	843,6	825,55	858,8	829,35	857,85	858,8	854,05
761,9	801,8	797,05	825,55	803,7	830,3	832,2	828,4
760,95	800,85	796,1	825,55	803,7	830,3	832,2	828,4
760	798,95	795,15	824,6	802,75	830,3	831,25	827,45
759,05	798,95	794,2	824,6	802,75	829,35	831,25	827,45
759,05	798	794,2	823,65	802,75	829,35	831,25	825,55
758,1	797,05	793,25	823,65	801,8	828,4	830,3	825,55
758,1	797,05	793,25	822,7	801,8	828,4	830,3	825,55
757,15	796,1	792,3	821,75	801,8	828,4	830,3	825,55
756,2	795,15	791,35	821,75	801,8	828,4	829,35	824,6
763,8	806,55	790,4	818,9	800,85	825,55	824,6	819,85
763,8	806,55	790,4	818,9	800,85	825,55	824,6	818,9
763,8	806,55	790,4	818,9	800,85	824,6	824,6	818,9
762,85	806,55	790,4	818,9	799,9	825,55	824,6	818,9
762,85	806,55	789,45	818,9	799,9	824,6	824,6	818,9
761,9	805,6	790,4	817,95	799,9	824,6	824,6	818,9
761,9	805,6	789,45	817,95	799,9	824,6	823,65	818,9
775,2	804,65	804,65	820,8	817,95	834,1	837,9	836

[504] Tansig – Trainscg Hesaplanan Fırın Sıcaklıkları							
776,445	808,537	790,304	811,222	809,551	827,332	826,613	818,022
781,500	814,935	796,209	811,529	807,541	827,988	830,411	815,087
773,824	810,531	794,331	809,585	805,986	826,059	826,531	818,701
770,466	811,352	793,056	810,166	804,423	827,613	825,080	816,511
770,619	812,688	792,451	811,121	805,954	825,907	823,294	813,221
768,793	810,690	792,599	809,628	807,087	825,704	822,997	812,914
769,135	808,516	793,978	807,928	806,883	827,171	824,741	815,087
771,745	807,219	794,034	807,673	807,003	827,187	824,962	816,318
776,445	808,537	790,304	811,222	809,551	827,332	826,613	818,022
729,302	765,634	761,954	778,205	782,812	797,029	800,115	790,286
732,535	763,783	762,724	777,571	784,302	796,228	800,586	789,936
734,247	763,605	764,896	779,172	786,070	797,138	801,082	792,496
731,407	761,527	763,203	776,503	785,535	797,407	801,209	794,749
726,481	758,709	760,434	771,985	784,929	797,334	800,498	796,164
725,053	758,537	760,527	771,074	785,122	797,364	800,223	796,264
725,922	760,391	761,254	772,043	784,224	796,724	800,205	794,352
725,172	762,200	761,476	773,450	782,076	795,897	798,640	792,074
786,586	825,000	818,669	835,651	826,026	844,240	841,969	833,255
784,898	824,834	815,815	835,259	822,501	844,772	843,091	831,136
788,434	827,406	818,251	837,164	824,997	844,212	843,695	832,835
790,653	830,222	821,345	839,262	827,930	844,535	844,063	834,649
793,067	831,375	826,670	840,336	831,506	847,482	848,916	836,441
709,988	755,719	759,689	768,944	783,275	802,527	803,984	797,467
722,567	757,460	764,762	774,301	789,229	810,770	812,332	805,301
728,674	759,124	768,754	778,603	794,110	814,218	814,933	809,762
727,266	756,653	769,307	778,102	794,348	812,333	814,514	809,930
724,241	753,359	767,277	775,841	790,892	808,949	814,409	807,969
736,237	766,086	775,188	786,002	793,342	812,876	819,707	810,278
744,801	776,244	782,311	793,568	797,665	817,688	824,063	814,271
746,980	780,660	786,434	795,295	801,482	821,917	826,311	817,786
746,291	782,097	788,300	795,880	804,189	825,436	825,330	820,823
781,500	814,935	796,209	811,529	807,541	827,988	830,411	815,087
773,824	810,531	794,331	809,585	805,986	826,059	826,531	818,701
770,466	811,352	793,056	810,166	804,423	827,613	825,080	816,511
770,619	812,688	792,451	811,121	805,954	825,907	823,294	813,221
768,793	810,690	792,599	809,628	807,087	825,704	822,997	812,914
769,135	808,516	793,978	807,928	806,883	827,171	824,741	815,087
771,745	807,219	794,034	807,673	807,003	827,187	824,962	816,318
732,364	766,112	773,751	781,456	789,507	810,671	813,535	808,249

Fark = Hesaplanan - Gerçek							
-7,895	5,613	4,846	12,428	-6,801	1,068	0,837	-1,022
-9,150	0,165	-0,109	12,121	-5,741	-0,538	-5,811	0,963
-1,474	5,519	1,769	14,065	-4,186	1,391	-1,931	-2,651
1,884	4,698	3,044	13,484	-2,623	-0,163	-0,480	-0,461
1,731	3,362	3,649	12,529	-4,154	1,543	1,306	2,829
2,607	5,360	3,501	14,022	-5,287	1,746	1,603	3,136
2,265	7,534	2,122	15,722	-5,083	0,279	-0,141	0,963
-0,345	8,831	2,066	15,977	-5,203	0,263	-0,362	-0,268
-5,045	7,513	5,796	12,428	-7,751	0,118	-2,013	-1,972
-2,552	1,016	-5,754	3,645	-12,362	-2,829	-3,065	2,964
-5,785	1,917	-7,474	3,329	-15,752	-2,978	-4,486	3,314
-8,447	1,145	-9,646	0,778	-17,520	-4,838	-5,932	-0,196
-6,557	3,223	-8,903	2,497	-17,935	-5,107	-6,059	-2,449
-2,581	5,091	-7,084	6,065	-18,279	-5,984	-6,298	-4,814
-2,103	4,313	-7,177	6,976	-18,472	-6,014	-6,023	-5,864
-2,972	1,509	-8,854	5,057	-18,524	-6,324	-6,955	-3,952
-3,172	-1,250	-10,026	2,700	-16,376	-6,447	-5,390	-1,674
15,214	17,650	6,881	24,099	4,274	15,510	17,781	22,695
16,902	17,816	9,735	23,541	7,799	14,028	16,659	23,864
13,366	15,244	7,299	21,636	5,303	14,588	16,055	22,165
10,197	13,378	4,205	19,538	1,420	14,265	14,737	20,351
7,783	12,225	-1,120	18,464	-2,156	10,368	9,884	17,609
51,912	46,081	37,361	56,606	20,425	27,773	28,216	30,933
38,383	43,390	31,338	51,249	14,471	19,530	19,868	23,099
31,326	39,826	26,396	45,997	8,640	16,082	16,317	17,688
31,784	42,297	24,893	46,498	8,402	17,017	16,736	17,520
34,809	44,641	26,923	47,809	11,858	20,401	16,841	17,581
21,863	30,964	18,062	37,648	8,458	15,524	10,593	15,272
13,299	20,806	10,939	29,132	4,135	10,712	6,237	11,279
10,170	15,440	5,866	26,455	0,318	6,483	3,989	7,764
9,909	13,053	3,050	25,870	-2,389	2,964	4,020	3,777
-17,700	-8,385	-5,809	7,371	-6,691	-2,438	-5,811	4,763
-10,024	-3,981	-3,931	9,315	-5,136	-0,509	-1,931	0,199
-6,666	-4,802	-2,656	8,734	-3,573	-3,013	-0,480	2,389
-7,769	-6,138	-2,051	7,779	-6,054	-0,357	1,306	5,679
-5,943	-4,140	-3,149	9,272	-7,187	-1,104	1,603	5,986
-7,235	-2,916	-3,578	10,022	-6,983	-2,571	-0,141	3,813
-9,845	-1,619	-4,584	10,277	-7,103	-2,587	-1,312	2,582
42,836	38,538	30,899	39,344	28,443	23,429	24,365	27,751

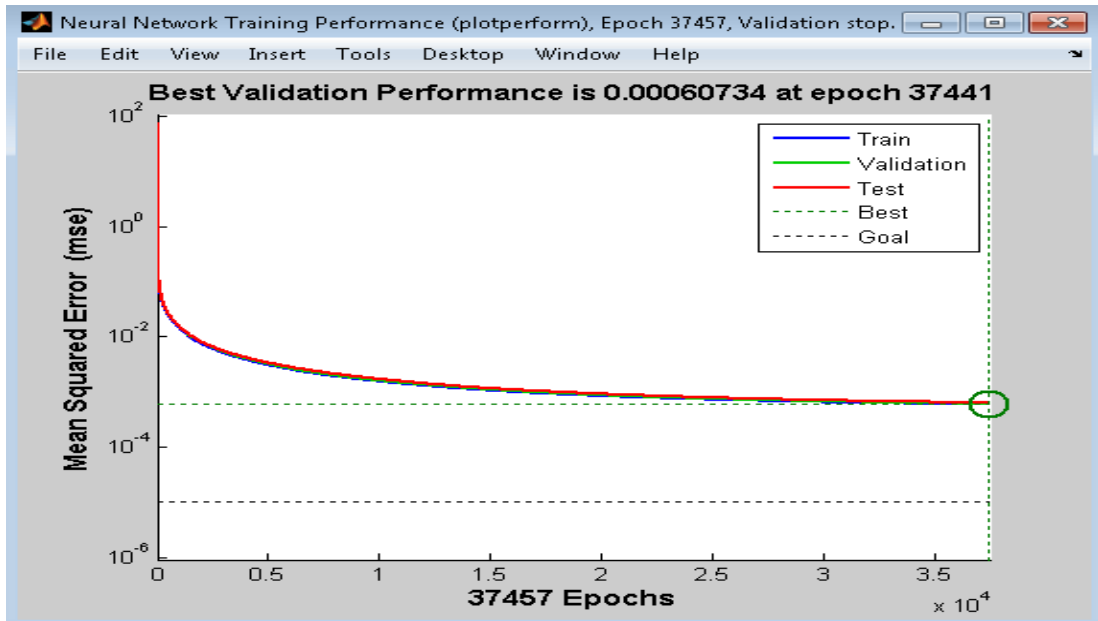
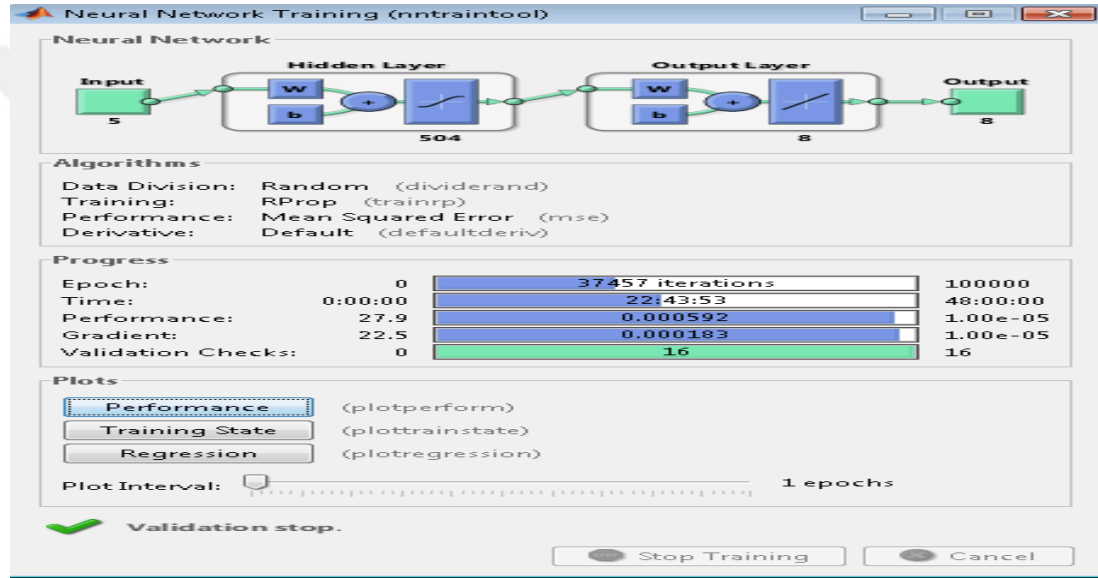
[504] – tansig – trainrp

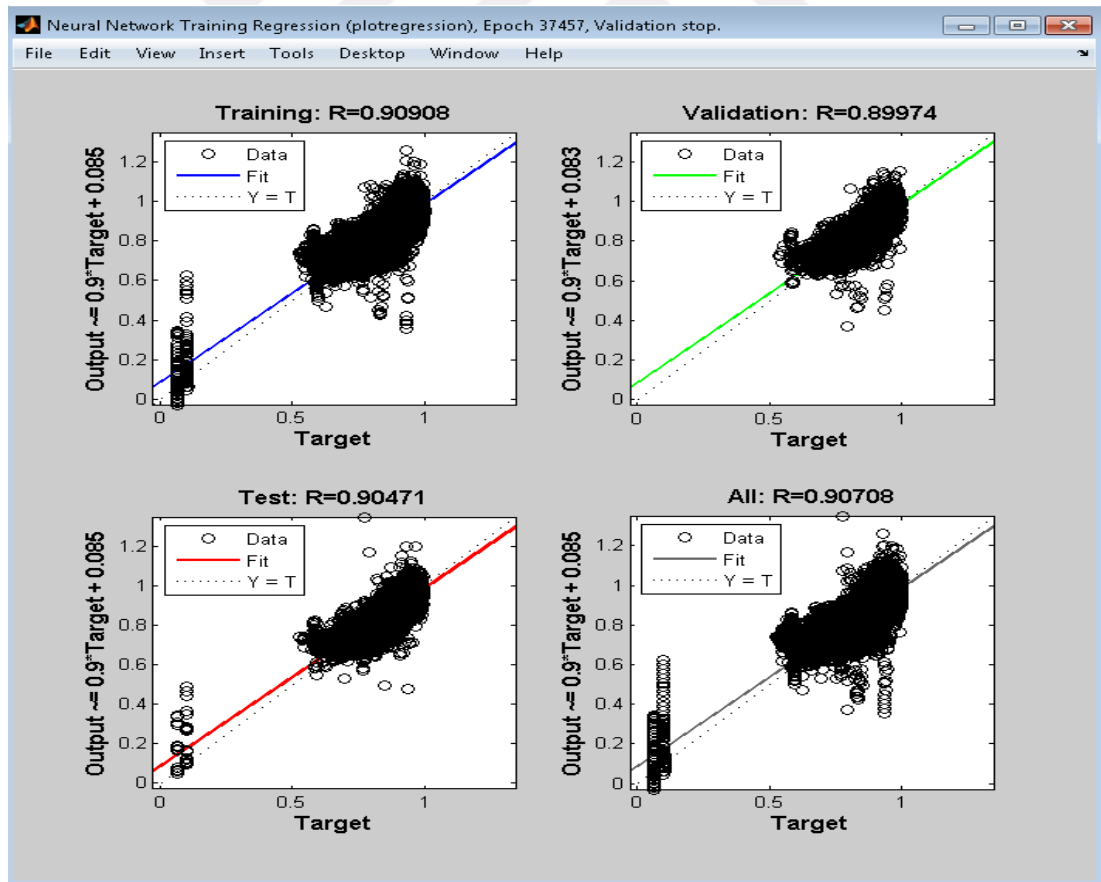
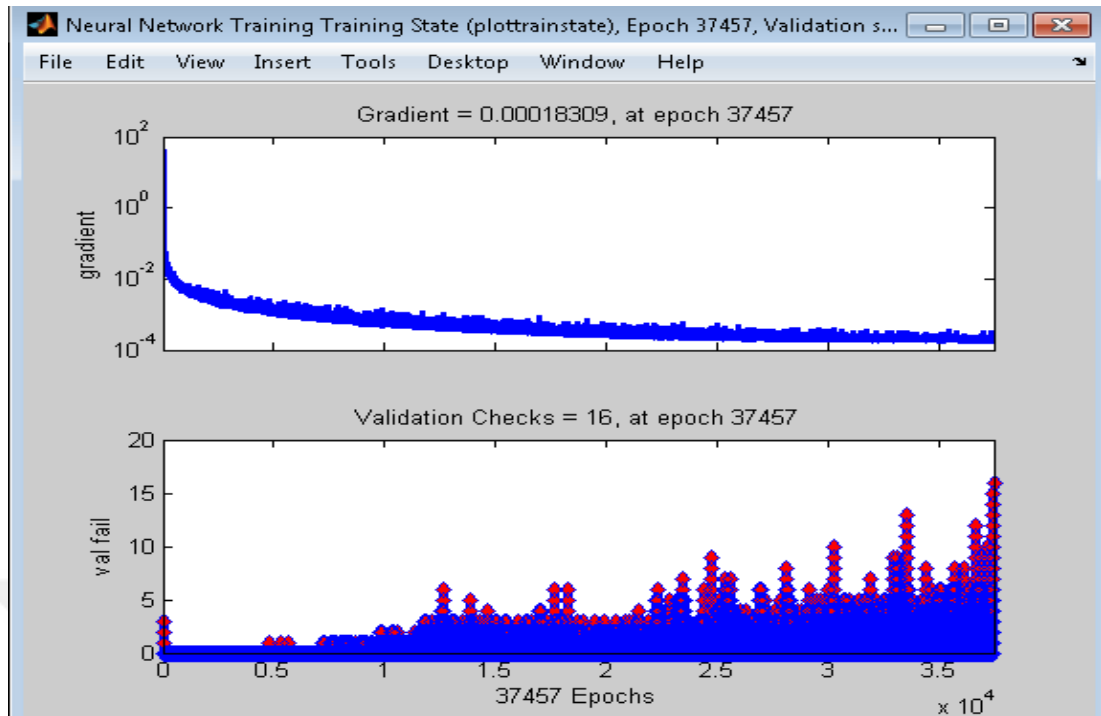
Uygulanan matlab program aşağıdaki gibidir. Nöron hesabı formülüne göre $\alpha = 9$ seçilerek nöron sayısı 504 bulunmuştur. Bir üstteki program ile aynı yapıdadır. Sadece nöron sayısı farklıdır.

Performans değerimiz ise;

$$\text{Performans (mse)} = 3.63 \cdot 10^{-4}$$

olarak bulunmuştur.





Gerçek Fırın Sıcaklıkları							
768,6	814,2	795,2	823,7	802,8	828,4	827,5	817,0
772,4	815,1	796,1	823,7	801,8	827,5	824,6	816,1
772,4	816,1	796,1	823,7	801,8	827,5	824,6	816,1
772,4	816,1	796,1	823,7	801,8	827,5	824,6	816,1
772,4	816,1	796,1	823,7	801,8	827,5	824,6	816,1
771,4	816,1	796,1	823,7	801,8	827,5	824,6	816,1
771,4	816,1	796,1	823,7	801,8	827,5	824,6	816,1
771,4	816,1	796,1	823,7	801,8	827,5	824,6	816,1
771,4	816,1	796,1	823,7	801,8	827,5	824,6	816,1
726,8	766,7	756,2	781,9	770,5	794,2	797,1	793,3
726,8	765,7	755,3	780,9	768,6	793,3	796,1	793,3
725,8	764,8	755,3	780,0	768,6	792,3	795,2	792,3
724,9	764,8	754,3	779,0	767,6	792,3	795,2	792,3
723,9	763,8	753,4	778,1	766,7	791,4	794,2	791,4
723,0	762,9	753,4	778,1	766,7	791,4	794,2	790,4
723,0	761,9	752,4	777,1	765,7	790,4	793,3	790,4
722,0	761,0	751,5	776,2	765,7	789,5	793,3	790,4
801,8	842,7	825,6	859,8	830,3	859,8	859,8	856,0
801,8	842,7	825,6	858,8	830,3	858,8	859,8	855,0
801,8	842,7	825,6	858,8	830,3	858,8	859,8	855,0
800,9	843,6	825,6	858,8	829,4	858,8	858,8	855,0
800,9	843,6	825,6	858,8	829,4	857,9	858,8	854,1
761,9	801,8	797,1	825,6	803,7	830,3	832,2	828,4
761,0	800,9	796,1	825,6	803,7	830,3	832,2	828,4
760,0	799,0	795,2	824,6	802,8	830,3	831,3	827,5
759,1	799,0	794,2	824,6	802,8	829,4	831,3	827,5
759,1	798,0	794,2	823,7	802,8	829,4	831,3	825,6
758,1	797,1	793,3	823,7	801,8	828,4	830,3	825,6
758,1	797,1	793,3	822,7	801,8	828,4	830,3	825,6
757,2	796,1	792,3	821,8	801,8	828,4	830,3	825,6
756,2	795,2	791,4	821,8	801,8	828,4	829,4	824,6
763,8	806,6	790,4	818,9	800,9	825,6	824,6	819,9
763,8	806,6	790,4	818,9	800,9	825,6	824,6	818,9
763,8	806,6	790,4	818,9	800,9	824,6	824,6	818,9
762,9	806,6	790,4	818,9	799,9	825,6	824,6	818,9
762,9	806,6	789,5	818,9	799,9	824,6	824,6	818,9
761,9	805,6	790,4	818,0	799,9	824,6	824,6	818,9
761,9	805,6	789,5	818,0	799,9	824,6	823,7	818,9
775,2	804,7	804,7	820,8	818,0	834,1	837,9	836,0

Hesaplanan Fırın Sıcaklıkları							
760,5	832,8	821,0	802,7	800,4	811,6	828,0	806,9
752,8	813,8	815,4	824,2	780,0	846,1	825,8	810,8
761,3	832,2	810,9	822,9	802,5	844,5	803,6	802,4
767,4	818,0	801,5	810,6	816,8	834,4	803,8	811,8
775,4	803,6	799,5	813,6	817,3	822,3	817,6	822,3
781,1	807,3	801,0	828,4	809,2	814,8	835,0	828,1
776,4	820,4	802,9	833,9	804,8	814,1	846,9	829,0
763,3	828,5	808,9	823,5	805,2	815,6	845,9	825,5
760,5	832,8	821,0	802,7	800,4	811,6	828,0	806,9
719,2	765,1	767,1	784,1	801,6	790,3	796,9	832,9
734,2	765,7	751,0	790,1	791,2	790,8	800,0	825,4
741,6	762,8	742,9	788,3	776,1	794,2	799,6	805,5
740,1	762,0	741,9	781,7	766,7	803,5	797,7	784,2
737,1	763,3	742,8	773,2	767,6	812,5	795,4	773,9
736,9	763,2	744,1	763,6	774,0	811,6	794,5	779,0
736,1	759,7	746,3	753,2	779,0	801,2	796,9	795,7
731,2	754,3	749,6	748,4	777,2	793,1	803,8	815,1
795,6	819,5	836,8	841,3	827,5	843,5	852,0	835,8
776,3	826,3	828,3	846,9	832,3	867,3	860,6	847,7
778,5	829,9	828,0	840,3	832,1	854,4	862,9	840,0
799,2	820,5	819,8	840,6	825,7	834,9	850,3	834,8
809,4	822,8	824,8	862,2	815,2	866,8	824,0	829,4
719,1	763,6	754,6	821,4	784,9	835,2	829,9	846,6
713,4	764,0	759,6	811,5	775,8	836,3	821,1	837,8
711,1	776,0	763,2	809,8	780,9	821,2	827,7	818,1
713,1	789,9	772,5	813,2	787,8	801,8	835,0	797,5
719,5	798,9	786,6	816,2	790,2	785,4	832,6	780,2
731,7	802,0	801,8	813,9	786,9	786,6	827,2	784,8
740,6	793,7	800,0	802,0	787,7	793,5	818,7	794,9
740,0	780,8	789,4	796,3	790,3	805,6	810,2	803,7
728,1	771,3	783,1	805,0	800,7	819,5	808,4	817,3
752,8	813,8	815,4	824,2	780,0	846,1	825,8	810,8
761,3	832,2	810,9	822,9	802,5	844,5	803,6	802,4
767,4	818,0	801,5	810,6	816,8	834,4	803,8	811,8
775,4	803,6	799,5	813,6	817,3	822,3	817,6	822,3
781,1	807,3	801,0	828,4	809,2	814,8	835,0	828,1
776,4	820,4	802,9	833,9	804,8	814,1	846,9	829,0
763,3	828,5	808,9	823,5	805,2	815,6	845,9	825,5
737,1	779,4	808,8	812,3	767,3	834,9	821,1	796,0

Fark = Hesaplanan - Gerçek							
8,0	-18,6	-25,8	20,9	2,3	16,8	-0,5	10,1
19,5	1,3	-19,3	-0,5	21,8	-18,7	-1,2	5,3
11,0	-16,2	-14,8	0,8	-0,7	-17,0	21,0	13,7
4,9	-1,9	-5,4	13,0	-15,0	-7,0	20,8	4,2
-3,0	12,4	-3,4	10,0	-15,5	5,1	7,0	-6,2
-9,7	8,7	-4,9	-4,7	-7,4	12,6	-10,4	-12,1
-5,0	-4,4	-6,8	-10,3	-3,0	13,4	-22,3	-13,0
8,1	-12,4	-12,8	0,1	-3,4	11,8	-21,3	-9,5
10,9	-16,7	-24,9	20,9	1,4	15,9	-3,4	9,1
7,6	1,5	-10,9	-2,2	-31,2	3,9	0,1	-39,6
-7,5	0,0	4,3	-9,2	-22,6	2,4	-3,9	-32,2
-15,8	1,9	12,3	-8,3	-7,6	-1,9	-4,5	-13,2
-15,2	2,7	12,4	-2,7	0,9	-11,2	-2,5	8,1
-13,2	0,5	10,5	4,8	-0,9	-21,1	-1,2	17,4
-13,9	-0,3	9,2	14,5	-7,3	-20,2	-0,3	11,4
-13,2	2,2	6,1	23,9	-13,3	-10,8	-3,6	-5,3
-9,2	6,7	1,9	27,7	-11,5	-3,7	-10,6	-24,7
6,2	23,2	-11,3	18,5	2,8	16,3	7,8	20,1
25,5	16,3	-2,8	11,9	-2,0	-8,5	-0,9	7,3
23,3	12,8	-2,5	18,5	-1,8	4,4	-3,1	15,0
1,6	23,1	5,8	18,2	3,6	23,9	8,5	20,2
-8,5	20,8	0,7	-3,4	14,2	-9,0	34,8	24,6
42,8	38,2	42,5	4,1	18,8	-4,9	2,3	-18,2
47,5	36,9	36,5	14,1	27,9	-6,0	11,1	-9,4
48,9	23,0	31,9	14,8	21,9	9,1	3,5	9,3
46,0	9,1	21,7	11,4	15,0	27,6	-3,7	29,9
39,6	-0,9	7,6	7,4	12,6	44,0	-1,3	45,4
26,4	-4,9	-8,6	9,8	14,9	41,8	3,1	40,8
17,5	3,4	-6,7	20,7	14,1	34,9	11,6	30,6
17,2	15,3	2,9	25,4	11,5	22,8	20,1	21,9
28,1	23,9	8,3	16,7	1,1	8,9	20,9	7,3
11,0	-7,2	-25,0	-5,3	20,8	-20,6	-1,2	9,1
2,5	-25,7	-20,5	-4,0	-1,7	-18,9	21,0	16,5
-3,6	-11,4	-11,1	8,3	-16,0	-9,8	20,8	7,1
-12,5	2,9	-9,1	5,3	-17,4	3,2	7,0	-3,4
-18,2	-0,8	-11,6	-9,5	-9,3	9,8	-10,4	-9,2
-14,5	-14,8	-12,5	-16,0	-4,9	10,5	-22,3	-10,1
-1,4	-22,9	-19,5	-5,6	-5,3	9,0	-22,3	-6,6
38,1	25,3	-4,2	8,5	50,6	-0,8	16,8	40,0

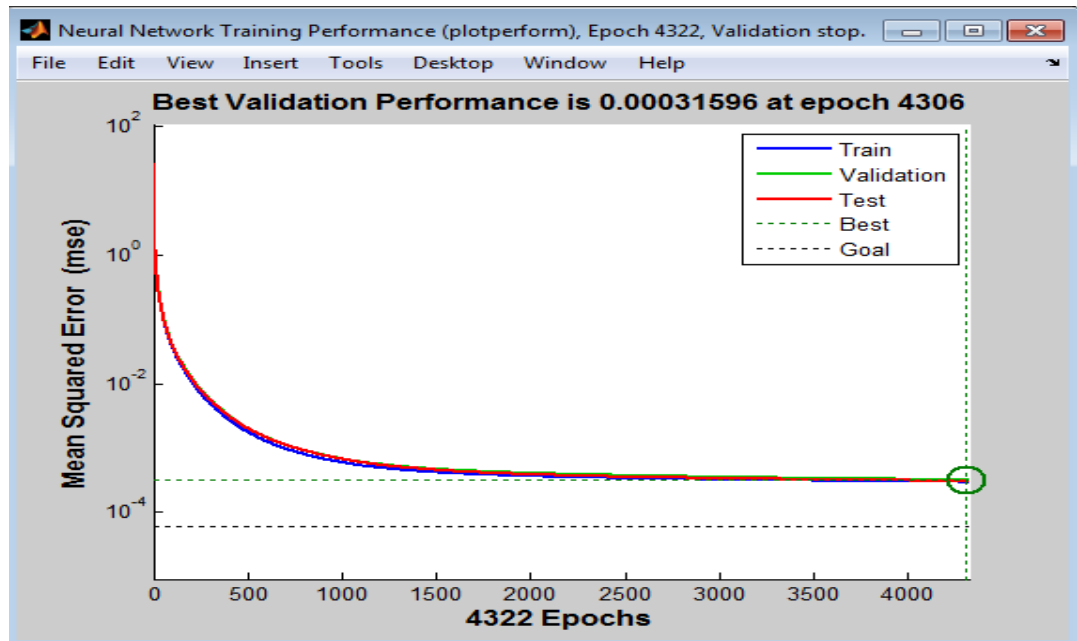
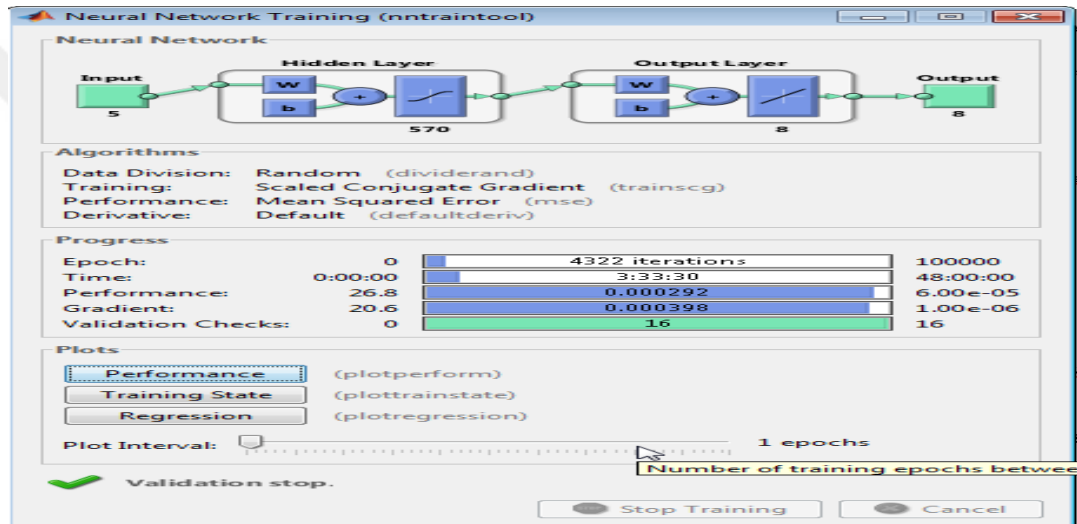
[570] – tansig – trainscg

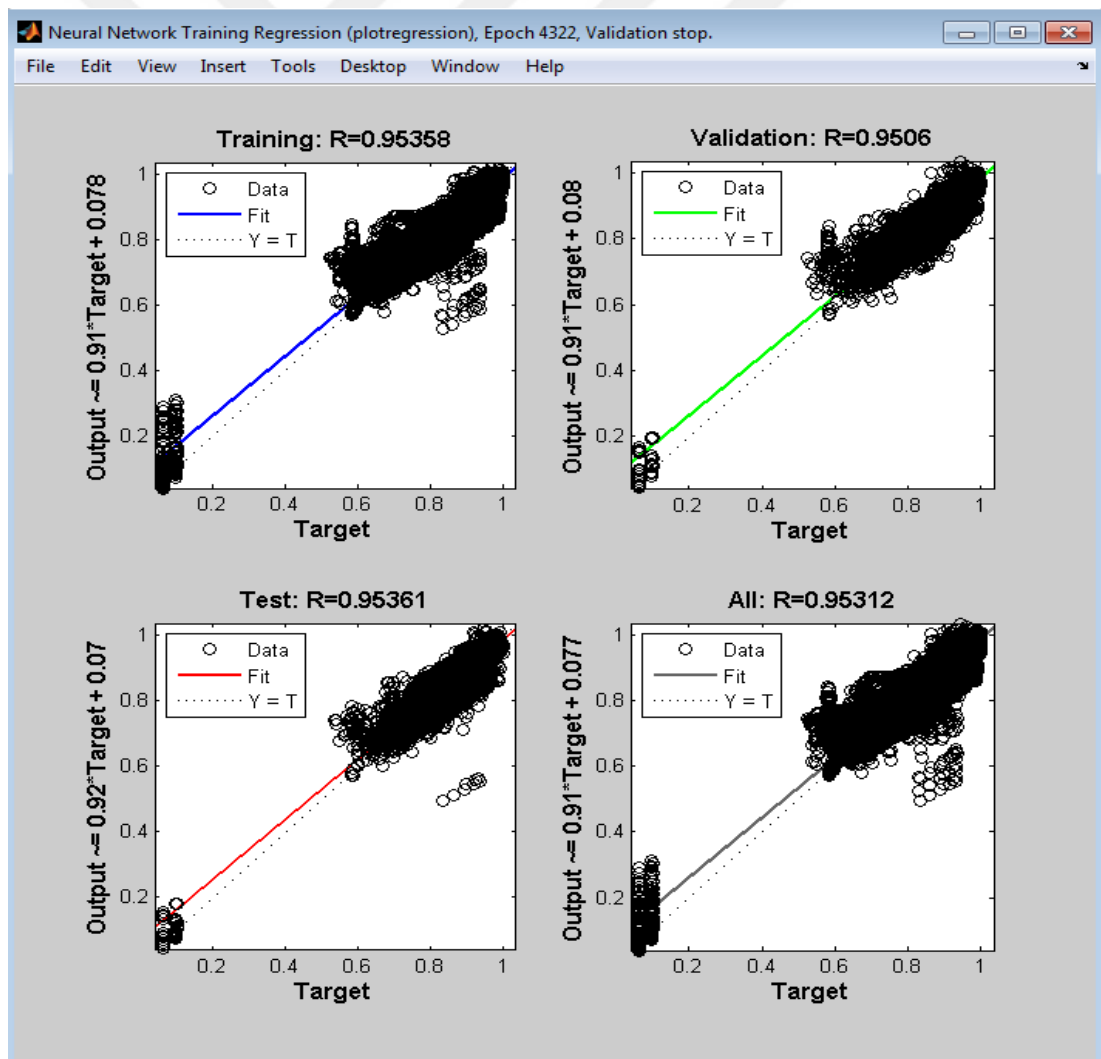
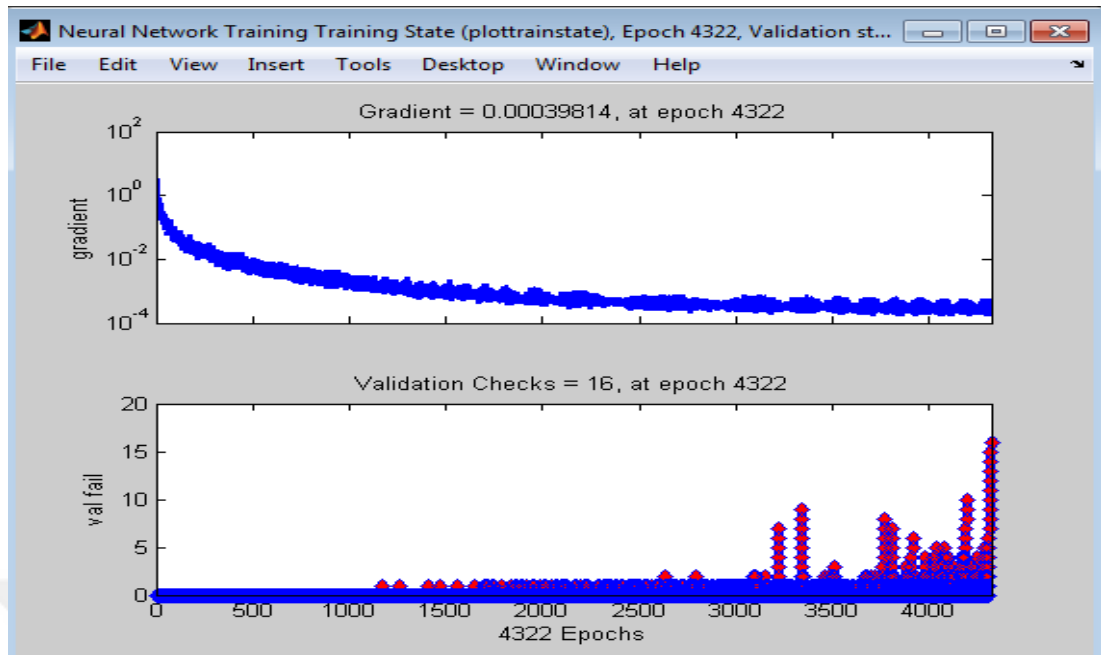
Uygulanan matlab program aşağıdaki gibidir. Nöron hesabı formülüne göre $\alpha = 8$ seçilerek nöron sayısı 570 bulunmuştur. Bir üstteki program ile aynı yapıdadır. Sadece nöron sayısı farklıdır.

Performans değerimiz ise;

$$\text{Performans (mse)} = 2.12 \cdot 10^{-4}$$

olarak bulunmuştur.





Gerçek Fırın Sıcaklıkları							
768,6	814,2	795,2	823,7	802,8	828,4	827,5	817,0
772,4	815,1	796,1	823,7	801,8	827,5	824,6	816,1
772,4	816,1	796,1	823,7	801,8	827,5	824,6	816,1
772,4	816,1	796,1	823,7	801,8	827,5	824,6	816,1
772,4	816,1	796,1	823,7	801,8	827,5	824,6	816,1
771,4	816,1	796,1	823,7	801,8	827,5	824,6	816,1
771,4	816,1	796,1	823,7	801,8	827,5	824,6	816,1
771,4	816,1	796,1	823,7	801,8	827,5	824,6	816,1
771,4	816,1	796,1	823,7	801,8	827,5	824,6	816,1
726,8	766,7	756,2	781,9	770,5	794,2	797,1	793,3
726,8	765,7	755,3	780,9	768,6	793,3	796,1	793,3
725,8	764,8	755,3	780,0	768,6	792,3	795,2	792,3
724,9	764,8	754,3	779,0	767,6	792,3	795,2	792,3
723,9	763,8	753,4	778,1	766,7	791,4	794,2	791,4
723,0	762,9	753,4	778,1	766,7	791,4	794,2	790,4
723,0	761,9	752,4	777,1	765,7	790,4	793,3	790,4
722,0	761,0	751,5	776,2	765,7	789,5	793,3	790,4
801,8	842,7	825,6	859,8	830,3	859,8	859,8	856,0
801,8	842,7	825,6	858,8	830,3	858,8	859,8	855,0
801,8	842,7	825,6	858,8	830,3	858,8	859,8	855,0
800,9	843,6	825,6	858,8	829,4	858,8	858,8	855,0
800,9	843,6	825,6	858,8	829,4	857,9	858,8	854,1
761,9	801,8	797,1	825,6	803,7	830,3	832,2	828,4
761,0	800,9	796,1	825,6	803,7	830,3	832,2	828,4
760,0	799,0	795,2	824,6	802,8	830,3	831,3	827,5
759,1	799,0	794,2	824,6	802,8	829,4	831,3	827,5
759,1	798,0	794,2	823,7	802,8	829,4	831,3	825,6
758,1	797,1	793,3	823,7	801,8	828,4	830,3	825,6
758,1	797,1	793,3	822,7	801,8	828,4	830,3	825,6
757,2	796,1	792,3	821,8	801,8	828,4	830,3	825,6
756,2	795,2	791,4	821,8	801,8	828,4	829,4	824,6
763,8	806,6	790,4	818,9	800,9	825,6	824,6	819,9
763,8	806,6	790,4	818,9	800,9	825,6	824,6	818,9
763,8	806,6	790,4	818,9	800,9	824,6	824,6	818,9
762,9	806,6	790,4	818,9	799,9	825,6	824,6	818,9
762,9	806,6	789,5	818,9	799,9	824,6	824,6	818,9
761,9	805,6	790,4	818,0	799,9	824,6	824,6	818,9
761,9	805,6	789,5	818,0	799,9	824,6	823,7	818,9
775,2	804,7	804,7	820,8	818,0	834,1	837,9	836,0

Hesaplanan Fırın Sıcaklıkları							
772,7	812,3	799,3	815,2	812,5	832,5	825,4	815,6
774,6	816,6	800,5	821,7	817,1	836,1	838,4	826,2
770,4	812,3	796,8	814,5	813,4	828,6	831,4	816,4
772,0	812,6	797,2	815,0	812,0	827,3	828,9	815,2
774,4	815,8	798,1	816,3	811,7	829,0	829,1	817,6
769,0	814,2	796,3	812,7	809,1	828,6	827,0	815,1
767,4	812,5	795,4	813,1	806,4	827,0	827,1	816,0
771,2	814,3	796,8	812,2	809,7	830,6	825,6	816,9
772,7	812,3	799,3	815,2	812,5	832,5	825,4	815,6
717,3	753,8	753,3	767,2	775,7	791,1	797,1	792,4
718,8	753,3	749,9	766,5	780,0	794,6	799,9	793,3
721,5	754,0	749,8	767,3	781,5	796,4	801,9	793,2
719,9	754,7	752,7	769,0	778,6	796,4	801,3	791,7
719,7	754,3	756,3	771,0	778,3	797,6	800,7	793,4
723,9	755,5	759,3	772,0	783,5	799,8	800,3	796,6
725,6	757,3	758,2	771,6	785,5	799,0	799,9	796,2
724,0	755,7	751,5	769,7	780,3	791,5	798,6	792,3
796,3	834,8	824,6	844,4	831,0	852,3	852,3	837,9
793,1	830,6	822,6	839,5	830,4	848,2	850,8	835,7
790,5	829,8	820,9	840,5	827,2	847,4	848,1	835,3
792,4	831,2	823,9	838,0	830,4	845,5	847,7	836,5
786,8	830,9	829,0	837,2	831,6	852,1	848,7	841,7
743,3	767,2	789,8	795,7	806,0	819,0	826,2	821,6
735,4	764,0	782,5	791,9	801,3	819,9	822,6	819,2
728,0	762,2	775,4	787,0	796,0	818,0	817,3	814,3
725,1	762,4	772,1	783,8	792,6	814,3	813,4	811,5
726,3	764,1	772,5	782,0	791,4	813,7	814,3	811,1
735,1	774,2	781,0	790,0	796,5	818,2	819,2	814,2
736,4	777,8	785,2	796,3	798,7	819,0	818,8	814,7
735,7	776,8	785,8	798,5	797,9	817,3	815,7	813,0
737,5	779,6	787,5	802,3	798,5	818,8	815,6	813,9
774,6	816,6	800,5	821,7	817,1	836,1	838,4	826,2
770,4	812,3	796,8	814,5	813,4	828,6	831,4	816,4
772,0	812,6	797,2	815,0	812,0	827,3	828,9	815,2
774,4	815,8	798,1	816,3	811,7	829,0	829,1	817,6
769,0	814,2	796,3	812,7	809,1	828,6	827,0	815,1
767,4	812,5	795,4	813,1	806,4	827,0	827,1	816,0
771,2	814,3	796,8	812,2	809,7	830,6	825,6	816,9
750,9	776,5	785,7	797,9	797,0	816,6	815,0	813,4

Fark = Hesaplanan - Gerçek							
-4,1	1,9	-4,2	8,5	-9,8	-4,1	2,1	1,4
-2,3	-1,5	-4,4	2,0	-15,3	-8,6	-13,8	-10,2
2,0	3,8	-0,7	9,2	-11,6	-1,1	-6,8	-0,4
0,3	3,4	-1,1	8,7	-10,2	0,1	-4,3	0,8
-2,1	0,3	-2,0	7,4	-9,9	-1,5	-4,5	-1,5
2,4	1,8	-0,2	11,0	-7,3	-1,2	-2,4	0,9
4,0	3,5	0,7	10,5	-4,6	0,4	-2,5	0,0
0,2	1,8	-0,7	11,4	-7,9	-3,1	-1,0	-0,8
-1,3	3,8	-3,2	8,5	-10,7	-5,1	-0,8	0,5
9,4	12,8	2,9	14,7	-5,3	3,1	0,0	0,9
7,9	12,4	5,3	14,4	-11,4	-1,4	-3,8	-0,1
4,3	10,7	5,4	12,6	-13,0	-4,1	-6,8	-0,9
5,0	10,0	1,6	10,0	-11,0	-4,1	-6,2	0,6
4,2	9,5	-3,0	7,1	-11,7	-6,3	-6,5	-2,0
-1,0	7,3	-6,0	6,0	-16,8	-8,5	-6,1	-6,2
-2,7	4,6	-5,8	5,5	-19,8	-8,6	-6,7	-5,8
-2,0	5,3	-0,1	6,5	-14,6	-2,0	-5,3	-1,9
5,5	7,9	1,0	15,4	-0,7	7,4	7,4	18,1
8,7	12,0	2,9	19,3	-0,1	10,6	8,9	19,3
11,3	12,8	4,6	18,3	3,1	11,4	11,6	19,7
8,4	12,4	1,7	20,8	-1,0	13,3	11,1	18,5
14,0	12,7	-3,4	21,6	-2,2	5,8	10,1	12,3
18,6	34,6	7,2	29,8	-2,3	11,3	6,0	6,8
25,5	36,9	13,6	33,7	2,4	10,4	9,6	9,2
32,0	36,7	19,8	37,6	6,8	12,3	14,0	13,1
34,0	36,5	22,1	40,8	10,2	15,0	17,9	16,0
32,8	33,9	21,7	41,7	11,3	15,7	17,0	14,4
23,0	22,8	12,2	33,6	5,3	10,2	11,1	11,3
21,7	19,3	8,1	26,4	3,1	9,4	11,5	10,8
21,4	19,3	6,5	23,2	3,9	11,1	14,6	12,6
18,7	15,6	3,9	19,5	3,3	9,6	13,7	10,7
-10,8	-10,1	-10,1	-2,8	-16,2	-10,5	-13,8	-6,4
-6,6	-5,7	-6,4	4,4	-12,5	-3,0	-6,8	2,5
-8,2	-6,1	-6,8	3,9	-11,1	-2,7	-4,3	3,7
-11,6	-9,2	-7,7	2,6	-11,8	-3,4	-4,5	1,3
-6,1	-7,7	-6,9	6,2	-9,2	-4,0	-2,4	3,8
-5,5	-6,9	-5,0	4,8	-6,5	-2,4	-2,5	2,9
-9,3	-8,7	-7,4	5,7	-9,8	-6,0	-1,9	2,0
24,3	28,1	19,0	22,9	21,0	17,5	22,9	22,6

[650] – tansig – trainscg

Uygulanan matlab program aşağıdaki gibidir. Nöron hesabı formülüne göre $\alpha = 7$ seçilerek nöron sayısı 650 bulunmuştur.

Performans değerimiz ise;

$$\text{Performans (mse)} = \underline{2.17 \cdot 10^{-4}}$$

olarak bulunmuştur.

[760] – tansig – trainscg

Uygulanan matlab program aşağıdaki gibidir. Nöron hesabı formülüne göre $\alpha = 6$ seçilerek nöron sayısı 760 bulunmuştur.

Performans değerimiz ise;

$$\text{Performans (mse)} = \underline{2.72 \cdot 10^{-4}}$$

olarak bulunmuştur.

[910] – tansig – trainscg

Uygulanan matlab program aşağıdaki gibidir. Nöron hesabı formülüne göre $\alpha = 5$ seçilerek nöron sayısı 910 bulunmuştur.

Performans değerimiz ise;

$$\text{Performans (mse)} = \underline{2.29 \cdot 10^{-4}}$$

olarak bulunmuştur.

[1150] – tansig – trainscg

Uygulanan matlab program aşağıdaki gibidir. Nöron hesabı formülüne göre $\alpha = 4$ seçilerek nöron sayısı 1150 bulunmuştur.

Performans değerimiz ise;

$$\text{Performans (mse)} = \underline{2.44 \cdot 10^{-4}}$$

olarak bulunmuştur.

[1512] – tansig – trainscg

Uygulanan matlab program aşağıdaki gibidir. Nöron hesabı formülüne göre $\alpha = 3$ seçilerek nöron sayısı 1512 bulunmuştur.

Performans değerimiz ise;

$$\text{Performans (mse)} = \underline{2.79 \cdot 10^{-4}}$$

olarak bulunmuştur.

[2269] – tansig – trainscg

Uygulanan matlab program aşağıdaki gibidir. Nöron hesabı formülüne göre $\alpha = 2$ seçilerek nöron sayısı 2269 bulunmuştur.

Performans değerimiz ise;

$$\text{Performans (mse)} = \underline{3.25 \cdot 10^{-4}}$$

olarak bulunmuştur.

[2269] – traingdx

Uygulanan matlab program aşağıdaki gibidir. Nöron hesabı formülüne göre $\alpha = 2$ seçilerek nöron sayısı 2269 bulunmuştur.

Performans değerimiz ise;

$$\text{Performans (mse)} = \underline{4.47 \cdot 10^{-4}}$$

olarak bulunmuştur.

[570] – tansig – traincgf

Uygulanan matlab program aşağıdaki gibidir. Nöron hesabı formülüne göre $\alpha = 8$ seçilerek nöron sayısı 570 bulunmuştur.

Performans değerimiz ise;

$$\text{Performans (mse)} = \underline{1.36 \cdot 10^{-4}}$$

olarak bulunmuştur.

EK-11: Benzer Hat Ara Bobin Listesi (9 Aylık)

URETİM TARİHİ	KALINLIK	GENISLIK	KALITE	AGIRLIK	UZUNLUK
13.01.2016	0.3	964	2244K	3950	1781
15.01.2016	0.254	905	2340K	4990	2816
15.01.2016	0.252	905	7112K	4820	2723
15.01.2016	0.301	1051	7112K	5390	2205
16.01.2016	0.253	905	2244K	4680	2646
16.01.2016	0.299	964	2244K	3750	1691
16.01.2016	0.367	1025	7112K	4890	1679
17.01.2016	0.252	905	2344K	4500	2541
17.01.2016	0.225	934	2244K	4880	3015
17.01.2016	0.226	934	2228K	4710	2911
19.01.2016	0.302	1050	2344K	5200	2104
19.01.2016	0.301	1054	7112K	4900	2006
23.01.2016	0.324	1063	7112K	5700	2148
26.01.2016	0.304	1054	2228K	4680	1919
26.01.2016	0.326	1063	2228K	5490	2074
26.01.2016	0.403	925	2228K	3520	1229
27.01.2016	0.252	905	7112K	4000	2449
27.01.2016	0.255	900	2228K	5310	3022
27.01.2016	0.304	950	2344K	5770	2596
28.01.2016	0.3	1054	7112K	4500	1847
28.01.2016	0.324	1063	7112K	5280	1997
28.01.2016	0.399	918	7112K	4300	1499
28.01.2016	0.498	788	7112K	4360	1419
29.01.2016	0.352	934	7112K	4750	1865
30.01.2016	0.253	905	2344K	4180	2368
30.01.2016	0.404	918	7512K	4120	1434
30.01.2016	0.395	954	7112K	3870	1329
30.01.2016	0.5	788	2228K	4050	1325
31.01.2016	0.296	1054	2242K	4270	1757
31.01.2016	0.226	934	2228K	4600	2843
31.01.2016	0.226	934	2227K	4560	2762
08.02.2016	0.255	855	2227K	5290	3175
08.02.2016	0.304	1054	2228K	4070	1673
09.02.2016	0.228	934	2241K	4320	2674
11.02.2016	0.227	934	2220K	4080	2524
12.02.2016	0.254	855	2220K	5090	3052
13.02.2016	0.275	908	2220K	5640	2965
13.02.2016	0.275	980	2220K	5590	2706

28.02.2016	0.272	980	7112K	0	1863
01.03.2016	0.366	1026	7112K	4750	1639
01.03.2016	0.398	878	7112K	5590	2073
01.03.2016	0.497	1004	7112K	2960	769
03.03.2016	0.271	900	7112K	5540	2932
03.03.2016	0.271	908	7112K	4470	2360
03.03.2016	0.272	908	7112K	3920	2070
03.03.2016	0.397	878	7112K	5390	1999
03.03.2016	0.252	855	7112K	4620	2771
05.03.2016	0.273	900	2344K	5420	2846
05.03.2016	0.227	854	2344K	4730	3179
05.03.2016	0.227	854	2344K	4560	3071
06.03.2016	0.274	900	2228K	5100	2713
06.03.2016	0.228	854	2228K	4450	2993
06.03.2016	0.226	854	2241K	5000	2904
07.03.2016	0.27	900	7112K	4930	2627
07.03.2016	0.201	854	2242K	3630	2691
07.03.2016	0.225	854	7112K	4190	2821
08.03.2016	0.303	901	2228K	4600	2196
08.03.2016	0.4	878	2228K	5230	1924
08.03.2016	0.221	878	2241K	5730	3861
30.03.2016	0.22	878	7512K	4550	3064
30.03.2016	0.264	1002	7112K	4220	2061
30.03.2016	0.362	1050	7512K	4840	1667
03.04.2016	0.22	878	2244K	4270	2879
06.04.2016	0.219	878	2244K	4130	2782
06.04.2016	0.219	878	2244K	4220	2849
07.04.2016	0.219	878	2228K	4080	2758
07.04.2016	0.251	903	2228K	4370	2515
08.04.2016	0.289	890	2344K	5420	2720
08.04.2016	0.219	878	2241K	3990	2690
08.04.2016	0.219	878	2241K	3920	2649
09.04.2016	0.287	890	2228K	4160	2091
03.05.2016	0.22	879	2241K	5110	3455

06.05.2016	0.219	879	2344K	4970	3360
07.05.2016	0.251	906	7112K	5510	3125
08.05.2016	0.323	1063	7112K	3480	1288
08.05.2016	0.273	908	2243K	4950	2610
08.05.2016	0.214	879	2243K	4830	3266
08.05.2016	0.238	1008	2244K	5680	3105
10.05.2016	0.219	879	2344K	4700	3176
11.05.2016	0.219	879	2344K	4560	3088
12.05.2016	0.214	936	7112K	5500	3552
12.05.2016	0.219	879	2228K	4420	2991
12.05.2016	0.251	906	7112K	5340	3041
12.05.2016	0.347	1056	7112K	4140	1466
13.05.2016	0.297	1006	7112K	5110	2213
13.05.2016	0.299	1006	7113K	4770	2080
13.05.2016	0.324	1063	7113K	3150	1204
13.05.2016	0.396	878	7123K	2820	1062
13.05.2016	0.393	878	7112K	2570	967
04.06.2016	0.3	1050	7112K	3890	1599
04.06.2016	0.214	936	7112K	4890	3162
04.06.2016	0.313	878	7112K	4180	1977
05.06.2016	0.313	878	7112K	4010	1898
05.06.2016	0.346	1050	7112K	5520	1946
05.06.2016	0.348	1056	7112K	5730	2019
05.06.2016	0.356	1050	7112K	3980	1375
06.06.2016	0.314	878	2344K	3820	1808
07.06.2016	0.336	897	2242K	4870	2068
07.06.2016	0.219	879	2228K	4170	2824
08.06.2016	0.303	1050	2228K	3660	1509
12.06.2016	0.344	897	2228K	4580	1945
12.06.2016	0.316	878	2228K	3620	1714
12.06.2016	0.356	1033	2228K	4230	1655
01.07.2016	0.347	1056	7112K	4230	1497
02.07.2016	0.254	900	2228K	5190	2990
02.07.2016	0.247	900	2242K	5050	2913
02.07.2016	0.216	936	2227K	4620	2989
03.07.2016	0.272	895	2344K	5980	3121
03.07.2016	0.216	932	2344K	4190	2711
03.07.2016	0.217	936	2228K	4350	2814

03.07.2016	0.238	988	2228K	5630	3106
04.07.2016	0.216	932	2228K	4040	2616
10.07.2016	0	900	2228K	4080	2355
10.07.2016	0.285	890	7512K	5490	2832
10.07.2016	0.27	955	7512K	3940	1965
10.07.2016	0.289	955	7512K	4490	2138
10.07.2016	0.352	1056	7512K	5070	1790
05.08.2016	0.398	809	7112K	4350	1740
05.08.2016	0.287	955	7112K	3310	1576
05.08.2016	0.347	1056	7112K	5190	1804
05.08.2016	0.356	900	7112K	3180	1286
06.08.2016	0.227	850	2241K	4250	2860
06.08.2016	0.301	809	2344K	4800	2566
06.08.2016	0.3	809	2344K	4650	2487
07.08.2016	0.284	890	2227K	4080	2111
07.08.2016	0.227	850	2227K	4100	2762
07.08.2016	0.227	850		3980	2680
07.08.2016	0.227	850	2227K	3850	2593
07.08.2016	0.301	809	2227K	4540	2388
08.08.2016	0.398	809	7112K	4170	1667
08.08.2016	0.299	809	7112K	4310	2308
09.08.2016	0.253	906	2227K	5830	3317
01.09.2016	0.319	874	7112K	5650	2632
01.09.2016	0.349	1056	7112K	3580	1269
02.09.2016	0.318	874	7112K	5480	2551
02.09.2016	0.349	1056	7112K	3360	1192
03.09.2016	0.268	1000	2344K	4920	2403
03.09.2016	0.279	911	2344K	5220	2690
03.09.2016	0.362	911	2344K	4500	1745
04.09.2016	0.253	906	2343K	4240	2425
04.09.2016	0.252	906	2344K	4080	2332
04.09.2016	0.2	860	2227K	3600	2731
06.09.2016	0.253	906	2228K	3920	2246
06.09.2016	0.237	988	2344K	3970	2194
06.09.2016	0.28	911	2228K	5040	2599
07.09.2016	0.304	808	7112K	4450	2357
07.09.2016	0.299	954	2344K	5130	2337
08.09.2016	0.398	809	7112K	2620	1055
08.09.2016	0.348	1056	7112K	3490	1234
08.09.2016	0.362	962	7112K	4430	1660

09.09.2016	0.366	900	2228K	5800	2251
09.09.2016	0.363	962	7112K	4180	1567
09.09.2016	0.365	962	7113K	3920	1478
09.09.2016	0.507	788	2228K	12620	1228
10.09.2016	0.253	906	2228K	3740	2143
10.09.2016	0.27	908	2228K	5780	3079
19.09.2016	0.268	908	7112K	3130	2977
19.09.2016	0.266	911	7112K	2500	1561
19.09.2016	0.266	1000	7112K	4750	2319
19.09.2016	0.277	911	7112K	4850	2500
19.09.2016	0.317	1056	7112K	4220	1626
20.09.2016	0.3	1050	2228K	3700	1543
20.09.2016	0.402	1070	2228K	4470	1350
21.09.2016	0.269	908	2344K	5410	2885
22.09.2016	0.27	908	2241K	5250	2800
22.09.2016	0.264	908	2242K	5110	2725
22.09.2016	0.3	954	2344K	4940	2253
22.09.2016	0	954	2241K	4630	2105
23.09.2016	0.363	900	7112K	5510	2170
23.09.2016	0.401	1070	7112K	4170	1263
25.09.2016	0.363	900	7112K	5280	2082
26.09.2016	0.298	954	7112K	4380	2002
27.09.2016	0.303	808	7112K	3770	1998
27.09.2016	0.354	1056	7512K	3180	1126
27.09.2016	0.366	900	7511K	4500	1958
27.09.2016	0.362	900	7112K	4730	1868

ÖZGEÇMİŞ

28 Mart 1966 yılında Samsun’da doğdu. İlk ve orta öğrenimini Samsun’da tamamladı. 1984 yılında İstanbul Teknik Üniversitesi Elektronik ve Haberleşme Bölümüne başladı ve 1989 yılında bitirdi. Aynı yıl PTT’nin haberleşme yatırımları dolayısıyla teknik bölümünde (Şu an ki adı Türk Telekom) işe başladı. 1990 yılında Kara Kuvvetleri Komutanlığı Disiplin Moral şubesinde Ordu Bilgi İşlem Subayı olarak askerlik görevini yerine getirdi. 1992 yılında askerlik sonrası başladığı PTT’deki görevinden istifa ederek 1993 yılında Ereğli Demir ve Çelik Fabrikalarının kapasite artırma yatırımları dolayısıyla ve kariyerini yazılım alanında geliştirmek amacıyla bu şirkete başvurdu ve işe başladı. Yeni soğuk haddehane yatırımlarında görev aldı. Daha sonraki soğuk haddehane modernizasyonları dolayısıyla şartname hazırlama, teklif değerlendirme ve devreye alma çalışmalarında sistem sorumlusu olarak çalıştı. 2009 yılında Sakarya Üniversitesi İşletme Fakültesi uzaktan eğitim programına dâhil oldu. 2010 yılında bitirdi. 2011 yılında Sosyal Bilimler Enstitüsü İşletme Anabilim dalı Üretim Yönetimi ve Pazarlama bilim dalında doktora eğitimine başladı. 2017 Temmuz ayında doktora eğitimini bitirdi. Halen Ereğli Demir ve Çelik Fabrikalarındaki çalışma hayatına devam etmektedir.