

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**ÇAMAŞIR MAKİNESİNDE ÇAMAŞIR YÜKÜNÜN KESTİRİMİ ÜZERİNE BİR
ÇALIŞMA**

**YÜKSEK LİSANS TEZİ
Reyhan TÜRK**

**Anabilim Dalı : Elektrik Mühendisliği
Programı : Elektrik Mühendisliği**

ŞUBAT 2010

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**ÇAMAŞIR MAKİNESİNDE ÇAMAŞIR YÜKÜNÜN KESTİRİMİ ÜZERİNE BİR
ÇALIŞMA**

**YÜKSEK LİSANS TEZİ
Reyhan TÜRK
(504041044)**

**Tezin Enstitüye Verildiği Tarih : 25 Aralık 2009
Tezin Savunulduğu Tarih : 09 Şubat 2010**

**Tez Danışmanı : Yrd. Doç. Dr. Özgür ÜSTÜN (İTÜ)
Diğer Jüri Üyeleri : Prof. Dr. Faik MERGEN (İTÜ)
Prof. Dr. Hacı BODUR (YTÜ)**

ŞUBAT 2010

ÖNSÖZ

Öncelikle, tez çalışmamda bana büyük destek veren, her konuyla ilgili bilgi birikimini benimle paylaşan Yük. Mühendis Kerem ERENAY'a ve Arçelik ARGE Güç Elektroniği Ailesi'ne teşekkürlerimi sunarım.

Tez çalışmamı yönlendiren ve çalışmalarımı takip eden, değerli hocam Yrd. Doç. Dr. Özgür ÜSTÜN'e teşekkür ederim.

"Endüstride Uygulama Destekli Tez Programı" kapsamında çalışmamın hayata geçirilmesini sağlayan ARÇELİK A.Ş. AR-GE Merkezi'ne ve Sn. Şemsettin EKSSERT'e teşekkür ederim.

Tez çalışmam ve tüm eğitim hayatım boyunca her zaman benim yanımda olan anneme, babama ve kardeşime teşekkür ederim.

ARALIK 2009

Reyhan TÜRK

İÇİNDEKİLER

Sayfa

ÖNSÖZ.....	iii
İÇİNDEKİLER.....	v
KISALTMALAR.....	vii
ÇİZELGE LİSTESİ.....	ix
ŞEKİL LİSTESİ.....	xi
SEMBOL LİSTESİ.....	xiii
ÖZET.....	xv
SUMMARY.....	xvii
1. GİRİŞ.....	1
2. LİTERATÜR ARAŞTIRMASI.....	5
2.1 Patent İncelemesi.....	5
3. REGRESYON ANALİZİ.....	11
3.1 Lineer Model.....	11
3.2 Matris Gösterimi.....	12
3.3 β Parametrelerinin Kestirimi.....	12
3.4 En Küçük Kareler Kestirimi.....	13
3.5 Neden $\hat{\beta}$ İyi Bir Kestirimdir?.....	14
4. YAPAY SİNİR AĞLARI.....	17
4.1 Giriş.....	17
4.2 Biyolojik Sinir Ağı.....	18
4.3 Yapay Sinir Ağlarının Kısa Tarihçesi.....	20
4.4 Yapay Sinir Ağlarının Uygulama Alanları.....	21
4.5 YSA'ların Özellikleri.....	22
4.5.1 Doğrusal olmama.....	23
4.5.2 Paralellik.....	23
4.5.3 Gerçeklenme kolaylığı.....	23
4.5.4 Yerel bilgi işleme.....	23
4.5.5 Hata toleransı.....	23
4.5.6 Öğrenebilirlik.....	24
4.5.7 Genelleme.....	24
4.5.8 Uyarlanabilirlik.....	24
4.5.9 Donanım ve hız.....	25
4.5.10 Analiz ve tasarım kolaylığı.....	25
4.6 Yapay Nöron Modeli.....	25
4.6.1 Girdiler.....	25
4.6.2 Ağırlıklar.....	26
4.6.3 Birleştirme fonksiyonu.....	26
4.6.4 Aktivasyon fonksiyonu.....	26
4.6.4.1 Doğrusal aktivasyon fonksiyonu.....	27
4.6.4.2 Sigmoid aktivasyon fonksiyonu.....	27
4.6.4.3 Tanjant hiperbolik.....	28
4.6.5 Çıktı.....	29
4.7 Yapay Sinir Ağlarının Sınıflandırılması.....	29
4.7.1 YSA'ların yapılarına göre sınıflandırılması.....	29

4.7.1.1 İleri beslemeli ağlar.....	29
4.7.1.2 Geri beslemeli ağlar.....	30
4.7.2 YSA'ların öğrenme algoritmalarına göre sınıflandırılması.....	31
4.7.2.1 Danışmanlı öğrenme.....	31
4.7.2.2 Danışmansız öğrenme.....	32
4.7.3 Çeşitli YSA yapıları ve öğrenme algoritmaları.....	33
4.7.3.1 Çok katmanlı perseptronlar.....	33
4.7.3.2 Radyal tabanlı sinir ağı.....	34
4.7.3.3 Hopfield ağı.....	35
4.7.3.4 Vektör kuantalama (Learning vector quantization ,LVQ).....	37
4.7.3.5 Kohonen ağı.....	38
4.7.3.6 Elman ve Jordan ağları.....	39
4.8 Geri Yayılım Algoritması.....	41
4.8.1 Delta kuralı.....	41
4.8.1.1 Sigmoidal aktivasyon fonksiyonuyla delta kuralı.....	43
4.8.2 Genelleştirilmiş delta kuralı.....	44
4.8.3 Delta-Bar-Delta.....	47
4.8.4 Genişletilmiş Delta-Bar-Delta.....	49
4.8.5 Hızlı yayılım.....	51
5. ÇAMAŞIR YÜKÜ TAHMİN MODELİ.....	53
5.1 Çamaşır Yükünü Belirleme Profili.....	53
5.2 Çamaşır Makinesinde Yük Kestirimi İçin Kullanılan Verilerin İncelenmesi.....	55
5.2.1 Dengesiz yük.....	58
5.2.2 Çıkış zamanı.....	63
5.2.3 Sürtünme.....	65
5.2.4 İniş zamanı.....	66
5.2.5 Gerilim.....	67
6. ÇAMAŞIR MAKİNESİNDE ÇAMAŞIR YÜKÜ KESTİRİMİNİN GERÇEKLENMESİ	69
6.1 Giriş.....	69
6.2 Lineer En Küçük Kareler ile Yük Kestirimi.....	71
6.3 İleri Beslemeli Geri Yayılım Ağı ile Yük Kestirimi.....	74
6.3.1 Verilerin hazırlanması.....	75
6.3.2 YSA algoritmasının bilgisayar ile benzetimi.....	76
6.3.3 YSA'nın mikrodenetleyici ile gerçekleşmesi.....	79
6.4 Sonuçlar ve Karşılaştırmalar.....	81
7. SONUÇ VE ÖNERİLER.....	83
KAYNAKLAR.....	85
EKLER.....	87
ÖZGEÇMİŞ.....	93

KISALTMALAR

YSA	: Yapay Sinir Ađı
EKK	: En küçük kareler
ADALINE	: Genlikte Sürekli Algılayıcı
MADALINE	: Multiple ADALINE
SOM	: Öz Denetimli Harita
MLP	: Çok Katmanlı Perseptron
LVQ	: Vektör Kuantalama
BP	: Geri Yayılım
DBD	: Delta Bar Delta
GDBD	: Genişletilmiş DBD
RTFA	: Radyal Tabanlı Fonksiyon Ağları
VLSI	: Büyük Ölçekli Entegre Devre
İE	: İşlemci Elemanı
DY	: Dengesiz Yük

ÇİZELGE LİSTESİ

Sayfa

Çizelge 6.1 : Değişen gerilimde kestirim ortalamaları ve standart sapmaları	72
Çizelge 6.2 : Sabit gerilimde kestirim ortalamaları ve standart sapmaları	73
Çizelge 6.3 : Yük kestirim profili sonucunda bir makineden elde edilen veri örnekleri	76
Çizelge 6.4 : Değişen gerilimde kestirim ortalamaları ve standart sapmaları	78
Çizelge 6.5 : Sabit gerilimde kestirim ortalamaları ve standart sapmaları	78
Çizelge 6.6 : Sigmoid fonksiyonu için 128 noktalık tablo kullanılan YSA ile sabit gerilimde tahmin ortalamaları ve standart sapmaları	80
Çizelge 6.7 : Sigmoid fonksiyonu için 128 noktalık tablo kullanılan YSA ile değişen gerilimde tahmin ortalamaları ve standart sapmaları	81
Çizelge 6.8 : Değişen gerilimde YSA ve EKK ile kestirim ortalamaları	82
Çizelge 6.9 : Değişen gerilimde YSA ve EKK ile kestirim ortalamalarının standart sapmaları	82

ŞEKİL LİSTESİ

Sayfa

Şekil 2.1 : Farklı yüklerde sabit torkta ivmeler farklı olur [2].	6
Şekil 2.2 : Yük tahmin profili [3].	7
Şekil 2.3 : Yük kestirim profili [4].	8
Şekil 2.4 : Sensörün(9) konumu amortisöre paraleldir [5].	9
Şekil 3.1 : β kestiriminin geometrik ifadesi. Veri (data) vektörü Y , X 'in gerdiği model uzayına ortogonal olarak yansıtılmıştır (projeksiyon). Kestirim, $\hat{y} = X\hat{\beta}$ ifadesiyle gösterilmektedir ve kestirim ile veriler arasındaki fark residual vektörü $\hat{\epsilon}$ ile gösterilmiştir.	13
Şekil 4.1 : Biyolojik sinir sisteminin blok şeması [9].	18
Şekil 4.2 : Sinir hücrelerinin basitleştirilmiş şematik yapısı [8].	19
Şekil 4.3 : Genel bir kimyasal sinapsın şematik görüntüsü [8].	20
Şekil 4.4 : Doğrusal aktivasyon fonksiyonu.	27
Şekil 4.5 : Sigmoid aktivasyon fonksiyonu.	28
Şekil 4.6 : Tanjant hiperbolik aktivasyon fonksiyonu.	28
Şekil 4.7 : Bir gizli katmanı olan YSA.	30
Şekil 4.8 : Geri beslemeli ağı blok şeması [9].	31
Şekil 4.9 : Danışmanlı öğrenme yapısı [9].	32
Şekil 4.10 : Danışmansız öğrenme yapısı [9].	32
Şekil 4.11 : Çok katmanlı perseptron modeli.	34
Şekil 4.12 : Hopfield ağı.	35
Şekil 4.13 : LVQ ağı [9].	37
Şekil 4.14 : Kohonen ağı [9].	39
Şekil 4.15 : a) Elman ağı ve b) Jordan ağı [9].	40
Şekil 5.1 : Yük kestirim profili.	54
Şekil 5.2 : Yük belirleme profili akış diyagramı.	54
Şekil 5.3 : Çamaşır makineleri boşken elde edilen iniş süreleri.	55
Şekil 5.4 : Homojen dağıldığı varsayılan çamaşırın basit 2 boyutlu modeli.	57
Şekil 5.5 : Tambur içindeki çamaşırın kütle-eylemsizlik ilişkisi (mavi eğri).	57
Şekil 5.6 : Farklı dengesiz yüklerde sabit hız kontrolünde salınım.	58
Şekil 5.7 : Çamaşır makinesinin iki boyutlu düzlemde fiziksel modeli [14].	59
Şekil 5.8 : Deplasman miktarının farklı dengesiz yüklerle açısal hıza göre değişimi [14].	61
Şekil 5.9 : Sabit yükte dengesiz yük miktarı ile iniş zamanı arasındaki ilişki	62
Şekil 6.1 : Motor kontrol kartı ve MPLAB® ICE 2000 In-Circuit Emulator	70
Şekil 6.2 : Deney ortamının genel görünümü	70
Şekil 6.3 : Değişen gerilimde EKK ile yük kestirim sonuçları; siyah eğri gerçek yük miktarını, kırmızı eğri ise EKK ile yük kestirimini göstermektedir.	72
Şekil 6.4 : Sabit gerilimde EKK ile yük kestirim sonuçları; siyah eğri gerçek yük miktarını, kırmızı eğri ise EKK ile yük kestirimini göstermektedir.	73
Şekil 6.5 : Çamaşır yükü kestiriminde kullanılan 5x6x1'lik çok katmanlı perseptron ağı	75
Şekil 6.6 : Değişen gerilimde YSA ile yük tahmin sonuçları; kırmızı eğri gerçek yük miktarını, mavi eğri ise YSA ile yük kestirimini göstermektedir.	77

Şekil 6.7 : Sabit gerilimde YSA ile yük tahmin sonuçları; kırmızı eğri gerçek yük miktarını, mavi eğri ise YSA ile yük kestirimini göstermektedir.	78
Şekil 6.8 : Sigmoid fonksiyonu tablosu.....	80

SEMBOL LİSTESİ

X	: Serbest deęişken vektörü (regresyon analizi)
Y	: Baęımlı deęişken vektörü (regresyon analizi)
$\beta_1, \beta_2 \dots$: Regresyon katsayıları
ε	: Hata işareti
H	: Projeksiyon matrisi
x	: YSA girişı
w	: Aęırlıklar
net	: Lineer toplayıcı çıkışı
v	: Aktivasyon fonksiyonu girişı
f	: Aktivasyon fonksiyonu
O	: YSA çıkışı
y	: İstenen çıkış
E	: Ani hata
δ	: Yerel gradyen
η	: Öğrenme hızı
α	: Momentum terimi
o	: Ara katman çıkışı
J	: Eylemsizlik
M	: Çamaşırın kütlesi
R_1	: Tambur merkezinin çamaşıra olan en yakın uzaklığı
R_2	: Tamburun yarı çapı
m	: Dengesiz yük miktarı
c	: Sönümlleme katsayısı
k	: Direngenlik katsayısı
T	: Tork
B	: Vizkoz sürtünme katsayısı
Ω	: Tamburun açısal hızı
ω	: Şebeke frekansı
V_{in}	: Giriş gerilimi
τ	: Zaman sabiti
V_{max}	: Gerçek deęerlerin en büyük olanı
V_{min}	: Gerçek deęerlerin en küçük olanı
A_{max}	: En büyük normalizasyon deęeri
A_{min}	: En küçük normalizasyon deęeri
V	: Normalize edilecek deęer
A	: Normalize edilmiş deęer

ÇAMAŞIR MAKİNESİNDE ÇAMAŞIR YÜKÜNÜN KESTİRİMİ ÜZERİNE BİR ÇALIŞMA

ÖZET

Çamaşır makinesinde yıkama prosesi başlamadan önce, makinenin içinde ne kadar çamaşır olduğunun bilinmesi, yıkama prosesinin verimini maksimum yapabilmek için gereklidir. Su tasarrufu, enerji tasarrufu, deterjan tasarrufu, uygun program seçimi yük miktarı bilindiği takdirde yapılabilir.

Bu tez çalışmasında, 5 adet çamaşır makinesinden elde edilen verilerin bir kısmı kullanılarak geriye yayılım algoritması ile eğitilen çok katmanlı yapay sinir ağı ile çamaşır makinesinde yük kestirimi yapılmıştır. Bunun yanı sıra lineer en küçük kareler yöntemi ile de yük kestirimi yapılmıştır. Bu iki yöntem incelenip karşılaştırılmıştır.

Çamaşır makinesinde yük kestirimi için, sinir ağı modeli olarak ileri beslemeli bir gizli katmanlı perseptron ağı kullanılmıştır. Öğrenme metodu olarak ise momentumlu geri yayılım algoritması kullanılmıştır. Ağın giriş katmanında 5 adet nöron bulunmaktadır, orta katmanda 6 nöron ve çıkış katmanında 1 nöron vardır. Giriş örüntüleri ağa uygulanmadan önce öğrenmeyi hızlandırmak amacıyla, giriş örüntüleri normalize edilmiştir.

Lineer en küçük kareler algoritmasıyla yapay sinir ağı algoritması önce MATLAB ortamında gerçekleştirilmiş ve karşılaştırılmıştır. Daha sonra, yapay sinir ağının mikrokontrolörle uygulaması yapılmış ve performansı değerlendirilmiştir.

Sonuçlar değerlendirildiğinde, giriş örüntülerinin gerçek çıkış değerleriyle lineer olmayan ilişkisi nedeniyle yapay sinir ağıyla kestirim sonuçları, lineer en küçük kareler ile kestirim sonuçlarından daha tatmin edici bir sonuç vermiştir. Ancak yapay sinir ağı modelini mikrokontrolörle gerçekleştirmek daha zordur. Lineer en küçük kareler yöntemi basit yapısıyla bu noktada avantaj sağlamaktadır. Ancak bu zorluklara rağmen yapay sinir ağı uygulanabilir görünmektedir.

A STUDY ON ESTIMATION OF WEIGHT OF LAUNDRY IN WASHING MACHINE

SUMMARY

In order to succeed better washing performance, it becomes essential to know the weight of laundry placed into the washing machine. It will be possible to save water, energy, detergent if the amount of laundry is known because it will be possible for washing machine to select the best washing program, without any intervention.

An artificial neural network which is trained with error back propagation algorithm is builded to implement the estimation of amount of laundry. In order to analyze the performance of the neural network algorithm, laundry weight estimation is performed with least mean squares estimation algorithm. For the training and testing purposes, a data set of 5 washing machines is builded.

A multilayer perceptron with one hidden layer is used to perform the estimation of laundry in the washing machine. Network has 5 input, 6 hidden nodes and one output node. In order to perform faster learning, the input patterns are normalized.

Both the artificial neural network algorithm and linear least squares estimation are built in MATLAB programming environment. Then the implementation of artificial neural network, that was trained at MATLAB environment, to microcontroller is done.

The performance of neural network seems better compared to linear least mean squares algorithm. However, it is found that implementing a trained neural network to a traditional microcontroller is not as straightforward as least squares algorithm is. Despite of some difficulties existing in implementation of artificial neural network algorithm, it is worth while.

1. GİRİŞ

Modern Dünyada gelişen teknolojiyle birlikte özellikle şehirlerde yaşayan insanların gündelik hayatı çok kolaylaştı. Bunun başlıca sebeplerinden biri icat edilen beyaz eşyalardır: çamaşır makinesi, fırın, bulaşık makinesi, buzdolabı, elektrikli süpürge vs. Beyaz eşyalar sayesinde gündelik veya haftalık yapılması gereken rutin işler (temizlik, yemek pişirmek, gıdaların muhafaza edilmesi vs.), daha kısa zamanda ve çok az insan emeği gerektirerek yapılabilmektedir. Ayrıca beyaz eşyanın sağladığı hizmet kalitesi, yoğun insan emeği gerektiren, geleneksel yöntemlerden daha iyidir.

İnsan hayatına giren beyaz eşyalar, sağladıkları konforun karşılığında insana ve doğaya belirli bir maliyet sunmaktadırlar. Bu maliyet hem ekonomik (parasal) hem de çevre kirliliği olarak karşımıza çıkmaktadır. Beyaz eşya sektörü rekabetin dünya çapında yoğun yaşandığı bir sektördür. Dolayısıyla üretici firmalar arası rekabet, hem ürünün maliyetinin hem de ürününün veriminin artırılmasına yönelik ARGE faaliyetlerini zorunlu kılmaktadır. Ayrıca çevre sorunlarının gittikçe artması nedeniyle, beyaz eşya ürünlerinin doğaya minimum zarar verecek şekilde tasarlanması önem kazanmaktadır. Ekonomik ve çevresel maliyeti azaltmak ve beyaz eşyayı buna göre sınıflandırmak için çeşitli beyaz eşya standartları saptanmıştır. Belirli aralıklarla çeşitli kriterlere göre bu standartlar yeniden tanımlanmaktadır.

Çamaşır makinesi uzun zamandır modern insanın gündelik hayatında yer almaktadır. Sağladığı konfor ile gündelik hayatın vazgeçilmez bir ürünü olmuştur. Türkiyede neredeyse her evde bir tane bulunmaktadır. Bu kadar yaygın bir ürün için üretici firmalar sürekli rekabet halindedir. Üreticilerin, beyaz eşya piyasasında yer edinebilmeleri için yıkama kalitesi yüksek, enerji ve su verimi yüksek, maliyeti düşük, güvenilir çamaşır makineleri üretmeleri gereklidir.

Çamaşır makinesinin temel işlevi, yıkama prosesini gerçekleştirmektir. Yıkama kalitesini etkileyen faktörler şunlardır:

- Mekanik hareket
- Makinenin aldığı suyun miktarı
- Deterjan miktarı ve kalitesi

- Kirin miktarı ve cinsi
- Çamaşırın cinsi

Çamaşır yıkama prosesinde çamaşır makinesinin doğrudan etki ettiği faktörler mekanik hareket ve alınan su miktarıdır. Makineye yerleştirilecek olan çamaşırın cinsi, kiri ve deterjan çamaşır makinesinin müdahalesinin mümkün olmadığı faktörlerdir. Önden yüklemeli çamaşır makinelerinin yıkama adımlarında makinenin yaptığı mekanik hareketler birbirine çok benzemektedir. Çamaşır genellikle 50-60 min⁻¹(rpm) hız aralığında yıkanır. Belirli sürelerle, seçilen programa göre saat yönünde ve saatin tersi yönünde çamaşır makinesinin tamburu belirlenen hızlarda döner. Yıkama sırasında amaç çamaşırı tamburun tepe noktasına kadar çıkarıp düşürmektir. Bu hareket çamaşırdaki kirin sökülmesine büyük katkı yapar [1].

Çamaşırdaki kirin sökülmesi için alınan suyun miktarı ve kimyasal yapısı büyük önem taşımaktadır. Suyun kimyasal yapısı, kirin suda deterjanla birlikte çözünme kapasitesini belirlemektedir. Suyun miktarı ise deterjan ve kirin birleşmesinden oluşan moleküllerin çözünmesi ve kirin tekrar çamaşıra yapışmadan atılması için önem taşımaktadır. Herhangi bir çamaşır makinesinde, belirli bir yıkama kalitesini tutturmak için makinenin maksimum kapasitesi kadar çamaşırın en verimli yıkanacağı miktarda su makineye alınır. Bunun anlamı, çamaşır makinesi en verimli yıkamayı kapasitesinin maksimumuna yakın dolulukta yapmaktadır. Beyaz eşya piyasasında birçok çamaşır makinesi modeli vardır; bunların her birinin alacağı su miktarı makinenin kapasitesine, sınıfına ve üretici firmaya göre değişkenlik gösterebilir.

Piyasada, su alma ve yıkama profilindeki farklılıktan dolayı aynı sınıfta ve aynı yıkama kapasitesinde olan çamaşır makinelerinden biri, diğer bir çamaşır makinesinden biraz daha iyi yıkama performansına sahip olabilir. Ancak hangi çamaşır makinesi olursa olsun, alınan su miktarı maksimum çamaşır yükü için optimize edilmektedir. Çamaşır yükü azaldıkça makineye alınan su miktarı azalmamaktadır. Bu durum, çamaşır yıkama kalitesini etkilememektedir; ancak alınan su miktarının düşük çamaşır yükleri için optimum olmayacaktır. Daha az su miktarının daha az çamaşır için en uygun olacağı açıktır. Su ve enerji verimi, makinedeki çamaşır miktarı azaldıkça düşecektir, çünkü vadedilen yıkama performansından ödün vermeden daha az çamaşırı daha az suyla yıkamak mümkündür.

Çamaşır makinesine, çamaşır miktarına göre su alımını gerçekleştirmek üretici firmaya piyasada çok önemli bir avantaj sağlayacaktır, çünkü yıkama kalitesinden

ödün vermeden, çamaşır makinesinin su ve enerji tüketimi önemli ölçüde azalacaktır. Bu durum da tüketicinin çamaşır makinesi satın alırken göz önünde bulunduracağı çok önemli bir kriterdir. Böyle bir gelişme, sadece tüketici ölçeğinde değil, ulusal ve uluslararası ölçekte de çok önemli olacaktır. Daha az elektrik enerjisi tüketilerek doğal kaynaklar daha tasarruflu kullanılmış olacaktır. Ayrıca dünyanın artan nüfusu ve gittikçe kirlenen kullanılabilir su kaynakları nedeniyle suyun değeri gittikçe artmaktadır. Dolayısıyla tasarruf edilen her litre su çok önemli olacaktır.

Çamaşır makinesinin, tambura yerleştirilen çamaşır miktarını kestirmesi ve yıkama programını buna göre belirlemesi kullanıcı müdahalesini de azaltma yönünde etki edecektir. Çamaşır miktarının kestirimi, çoğu çamaşır makinesinde az çamaşır yükü için bulunan yıkama programının seçimi kullanıcıya bırakılmadan, çamaşır miktarının ihtiyacı olan yıkama programı ve süresinin seçiminin, çamaşır makinesi tarafından yapılmasını olanaklı kılacaktır. Çamaşır miktarını kestiren bir donanım ve yazılıma sahip çamaşır makinesinin, maliyet farkı çok değilse, standart bir çamaşır makinesine tercih edileceği açıktır, çünkü daha az enerji ve su tüketerek kullanıcının bütçesine de katkısı olacaktır. Böyle bir teknolojiye sahip bir çamaşır makinesinin doğaya zararı da minimum olacaktır. İnsanların gittikçe çevre konusundaki duyarlılığının artması nedeniyle sadece bu sebeple bile çamaşır miktarına göre yıkama yapan bir çamaşır makinesi tercih edilebilir.

Bu çalışmada, çamaşır makinesinde, ek maliyet getirmeden, çamaşır yükünü algılama yöntemleri ve bunların gerçekleştirilebilirliği üzerinde durulacaktır. Ek malzeme maliyetinin olmaması, çamaşır yükünü algılamak için ek sensörler ve çeşitli donanımların kullanılmaması anlamına gelmektedir. Detaylı olarak irdelenecek olan yöntemlerde, çamaşır makinesine belirli hareketler yaptırılarak yükü kestirmek için kullanılacak veriler toplanmaktadır. Yük kestirim algoritmasıyla toplanan veriler değerlendirilerek çamaşır makinesinin içindeki yük kestirilmektedir.

İrdelenen yöntemlerde yük kestirimi için kullanılan temel veriler çamaşırın tambur içindeki eylemsizliğine sıkı sıkıya bağlıdır. Çamaşır yükünü kestirmek için kullanılan yöntemlerden biri en küçük kareler kestirimidir. Bu yöntem doğrusal bir yöntemdir ve genelde doğrusal sistemlerin bilinmeyen parametrelerini kestirmek için kullanılır. İlerideki bölümlerde inceleneceği gibi çamaşırın kütlesiyle çamaşırın tambur içindeki eylemsizliği arasındaki ilişki doğrusal değildir. Ancak belirli bir aralık için kısmen doğrusal bir ilişki olduğu kabul edilebilir. Çamaşırın kütlesiyle eylemsizliği arasındaki ilişkinin doğrusal olmadığı göz önünde bulundurularak yük kestirimi için doğrusal olmayan yöntemler kullanmanın daha iyi sonuçlar vereceği tahmin edilebilir. Bu çalışmada detaylı olarak irdelenecek konulardan biri de yapay sinir ağı ile çamaşır

yükünü kestirmektir. Yapay sinir ağı ile kestirim, özellikle lineer olmayan sistemlerin parametre kestiriminde sıklıkla kullanılan ve bu konuda oldukça başarılı bir yöntemdir.

Bu tezin 2. bölümünde, çamaşır makinesinde çamaşır yükünü algılamaya yönelik beyaz eşya üreticilerine ait çeşitli çalışmalar ve patentlerden bahsedilecektir. Yük algılamak için çeşitli yöntemler patentlerle birlikte ele alınacaktır.

Bu tezin 3. bölümünde, lineer regresyon kısaca anlatılacaktır. En küçük kareler (EKK) regresyonu ile parametre kestirimi tanıtılacaktır.

Bu tezin 4. bölümünde, yapay sinir ağlarının (YSA) tarihsel gelişimi, kullanıldığı alanlar ve özellikleri anlatılacaktır. Yapay sinir ağları son yıllarda yaşanan gelişmelerle önemi artan bir çalışma alanı haline gelmiştir.

Bu tezin 5. bölümünde, çamaşır makinesinin elektriksel sürüş sistemi ve dönen mekanik yapının dinamiği kısaca incelenmiştir. Çamaşır yükünün kestirimi için elde edilen veriler çamaşır makinesinin mekanik ve elektriksel yapısıyla doğrudan ilişkilidir. Yük kestirimi algoritması oluşturabilmek için ve hangi parametrenin yük kestirimine nasıl etki ettiğini anlamak için makinenin elektriksel sürüş sisteminin ve mekanik yapısının anlaşılması gerekmektedir.

Bu tezin 6. bölümünde, çamaşır yükünü kestiren YSA algoritmasının uygulaması ve lineer en küçük kareler ile kestirim yönteminin uygulaması incelenmiştir. Bahsi geçen doğrusal ve doğrusal olmayan yöntemlerin çamaşır makinesinde yük algılama uygulaması için birbirlerine göre üstünlükleri, özellikleri ve farklılıkları 6. bölümün sonunda değerlendirilmiştir.

Bu çalışmanın amacı, çamaşır makinesinde yük algılamak için kullanılabilecek en küçük kareler kestirimi gibi lineer yöntemlerin yanında son yirmi yılda önemli aşamalar kaydeden yapay sinir ağları ile yük kestirimini gerçekleştirmek, yapay sinir ağının çamaşır makinesinde yük algılamak için kullanılmasının ekonomik değerinin olup olmadığını araştırmaktır.

2. LİTERATÜR ARAŞTIRMASI

Beyaz eşya ürünleriyle ilgili araştırmalar büyük ölçüde üretici firmaların ARGE departmanlarında yapılmaktadır. Firmalar, rekabet edebilmek için yaptıkları çalışmaların, ürettikleri bilginin detaylı olarak bilinmesini istemezler. Bu nedenle, çamaşır makinesinde yük algılamayla ilgili makale, tez gibi akademik çalışmalar bulmak çok güçtür. Bu konuyla ilgili yapılan çalışmalar, yakın gelecekte beyaz eşya piyasasında rekabette ciddi avantaj sağlayacaktır. Konuyla ilgili bilgi edinmek ancak yapılacak bir patent araştırmasıyla mümkündür. Çamaşır makinesinde yük algılamayla ilgili yenilikleri takip etmek, patentleri incelemekle mümkündür. Patent içeriği konuyla ilgili ana hatları vermekle birlikte fazla detay içermez.

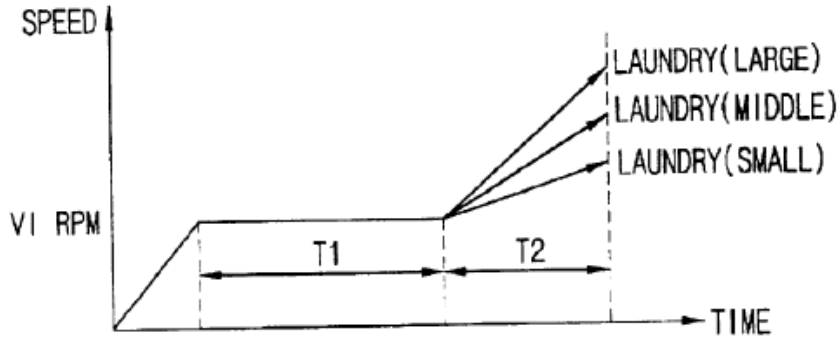
2.1 Patent İncelemesi

Bir firmanın faaliyet gösterdiği alanda piyasada yer bulabilmesi için, başarılı olması için alandaki yenilikleri takip etmesi ve yenilik üretmesi şarttır. Yenilikleri takip etmek ve yenilik yaratmak için ARGE faaliyetlerinin yapılması şarttır. Ancak bu da tek başına yeterli değildir; yapılan yeniliğin, üretilen bilginin korunması gerekir. Firmaların, kuruluşların, bireylerin yaptıkları buluş üzerindeki fikri hakları patentlerle korunur. Dolayısıyla yeni olduğu ve ticari değere sahip olduğu düşünülen fikirler patentlenebilir. Böylece bu fikirler patent haklarının sahibinin izni olmadan kullanılamazlar.

Patentler, esas olarak kuruluşların, kişilerin ticari fikir haklarını korumaya yöneliktir; ancak araştırmacılara, bir konuyla ilgili çalışma yapmak isteyenlere hem bilgi kaynağı hem de yeni fikir üretmek için esin kaynağı da olurlar. Bir konuyla ilgili çalışma yapan araştırmacıyı var olanın dışında fikirler, çözümler bulmaya iter. Bu yönüyle patent haklarının var olması yaratıcılığı teşvik eder.

Beyaz eşya firmaları yaptıkları birçok çalışmayı, buluşu korumak için yaptıkları buluşlarla ilgili patentler almaktadırlar. Bu çalışmanın konusu olan çamaşır makinesinde yük miktarını algılamak ile ilgili çeşitli firmaların, araştırmacıların patentleri vardır. Konuyla ilgili neler yapıldığını ve niçin yapıldığını anlamak için önemli birkaç tane patenti incelemek yararlı olacaktır.

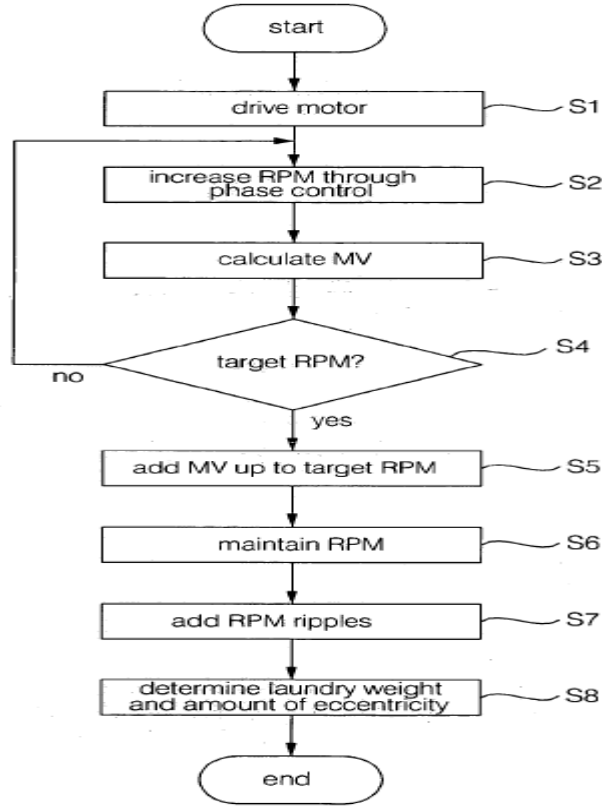
Soon Bae Yang, Kwan Yeul ve Chan Hee Hong'a [2] ait bir çalışmada sensörsüz fırçasız doğru akım motoruna (FDAM) sahip bir çamaşır makinesinde çamaşır ağırlığını algılamak için bir yöntem sunulmuştur. Bu yöntemde göre motorun hızı önceden belirlenen bir değere çıkarılıyor. Bahsedilen hız değeri, çamaşırın tambura yapıştığı minimum hızın üstündedir. Bir süre bu hızda döndükten sonra kontrol yöntemi değiştirilir ve hız kontrolünden tork kontrolüne geçilir. Motor, belirli bir süre (1-2 saniye) önceden belirlenmiş bir torku mekanik sisteme (tambur, çamaşır sistemine) aktarır. Bu sürenin sonunda motorun çıktığı hız ölçülür ve çamaşır makinesinin hafızasına kaydedilmiş hız tablosu kullanılarak tamburun içindeki çamaşır miktarı hesaplanır. Belirtilen yöntemle ilişkin bir fikir şekil 2.1'den edinilebilir.



Şekil 2.1 : Farklı yüklerde sabit torkta ivmeler farklı olur [2].

Bahsedilen yöntemin en önemli özelliği T1 süresinin sonunda tamburun belirli sabit tork ile ivmelendirilmesidir. Böylece şebeke gerilim değişimlerinden etkilenmeyeceği varsayılmıştır. T1 süresince RPM1 hızında sürüldekte sonra tork kontrolünün yapılmasının nedeni düşük hızlarda sensörsüz FDAM kontrolünde torkun ve hızın kestiriminde önemli hatalar olabileceği şeklinde açıklanmıştır. Bu yüzden önce belirli bir RPM1 hızına çıkılmaktadır.

Keol Choi Byung, Won Kim Dong ve Soo Lee Dong'un yaptığı çalışma [3] faz kontrolüne uygun bir motora sahip çamaşır makinesinde çamaşır yükünü kestirmek üzerinedir. Yükü kestirmek için önerdikleri algoritma şekil2.2'de gösterilmiştir.



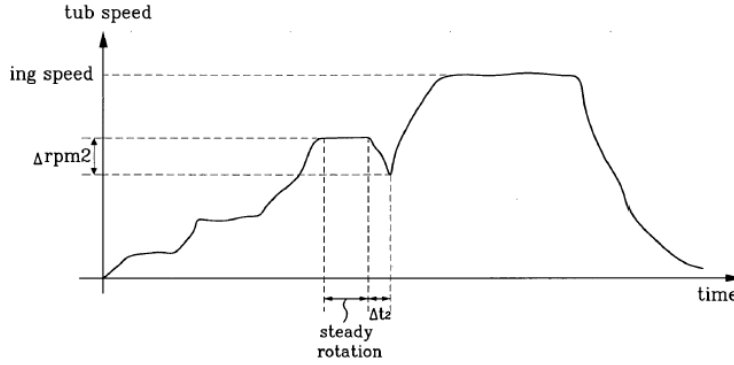
Şekil 2.2 : Yük tahmin profili [3].

Bahsedilen yöntemde, motor faz açısı kontrolü yapılarak belirli bir hıza çıkarılır. Motor hızlanırken uygulanan faz açısı değerleri işlemcide modifiye edilerek toplanır (modifikasyon şu şekildedir: Motora uygulanan şebeke geriliminin alanı hesaplanarak sinüs yarıdalgasına olan oranına göre bir değer atanır. Mesela şebeke sinüsünün yarısı 100'e karşılık gelir. Tetikleme 90 derecede yapıldıysa modifiye edilmiş değer 50 olur). Modifiye edilmiş gerilim MV olarak adlandırılmıştır. Motor referans hıza ulaştığında belirli sayıda turlar boyunca motor hızının referansa göre salınımları ölçülür ve bu genlikler toplanır. Hız salınım bilgisini edinmedeki amaç tambur içindeki çamaşırın dengesiz dağılımını saptamaktır. Çamaşır dağılımının dengesizliği, tamburun sıkma adımında yüksek hızlara çıkıp çıkamayacağını belirler. Büyük dengesiz yüke sahip bir makinenin yüksek sıkma hızlarına çıkması makinenin yıpranmasına yol açar ve güvenilirliğini azaltır. Bunun yanında, dengesiz yükün varlığı çamaşırın toplam yükünü kestirmeyi güçleştirmektedir.

Çamaşır miktarı çoksa motor belirlenen hıza çıkması daha fazla zaman alır ve dolayısıyla modifiye edilmiş gerilimlerin toplamı $\sum MV$ da daha büyük olur. Çamaşır miktarı azsa $\sum MV$ küçük olur. Ancak, aynı miktarda çamaşırın dengesiz dağılmış halinin $\sum MV$ 'ı, dengeli dağılmış halinin $\sum MV$ 'ına göre daha yüksektir, çünkü

dengesiz dağılmış yükü belirli bir hıza çıkarmak daha fazla enerji gerekir. Belirtilen çalışmada, $\sum MV$ 'nin dengesiz yüke göre kompanse edilmesi gerektiği vurgulanmış ve kompanzasyon $\sum MV - \sum HizHatasi$ yöntemiyle yapılmıştır.

Ho-Kim Yong ve Chol Lyu'nun yaptığı bir çalışmada [4], sıkma adımına geçilmeden önce çamaşır makinesindeki yükü ve dengesiz dağılmış yük miktarını belirlemek için bir yöntem önerilmiştir. Bu çalışmada, yük miktarını belirleyerek sıkmada hangi hızlara çıkılabileceğine karar verilebileceğinden bahsedilmiştir; eğer aşırı yük ya da yüksek dengesiz yük dağılımı tespit edilirse sıkma adımına geçilmeden çamaşır makinesinin durdurulabileceğinden bahsedilmiştir. Yük kestirim profili şekil 2.3'te verilmiştir.

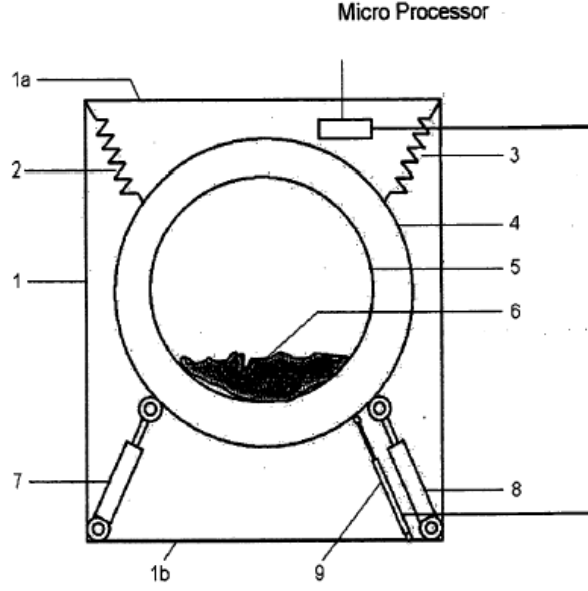


Şekil 2.3 : Yük kestirim profili [4].

Çamaşır makinesindeki yükü kestirmek için makinenin hızlanma ve yavaşlama profilinin kullanıldığından bahsedilmektedir. Yükü kestirmek için uygulanan adımlardan bahsedilmektedir. Önce motor belirli bir hızdan daha yüksek bir hıza sabit torkla hızlandırılır. Çıkış zamanı ölçülür. Daha sonra iniş zamanını daha yüksek çözünürlükte ölçebilmek için makine daha yüksek bir hıza çıkarılır. Bu hızdan serbest olarak veya frenleme yaptırılarak makinenin hızı belirli bir değere düşürülür. İniş zamanı ölçülür. Elde edilen veriler, çamaşır makinesinin hafızasında kayıtlı bulunan bir tablo ile değerlendirilerek yük kestirilir. Bu yöntemle elde edilen iniş ve çıkış zamanlarının, çamaşırın eylemsizliği ile doğrudan ilişkisi olduğundan ve yükün bu yöntemle doğru bir şekilde kestirilebileceğinden bahsedilmiştir.

Dominic Beier, Juergen Koschnike ve Diethard Krause'nin yaptığı çalışmada [5], çamaşır miktarını algılamak için deplasman sensörü kullanılmaktadır. Şekil 2.4'te gösterildiği gibi sensör çamaşır makinesinde amortisörlere paralel olarak kazanla zemin arasına yerleştirilmiştir. Tamburun içine çamaşır yerleştirildiğinde kazanın toplam ağırlığı değişecek ve kazan, askı - yay sistemine bağlı olduğu için içine

yerleřtirilen amařır miktarına gre bir miktar zemine dođru kecektir. kme miktarına gre sensrde iřaret retilecek ve amařır makinesinin iřlemcisine iletilecektir.



řekil 2.4 : Sensrn (9) konumu amortisre paraleldir [5].

amařır ađırlıđının dođru bir řekilde algılanabilmesi iin sensre kazan ađırlıđından bařka bir kuvvet etki etmemelidir. Bu nedenle amortisrler yađlı tip olmalıdır. Yađlı tip amortisrlerin uyguladıkları karřı kuvvet hızla ilintili olduđu iin amařır tambura yerleřtirildiđinde kuvvet etkisi sıfırdır. Fakat kuru tip amortisrler kullanılırsa, amařır tambura yerleřtirilince amortisrler karřı bir kuvvetle etki ederler. Bu da ykn dođru algılanmasını ok gleřtirmektedir.

3. REGRESYON ANALİZİ

Regresyon analizine ilişkin problemler ilk 18. y.y.'da astronomide gök cisimlerinin yörüngesini hesaplayabilmek için ortaya çıkmıştır. 1809 yılında Gauss "en küçük kareler" yönteminin, eğer regresyon hatası normal dağılım gösteriyorsa ve hatanın ortalaması sıfır ise, en optimal çözüm olacağını göstermiştir.

Regresyon analizi, bir Y değişkeninin (çıktı, bağımlı değişken ya da tepki olarak da adlandırılır) yine bir veya birden fazla giriş değişkeniyle (bağımsız değişken, prediktör ya da regresör de denir), X_1, \dots, X_p , olan ilişkisini modellemek ya da açıklamak için kullanılır. Eğer $p=1$ ise bu ilişkiye basit regresyon denir.

Regresyon analizi yapmanın birkaç hedefi olabilir. Bunlar,

- Bir sistemin olası gelecek davranışını kestirmek
- Giriş değişkenlerinin çıkış üzerindeki etkisini saptamak
- Veri yapısının genel bir tanımını yapmak

olabilir.

3.1 Lineer Model

Herhangi bir Y bağımlı değişkeninin genel modeli,

$$Y=f(X_1, X_2, X_3)+\varepsilon \quad (3.1)$$

ifadesiyle açıklanabilir. f , bilinmeyen herhangi bir fonksiyondur; ε ise modelleme hatasını temsil eder. Genellikle f fonksiyonunu doğrudan kestirmek için yeterli veri olmadığından, f 'in daha sınırlı bir ifade olduğunu, çoğunlukla da (3.2) ifadesi gibi lineer olduğunu kabul ederiz.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \varepsilon \quad (3.2)$$

β_i , $i=0,1,2,3$ bilinmeyen parametrelerdir. β_0 terimine itersept denir. Böylece, sonsuz boyutlu f yerine model 4 tane bilinmeyen parametrenin kestirimine indirgenmiş olur.

Lineer modelde parametreler lineer olarak ifade edilir, prediktörlerin lineer olması gerekmez. Mesela,

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 \log(X_2) + \varepsilon \quad (3.3)$$

lineerdir. Ancak,

$$Y = \beta_0 + \beta_1 X_1^{\beta_2} + \varepsilon \quad (3.4)$$

lineer değildir. Lineer modeller sınırlayıcı görünür, ancak prediktörlerin değişik kombinasyonları mümkün olduğu için ve dönüştürülebilir oldukları için aslında lineer modeller belirli bir esnekliğe sahiptir. Lineer olmayan modelleri kullanmak gerçekte nadiren gerekli olur.

3.2 Matris Gösterimi

Deneylerle ya da gözlemlerle gerçek veriler elde edildiğinde

$$Y_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \beta_3 X_{3i} + \varepsilon_i, \quad i=1,2,\dots,n \quad (3.5)$$

ve çok fazla veri olduğunda denklem (3.5)'teki gibi alt simge kullanmak zahmetli bir hal alacağından, matris veya vektör gösterimi kullanmak daha kullanışlı olacaktır. Regresyon denklemi

$$y = \beta X + \varepsilon \quad (3.6)$$

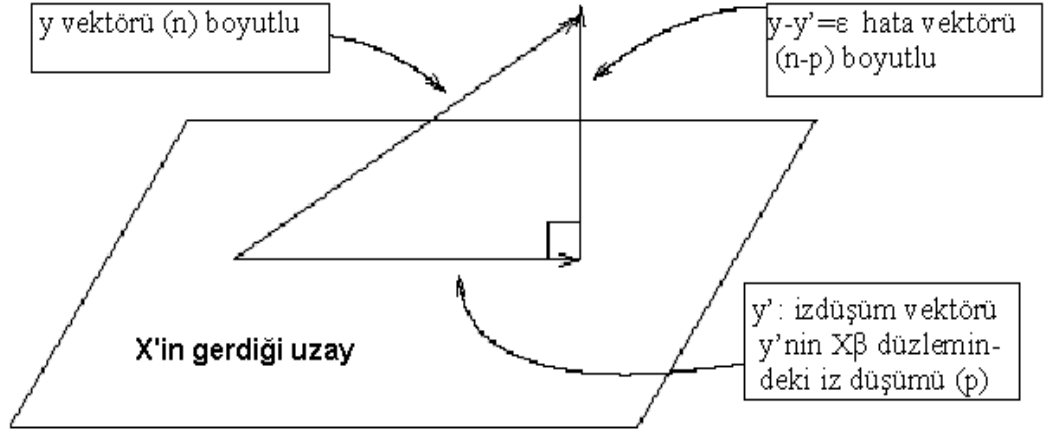
şeklinde yazılır, $y = (y_1 \dots y_n)^T$, $\varepsilon = (\varepsilon_1 \dots \varepsilon_n)^T$, $\beta = (\beta_0 \dots \beta_3)^T$ ve

$$X = \begin{pmatrix} 1 & x_{11} & x_{12} & x_{13} \\ 1 & x_{21} & x_{22} & x_{23} \\ \dots & & & \\ 1 & x_{n1} & x_{n2} & x_{n3} \end{pmatrix}$$

1'ler kolonu, intersept kolonu olarak adlandırılır.

3.3 β Parametrelerinin Kestirimi

Regresyon analizinin amacı (3.6) regresyon denkleminde $X\beta$ 'yi, tesadüfi ε 'den en iyi şekilde ayıracak β parametre vektörünü bulmak için ne yapılabilir sorusuna yanıt aramaktır. Soruya geometrik yaklaşımla bakılırsa, $y \in R^n$ iken $\beta \in R^p$, p parametre sayısıdır.



Şekil 3.1 : β kestiriminin geometrik ifadesi. Veri (data) vektörü Y , X 'in gerdiği model uzayına ortogonal olarak yansıtılmıştır (projeksiyon). Kestirim, $\hat{y} = X\hat{\beta}$ ifadesiyle gösterilmektedir ve kestirim ile veriler arasındaki fark residual vektörü $\hat{\epsilon}$ ile gösterilmiştir.

Problem, $X\beta$ vektörünü Y 'ye mümkün olduğu kadar yakın kılacak β 'yi bulmaktır. Şekil 3.1'de geometrik gösterimde en iyi β seçimi açıktır. X model uzayında en iyi β kestirimi $\hat{\beta}$ 'dir. Model tarafından kestirilen çıkış $\hat{y} = X\hat{\beta}$ veya $H\hat{y}$ ile gösterilir. H burada ortogonal projeksiyon matrisidir. Gerçek çıktı ile kestirilen çıktı arasındaki fark, residual vektörü, $\hat{\epsilon}$ ile gösterilir.

Modellemenin amacı, muhtemelen kompleks bir yapı olan n -boyutlu çıkış vektörü Y 'yi, p -boyutlu nispeten daha basit forma sahip bir ilişkiyle göstermektir. Eğer yapılan modelleme başarılıysa, verilerdeki yapı p -boyutta elde edilebiliyor olmalı ve modelin dışında sadece rastlantısal değişimler fazlalık(residual) olarak kalmalıdır. Bunu şu şekilde de ifade edebiliriz:

Veri = Sistemik Yapı + Rastlatısal Değişimler

3.4 En Küçük Kareler Kestirimi

β vektörünün kestirimine geometrik olmayan bir bakış açısıyla da yaklaşılabilir. β 'nin en iyi kestirimi, $\epsilon^T \epsilon$ hataların karelerinin toplamını minimum yapan $\hat{\beta}$ olarak tanımlanabilir. Şu ifade minimize edilerek bulunur:

$$\sum \epsilon^2 = \epsilon^T \epsilon = (y - X\beta)^T (y - X\beta) \quad (3.7)$$

Bu ifade açıldığında,

$$y^T y - 2\beta X^T y + \beta^T X^T X \beta \quad (3.8)$$

olur. Amaç bu ifadenin minimum olmasını sağlayacak β 'yı bulmak olduğuna göre, (3.8) ifadesinin β 'ya göre türevi alınıp sıfıra eşitlendiğinde bunu sağlayacak β vektörü elde edilebilir.

$$\frac{\partial(\varepsilon^T \varepsilon)}{\partial \beta} = -2X^T y + 2X^T X \hat{\beta} \quad (3.9)$$

ifadesi sıfıra eşitlendiğinde

$$-2X^T y + 2X^T X \hat{\beta} = 0 \quad (3.10a)$$

$$X^T y = X^T X \hat{\beta} \quad (3.10a)$$

ifadesi elde edilir. Bu denklemlere normal denklemler denir. bu sonuca geometrik yaklaşımla da ulaşmak mümkündür. Eğer $X^T X$ 'in tersi alınabiliyorsa,

$$\hat{\beta} = (X^T X)^{-1} X^T y \quad (3.11a)$$

$$X \hat{\beta} = X(X^T X)^{-1} X^T y = Hy \quad (3.11b)$$

$H = X(X^T X)^{-1} X^T$ matrisi, y 'nin X 'in geldiği uzay üzerindeki projeksiyonudur. H teorik manipülasyonlar yapmak için elverişli olsa da $n \times n$ bir matris olduğu için hesaplanması zordur[6].

3.5 Neden $\hat{\beta}$ İyi Bir Kestirimdir?

Eğer hatalar korelasyonsuzsa ve varyanslar, $\text{var } \varepsilon = \sigma^2 I$, eşitse; en küçük kareler kestirimi, β 'yı kestirmek için en iyi yollardan biridir. EKK'nin, en iyi kestirim yöntemlerinden biri olduğunu şu şekilde açıklayabiliriz:

- EKK'nin geometrik bir anlamı vardır: $\hat{\beta}$ ile kestirilen \hat{y} , y 'nin model uzayında ortogonal projeksiyonudur.
- Eğer hatalar bağımsız ve normal bir dağılım gösteriyorsa, $\hat{\beta}$ en büyük olabilirlik kestiricisidir(maksimum likelihood estimator).
- Gauss-Markov teoremi, en küçük kareler kestiriminin en iyi doğrusal yansız tahmin edici(lineer unbiased estimator) olduğunu gösterir.

Gauss-Markov teoremi, en küçük kareler kestirimiyle bulunan $\hat{\beta}$ 'nin, modelleme için iyi bir tercih olduğunu gösterir; ama hatalarda korelasyon varsa veya varyanslar eşit değilse, daha iyi kestirim yöntemleri vardır. Hatta hataların dağılımı normal bir dağılım değilse, o zaman lineer olmayan kestirim yöntemleri daha iyi sonuç verebilir. Dolayısıyla, Gauss-Markov teoremi, EKK'nin her zaman kullanılması gereken bir yöntem olduğunu göstermez, sadece tavsiye eder, eğer başka yöntemlerin kullanılması için güçlü nedenler yoksa.

EKK dışında kestirim yöntemlerinin düşünülmesi gereken durumlar:

1. Hatalarda korelasyon varsa ve varyanslar eşit değilse genelleştirilmiş en küçük kareler kestirim yöntemi düşünülebilir.
2. Hata dağılımı uzun kuyruk biçimindeyse, dayanıklı kestirim yöntemleri kullanılabilir.
3. Prediktörler (giriş değişkenleri) arasında yüksek korelasyon varsa, yanlı (biased) kestirim yöntemleri kullanılabilir.

Modelleme yaparken, hatayı belirli bir güven aralığının içinde tutmak ya da hipotez testi yapmak için hataların ε dağılımının şekline bakmak gereklidir. Genel kabul, hatanın normal (Gauss) dağılım gösterdiği şeklindedir; pratikte, her zaman olmasa da makul bir kabuldür [6].

4. YAPAY SİNİR AĞLARI

4.1 Giriş

Yapay sinir ağı sistemleri, biyolojik bir organizmanın beyninin ve sinir sisteminin çok basitleştirilmiş matematiksel modelleridir. Yapay sinir ağı, paralel dağıtılmış bilgisayar ağı gibi çalışır. Ancak, belirli işleri ve görevleri yerine getirmek için programlanan geleneksel bilgisayarlardan farklı olarak yapay sinir ağlarının öğrenme, sınıflandırma, genelleme, ilişkilendirme ve optimizasyon yapma gibi yetenekleri vardır. İnsan beynine benzer şekilde yeni ilişkileri, yeni bağıntıları, yeni örüntüleri başarılı bir şekilde öğrenme, ayırt etme yetenekleri vardır.

Bir tanıma göre yapay sinir ağı sistemleri, deneyimle elde edilen bilgiyi toplayan, depolayan ve kullanan fiziksel hücresel sistemlerdir [7]. Yapay sinir ağını matematiksel olarak modelleme ihtiyacı ve gün geçtikçe modellerin çeşitlenmesi ve gelişmesi; hem endüstride akıllı bir organizma gibi karar verebilen makinelere ihtiyaç duyulmasından hem de insan beyninin düşünme, öğrenme, yeni durumlara uyum sağlama gibi yeteneklerinin matematiksel bir yapıyla nasıl ifade edilebileceği sorusuna bazı araştırmacıların büyük ilgi göstermesi sayesinde ortaya çıkmıştır. Böylelikle dış müdahale olmadan kendi kendine karar verebilen, öğrenebilen yapıların ortaya çıkması mümkün olmuştur.

Günümüzde yapay sinir ağları, her ne kadar gerçek insan beyninin işlevselliğinden, öğrenme başarısından çok uzak olsalar da birçok uygulamada başarıyla kullanılmaktadır. Yapay sinir ağları, bilgisayar programı ile benzetimi yapılarak kullanılabilir ya da yapay sinir ağları uygulaması için özel tasarlanmış entegre devrelerle (fpga, vlsi gibi) gerçekleştirilebilir.

Birçok yapay sinir ağı modeli vardır ancak bu modellerin temelini teşkil eden nöron yapısı hepsinde hemen hemen aynıdır. Nöronların birbirleriyle bağlantı kurma biçimleri, nöronların giriş sinyali ile çıkış sinyali arasındaki ilişki ve sinir ağının öğrenme algoritması, yapay sinir ağı modellerinin birbirinden ayrıştığı noktalardır.

4.2 Biyolojik Sinir Ağı

Sinir sistemi genel olarak, merkezi ve çevresel sinir sistemi olarak iki kısma ayrılır. Çevresel sistem, vücudun her yanından alınan duyu (tat, dokunma, görme, işitme, vücudun pozisyonu, ağrı, ısı, titreşim vb) bilgilerini merkeze taşıyan ve merkezden çıkan emirleri kas veya salgı bezi gibi ilgili yerlere götüren sinir kablolarından oluşur. Yani çevresel sinir sistemi, o kadar basit değilse de, bir veri taşıyıcısı olarak düşünülebilir [8].

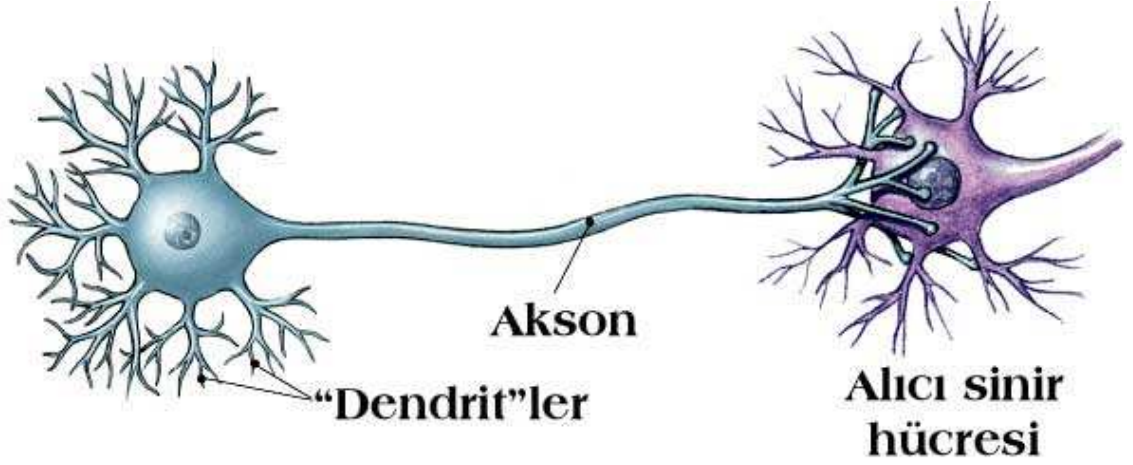
Biyolojik sinir sistemi, merkezinde sürekli olarak bilgiyi alan, yorumlayan ve bir karar üreten beyin (merkezi sinir ağı) bulunduğu üç katmanlı bir sistem olarak açıklanır. Alıcı sinirler (reseptör), organizma içerisinden ya da dış ortamlardan algıladıkları uyarıları, beyne bilgi ileten elektriksel sinyallere dönüştürür. Tepki sinirleri (efektör) ise, beynin ürettiği elektriksel darbelerin organizmada uygun tepkilere dönüşmesine aracılık eder [9]. Şekil 4.1' de bir sinir sisteminin blok şeması verilmiştir.



Şekil 4.1 : Biyolojik sinir sisteminin blok şeması [9].

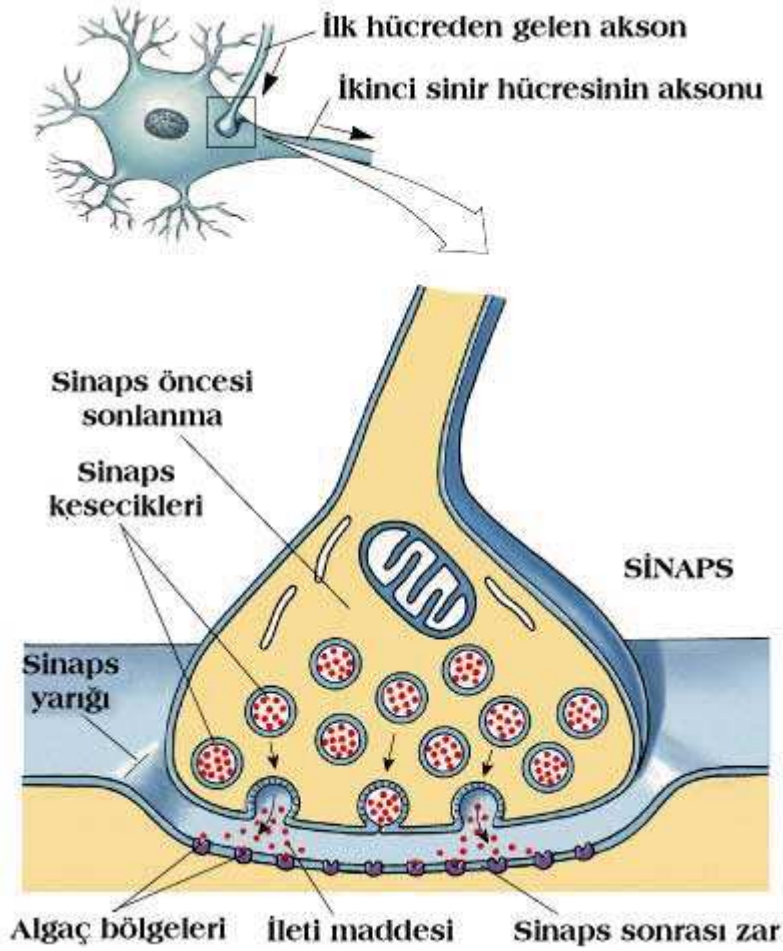
Sinir sisteminin ana yapısını oluşturan hücreler, nöron denen özel hücrelerdir. Bu hücreler, istisnaları olmak üzere, bir gövde, ağaç gibi yan dallar (dendritler) ve bir de, bazen dallanabilen ve hücrenin “kararlarını” diğerlerine ileten, tek bir uzantıdan (akson) oluşurlar. Nöronlar, görevleri ve buldukları 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[8].

Sinir hücreleri aynı zamanda birbirleri ile sıkı ilişki halindedirler. Bu sıkı ilişki, sinirsel işlevin temelini oluşturan bilgi akışını sağlar. Hücreler arası bu bilgi geçiş noktalarına sinaps denir. Sinapslar, değişik tip ve özelliklerdedirler, ancak hemen hepsi bilginin iletimi işlevine sahiptir. Kısacası, nöronlar kendi aralarında bağlantılar kurarak, elektrik devrelerine benzer yollarla iletişim sağlayıp, beyin işlevlerinin ortaya çıkmasını sağlayan ana elemanlardır. Ancak, bu elektriksel devre sistemi, bir sinir bilimcinin hayal edebileceği karmaşıklığın çok ötesinde bir karmaşıklığa sahiptir [8]. Şekil 4.2 'de nöronların çok basitleştirilmiş bir şeması verilmiştir.



Şekil 4.2 : Sinir hücrelerinin basitleştirilmiş şematik yapısı [8].

Kimyasal bir sinapsta, sinyalin bir hücreden diğerine geçişi, nörotransmitter olarak adlandırılan ileti maddeleri aracılığıyla olur. Bu ileti maddeleri, iletinin geldiği kaynak (sinaps öncesi) hücrenin aksonunun ucundan salgılanır. Bu salgılanma, elektriksel uyarının aksonun ucuna gelmesi sayesinde olur. Salgılanan bu ileti maddeleri, sinapsı oluşturan o iki hücre arasındaki ince aralığa salgılanmaktadır. Bu salgılanmayı takiben, çok hızlı bir şekilde, bu ileti maddeleri, karşıdaki hedef (sinaps sonrası) hücrenin zarı üzerindeki uygun algaç (reseptör) moleküllerine bağlanırlar. İşte bu bağlanma, sebep olduğu çeşitli kimyasal olaylar sonucu, yeni hücrede bir elektriksel sinyalin doğmasına sebep olur. Çeşitli sinapslardan gelen verilerin toplanması veya bir sinapstan ardı ardına birkaç sinyalin yeni hücreye geçirilmesi ise, yüksek bir elektriksel potansiyel doğurur. Bu potansiyel, aksiyon potansiyeli adını alır ve işte bu potansiyel, diğer hücelere aktarılmak üzere, akson vasıtasıyla gönderilen elektriksel sinyalin kendisidir [8]. Nöronlar arası iletişimi sağlayan sinapsın şeması şekil 4.3'te görülebilir.



Şekil 4.3 : Genel bir kimyasal sinapsın şematik görüntüsü [8].

Sinyal geçişi, sadece sinir hücreleri arasında değil, kasılma emrini kas hücrelerine taşıyan sinir uçlarıyla kas hücreleri arasında ve bezlere salgı emrini veren uçlarla salgı bezi hücreleri arasında da mevcuttur. Küçük farklarla beraber, mekanizma benzerdir[8].

4.3 Yapay Sinir Ağlarının Kısa Tarihçesi

1942 yılı YSA'nın gelişiminin başlangıç yılı olarak kabul edilmektedir. Bu tarihte, nörobiyolojist olan Warren McCulloch ve istatistikçi olan Walter Pitts, ilk hücre modelini geliştirmişlerdir.

Hebb, 1949 yılında nöron bağlantılarını ayarlamak için kendi adıyla da anılan ilk öğrenme kuralını önerdi. Bu kural, öğrenen uyarlamalı sistemlerin geliştirilmesine de esin kaynağı olmuştur.

1958'de Rosenblatt, perseptron modelini ve öğrenme kuralını geliştirerek, bugün kullanılan kuralların temelini attı.

1960-1962 yılında, Widrow ve Hoff tarafından ADALINE ve LMS kuralı geliştirildi.

1969 yılında, Minsky ve Papert, perseptronların yapısını analiz ettiler ve perseptronun karmaşık lojik fonksiyonlar için kullanılamayacağını ispatladılar. Örneğin, tek katmanlı bir YSA'nın eğitilerek XOR lojik kapsını gerçekleyemediğini kanıtladılar. Perseptronun sadece lineer olarak ayrıştırılabilir problemlerin çözümünde kullanılabileceğini gösterdiler. Bunun üzerine, yapay sinir ağları üzerine çalışan birçok araştırmacı konuyla ilgilenmekten vazgeçti ve yapılan çalışmalar hemen hemen durma noktasına geldi. 1960 yılının ortalarından, 1980 yılının başına kadar bir durgunluk dönemi yaşandı. Çalışmaları büyük ölçüde azaltan bu durgunluğun bir diğer önemli sebebi, YSA'nın bilgi işlemede alternatifi olan günümüzün sayısal bilgisayarlarının, o dönemde büyük gelişmeler kaydetmesidir. Seri işlem yapan ve hızlı birimlerden oluşmuş sayısal bilgisayarlar, aritmetik işlemlerde yüksek hız ve güvenilirlik sağlamışlardır. Ancak tüm bunların yanında bazı bilim adamları (Grossberg, Amari, Fukushima, Kohonen, Taylor, vd.) çalışmalarına devam ettiler.

1974 yılında, Werbos geriye yayılım algoritmasını ortaya attı.

1982 yılında Hopfield YSA'nın birçok problemi çözebilecek kabiliyeti olduğunu gösterdi. Optimizasyon gibi teknik problemleri çözmek için doğrusal olmayan Hopfield ağını geliştirdi. Hopfield ağının, ünlü bir problem olan "gezgin satıcı" probleminin çözümünde başarıyla kullanılabileceği gösterildi.

1982-1984 yılında Kohonen öz düzenlemeli harita (self-organizing map) yı tanımladı. Kendi adıyla anılan eğiticiisiz öğrenen bir ağ geliştirdi.

1986 yılında Rumelhart çok katmanlı perseptronlarda geriye yayılım algoritmasını tekrar ortaya çıkardı. Geri yayılım algoritmasıyla en az bir tane gizli katmanı olan YSA'nın herhangi bir sürekli fonksiyonu düzgün bir şekilde yaklaştırabileceği, Cybenko tarafından gösterildi. Böylelikle perseptronların lineer olmayan yapıları gerçekleyebilecekleri ortaya çıktı. YSA'larına ilgi yeniden arttı.

1988 yılında Chua ve Yang hücresele sinir ağlarını geliştirdiler [10, 11].

4.4 Yapay Sinir Ağlarının Uygulama Alanları

YSA'nın en güçlü özelliği karmaşık, doğrusal olmayan sistemlerde gözlemlerden elde edilen bilgiyle çıkarsamalar yapabilmesidir. Bu özelliğiyle, bir sistemi modellemenin ya da elde edilen verilerle bir sistemin davranışını ifade edebilecek bir fonksiyon çıkarsamanın, bir insan için çok zor olacağı durumlarda YSA'lar çok kullanışlıdır.

YSA fikrinin ortaya çıktığı ilk yıllarda yapılan çalışmalar çok teorik kaldı, uygulama alanı henüz yoktu çünkü henüz yeterli teorik alt yapı oluşmamıştı. YSA'nın kısa tarihçesinde de belirtildiği gibi 80'li yıllara gelinene kadar YSA sistemlerinde aşama kaydedilebilmesi için çözülmesi gereken çok önemli problemler vardı. Ancak o tarihlerden itibaren YSA'lar tekrar gündeme geldi ve çeşitli alanlarda birçok konuda uygulama alanı buldu.

Son 20 yılda hem YSA alanında yaşanan gelişmeler, hem de bilgisayarların hızlı bir şekilde gelişmesi YSA'ların gittikçe yaygınlaşmasına yol açmıştır. Artık herhangi bir kişisel bilgisayarda (PC) çok karmaşık olmayan bir YSA rahatlıkla gerçekleştirilebilir. YSA'lar uygulamalarda bilgisayarlarla ya da özel tasarlanmış entegre devrelerle gerçekleştirilebilir. Bugün YSA'larla mühendislik, tıp, finans ve üretim alanlarında birçok uygulama yapıldığı görülebilir.

Mühendislikte, doğrusal olmayan sistem modelleme, süreç kontrolü, robot sistemlerinin kontrolü, elektrikli sürücü sistemlerinin kontrolü, örüntü tanıma (yüz tanıma, ses tanıma, el yazısını tanıma), görüntü ve veri sıkıştırma, gürültü filtreleme vs. gibi birçok uygulama alanı bulmuştur.

Tıpta, kanserli hücrelerin analizi, EEG ve ECG gibi tıbbi sinyallerin analizi, organ nakli zamanlarının optimizasyonu ve hastanelerde giderlerin optimizasyonu vs. gibi uygulamalarda yer bulmaktadır.

Finans alanında, geçmiş veriler kullanılarak borsa analizi yapmak, yatırım için risk analizi yapmak, pazarlama stratejileri belirlemek gibi konularda kullanılmaktadır.

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

Arıza analizi ve tespitinde de YSA'lar başarılı sonuçlar vermiştir. Bir sistemin ya da elemanın doğru çalışma şeklini öğrenen bir YSA yardımıyla bu sistemlerde meydana gelebilecek arızaların tanımlanma olanağı vardır. Bu amaçla YSA; elektrik makinelerinin, uçak motorlarının ya da entegre devrelerin arıza analizinde kullanılmaktadır [12].

4.5 YSA'ların Özellikleri

Sinir sisteminin modellenmesi için yapılan çalışmalar sonucu oluşturulan yapay sinir ağları, biyolojik sinir sisteminin üstünlüklerine de sahiptir. Bu üstünlükleri şu başlıklar altında özetlemek mümkündür: doğrusal olmama, paralellik, gerçekleşme kolaylığı,

yerel bilgi işleme, hata toleransı, öğrenilebilirlik, genelleme, uyarlanabilirlik, donanım ve hız, analiz ve tasarım kolaylığı.

Yapay sinir ağı özellikle doğrusal olmayan sistemlerde öngörüler (kestirimler) açısından istatistik tekniklere göre daha kolaylık sağlayan bir özelliğe sahiptir. Bundan dolayı başta işletmecilik ve finans olmak üzere bir çok değişik alanlarda kullanım imkanı bulur.

4.5.1 Doğrusal olmama

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

4.5.2 Paralellik

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

4.5.3 Gerçeklenme kolaylığı

Yapay sinir ağlarında basit işlemler gerçekleyen türden hücrelerden oluşması ve bağlantıların düzgün olması, ağların gerçekleşmesi ki açısından büyük kolaylık olmasını sağlamaktadır

4.5.4 Yerel bilgi işleme

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

4.5.5 Hata toleransı

Sayısal bir bilgisayarda, herhangi bir işlem elemanını yerinden almak, onu etkisiz bir makineye dönüştürmektedir. Ancak yapay sinir ağlarında bir elemanda meydana gelebilecek hasar çok büyük önem teşkil etmez. Yapay sinir ağlarının paralel çalışması hız avantajı ile birlikte yüksek hata sağlamaktadır. Seri bilgi işlem yapan

bir sistemde herhangi bir birimin hatalı çalışması, hatta bozulmuş olması tüm sistemin hatalı çalışmasına veya bozulmasına sebep olacaktır. Paralel bilgi işleme yapan bir sistemde ise, sistemin ayrı ayrı işlem elemanlarında meydana gelecek olan hatalı çalışma veya hasar, sistemin performansında keskin bir düşüşe yol açmadan, performansın sadece hata birimlerinin bir oranınca düşmesine sebep olur. YSA, çok sayıda hücrenin çeşitli şekillerde bağlanmasından oluştuğundan paralel dağılmış bir yapıya sahiptir ve ağın sahip olduğu bilgi, ağdaki bütün bağlantılar üzerine dağılmış durumdadır. Bu nedenle, eğitilmiş bir YSA'nın bazı bağlantılarının hatta bazı hücrelerinin etkisiz hale gelmesi, ağın doğru bilgi üretmesini önemli ölçüde etkilemez. Bu nedenle, geleneksel yöntemlere göre hatayı tolere etme yetenekleri son derece yüksektir.

4.5.6 Öğrenebilirlik

Alışlagelmiş veri işleme yöntemlerinin çoğu programlama yolu ile hesaplamaya dayanmaktadır. Bu yöntemler ile tam tanımlı olmayan bu problemin çözümü yapılamaz. Bunun yanında, herhangi bir problemin çözümü için probleme yönelik bir algoritmanın geliştirilmesi gerekmektedir. Yapay sinir ağları, problemleri verilen örneklerle çözer. Çözülecek problemler için yapı aynıdır. YSA'nın arzu edilen davranışı gösterebilmesi için amaca uygun olarak ayarlanması gerekir. Bu, hücreler arasında doğru bağlantıların yapılması ve bağlantıların uygun ağırlıklara sahip olması gerektiğini ifade eder. YSA'nın karmaşık yapısı nedeniyle bağlantılar ve ağırlıklar önceden ayarlı olarak verilemez ya da tasarlanamaz. Bu nedenle YSA, istenen davranışı gösterecek şekilde ilgilendiği problemde aldığı eğitim örneklerini kullanarak problemi öğrenmelidir.

4.5.7 Genelleme

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

4.5.8 Uyarlanabilirlik

YSA, ilgilendiği problemdeki değişikliklere göre ağırlıklarını ayarlar. Yani, belirli bir problemi çözmek amacıyla eğitilen YSA, problemdeki değişimlere göre tekrar eğitilebilir, değişimler devamlı ise gerçek zamanda da eğitime devam edilebilir. Bu

özelliđi ile YSA, uyarlamalı örnek tanıma, sinyal işleme, sistem tanılama ve denetim gibi alanlarda etkin olarak kullanılır.

4.5.9 Donanım ve hız

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

4.5.10 Analiz ve tasarım kolaylıđı

YSA'nın temel işlem elemanı olan hücrenin yapısı ve modeli, bölüm 4.2'de açıklandığı gibi bütün YSA yapılarında yaklaşık aynıdır. Dolayısıyla, YSA'nın farklı uygulama alanlarındaki yapıları da standart yapıdaki bu hücrelerden oluşacaktır. Bu nedenle, farklı uygulama alanlarında kullanılan YSA'ları benzer öğrenme algoritmalarını ve teorilerini paylaşabilirler. Bu özellik, problemlerin YSA ile çözümünde önemli bir kolaylık getirecektir [9, 13].

4.6 Yapay Nöron Modeli

Yapay sinir ağlarında temel unsur, işlem elemanı olarak da adlandırılan nörondur (yapay sinir hücresi). Nöron, YSA'nın çalışmasına temel oluşturan en küçük ve en temel bilgi işleme birimidir. Ağ içinde yer alan tüm nöronlar bir veya birden fazla girdi alırlar ve tek bir çıktı verirler. Bu çıktı, YSA dışına verilen çıktı olabileceđi gibi başka nöronlara girdi olarak da kullanılabilir. Ortaya atılan nöron modellerinde bazı farklılıklar olmakla birlikte genel özellikleri ile bir yapay nöron modeli 5 bileşenden oluşmaktadır.

Bunlar;

1. Girdiler
2. Ağırlıklar
3. Birleştirme Fonksiyonu
4. Aktivasyon Fonksiyonu
5. Çıktı

4.6.1 Girdiler

Girdiler, diđer nöronlardan ya da dış ortamlardan hücreye aktarılan bilgilerdir.

4.6.2 Ağırlıklar

Bilgiler, nörona girişler yoluyla hücreye ulaşır ve her girişin ağırlık katsayısı, ilgili girişin nöron üzerindeki etkisini belirler. Ağırlıklar bir nöronda girdi olarak kullanılacak verilerin göreceli kuvvetini yansıtır. Yapay sinir ağı içinde girdilerin nöronlar arasında iletimini sağlayan tüm bağlantıların farklı ağırlık değerleri bulunmaktadır. Böylelikle ağırlıklar her işlem elemanının her girdisi üzerinde etki yapmaktadır.

4.6.3 Birleştirme fonksiyonu

Birleştirme fonksiyonu, bir hücreye gelen net girdiyi hesaplayan bir fonksiyondur ve birleştirme fonksiyonunun işlevi aktivasyon fonksiyonuna girdi oluşturmaktır. Genellikle net girdi, girişlerin ilgili ağırlıkla çarpımlarının toplamıdır. Birleştirme fonksiyonu, ağ yapısına göre maksimum alan, minimum alan ya da çarpım fonksiyonu olabilir.

$$v = \sum x_i w_i + \theta \quad y = F(v) \quad (4.1)$$

w: Nöronun ağırlıklar matrisi

x: Nöronun giriş vektörü

v: Nöronun net girişi

y: Nöronun çıkışı

4.6.4 Aktivasyon fonksiyonu

Transfer fonksiyonu olarak da geçen aktivasyon fonksiyonu, birleştirme fonksiyonundan elde edilen sonucu işlemde geçirerek nöron çıktısını belirleyen bir fonksiyondur. Ağ modellerinde, nöronun gerçekleştireceği işleve göre çeşitli aktivasyon fonksiyonları kullanılabilir. En uygun aktivasyon fonksiyonu tasarımcının denemeleri sonucunda belli olur. Aktivasyon fonksiyonunun seçimi çoğunlukla uygulamaya ve ağın öğreneceği yapıya bağlıdır. Örneğin eğer ağın bir modelin ortalama davranışını öğrenmesi isteniyorsa sigmoid fonksiyon, ortalamadan sapmanın öğrenilmesi isteniyorsa hiperbolik tanjant fonksiyon kullanılması önerilmektedir [9].

Aktivasyon fonksiyonları bir YSA'da nöronun çıkış genliğini, istenilen değerler arasında sınırlar. Bu değerler genellikle [0,1] veya [-1,1] arasındadır. YSA'nda kullanılacak aktivasyon fonksiyonlarının türevi alınabilir ve sürekli olması, çoğu YSA eğitme algoritmasının uygulanabilmesi için gereklidir. Lineer olmayan aktivasyon

fonksiyonlarının kullanılması YSA'ların karmaşık ve çok farklı problemlere uygulanmasını sağlamıştır.

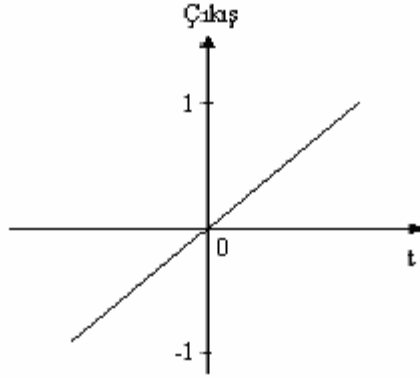
Nöron modellerinde yaygın olarak kullanılan çeşitli aktivasyon fonksiyonları vardır.

4.6.4.1 Doğrusal aktivasyon fonksiyonu

Doğrusal problemleri çözmek amacıyla kullanılan aktivasyon fonksiyonudur. Genellikle çok katmanlı YSA'nın çıkış katmanında kullanılan doğrusal fonksiyon, nöronun net girdisini doğrusal olarak nöron çıkışına verir. Doğrusal aktivasyon fonksiyonu matematiksel olarak,

$$f = A \cdot v \quad (4.2)$$

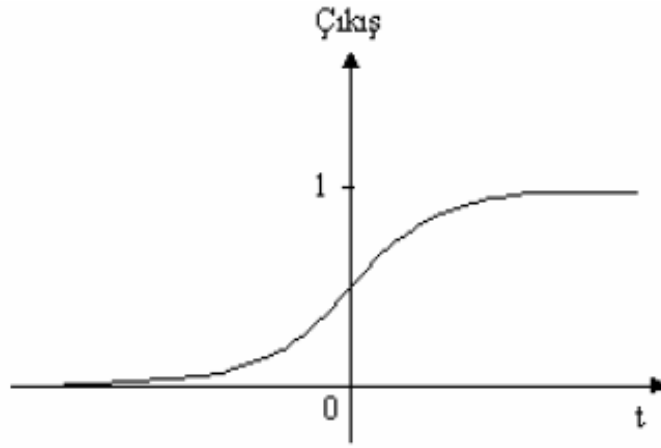
şeklinde tanımlanabilir. "A" sabit bir katsayıdır. YSA ların çıkış katmanında kullanılan doğrusal fonksiyon şekil 4.4'de verilmiştir.



Şekil 4.4 : Doğrusal aktivasyon fonksiyonu.

4.6.4.2 Sigmoid aktivasyon fonksiyonu

Sigmoid aktivasyon fonksiyonu türevi alınabilir, sürekli ve doğrusal olmayan bir fonksiyon olması nedeniyle uygulamada en çok kullanılan aktivasyon fonksiyonudur. Bu özellikleriyle lineer olmayan problemlerin çözümünde, karmaşık ilişkilerin ayrıştırılmasında sıklıkla kullanılır. Şekil 4.5'te gösterilen fonksiyon, her girdi kombinasyonu için sıfır ile bir arasında bir değer üretir.



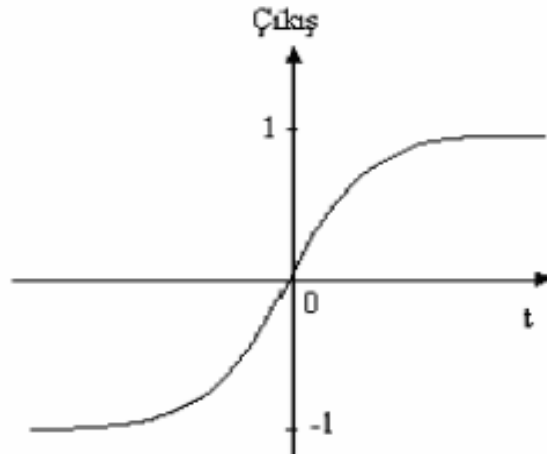
Şekil 4.5 : Sigmoid aktivasyon fonksiyonu.

Sigmoid fonksiyonu denklemleri:

$$f = \frac{1}{1 + e^{-v}} \quad (4.3)$$

4.6.4.3 Tanjant hiperbolik

Tanjant hiperbolik fonksiyonu, sigmoid fonksiyonunun biraz farklı şeklidir. Giriş uzayının genişletilmesinde etkili bir aktivasyon fonksiyonudur. Sigmoid fonksiyonun çıktı aralığı 0 ve 1 olurken, hiperbolik tanjant fonksiyonunun çıktısı -1 ve 1 aralığında oluşmaktadır. Şekil 4.6'da fonksiyonun grafiği görülebilir.



Şekil 4.6 : Tanjant hiperbolik aktivasyon fonksiyonu.

Tanjant fonksiyonunun formülü;

$$f = \frac{1 - e^{-2v}}{1 + e^{2v}} \quad (4.4)$$

Yukarıda anlatılan aktivasyon fonksiyonlarından başka, literatürde geçen diğer aktivasyon fonksiyonları;

- Basamak Fonksiyonu
- Kutuplamalı Basamak Fonksiyonu
- Parçalı Doğrusal Fonksiyon

4.6.5 Çıktı

Nörona gelen giriş sinyallerinin birleştirme fonksiyonu ve aktivasyon fonksiyonundan geçmesinden sonra ortaya çıkan değer, çıktı değeridir. İşlemci elemanın bir den fazla çıkışı olabilir, ancak tek bir çıktı değeri vardır ve bu diğer bir işlemci elemanına girdi olarak girebilir veya ağın çıkışı olabilir.

4.7 Yapay Sinir Ağlarının Sınıflandırılması

YSA'lar, birbirleri ile bağlantılı işlemci birimlerden (nöron) oluşurlar. Her bir nöronun diğer nöronlarla kurduğu bağlantıların yapısı ağın yapısını belirler. YSA ile bir problemi çözebilmek için bağlantıların nasıl değiştirileceği öğrenme algoritması tarafından belirlenir. Kullanılan öğrenme algoritmasına göre, her adımda problemin çözümüne yaklaşmak için ağın ağırlıkları değiştirilir.

YSA'ları daha iyi değerlendirebilmek için YSA'lar yapılarına ve öğrenme algoritmalarına göre sınıflandırılabilir.

4.7.1 YSA'ların yapılarına göre sınıflandırılması

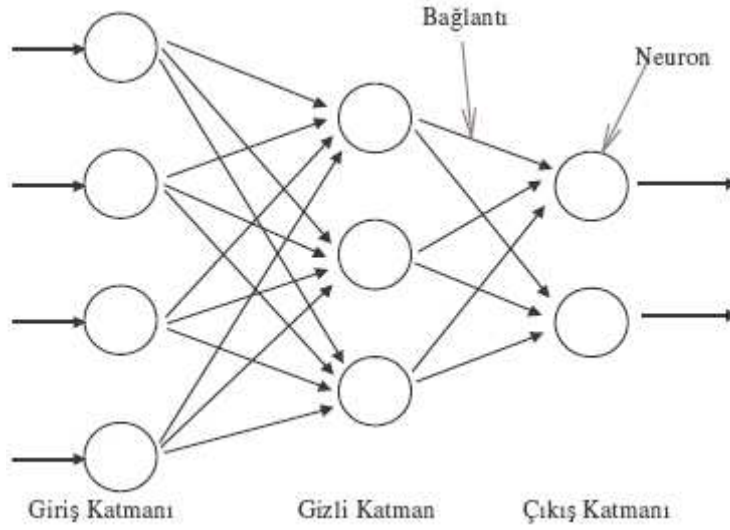
YSA'lar, yapılarına göre, ileri beslemeli (feedforward) ve geri beslemeli (feedback) ağlar olmak üzere iki şekilde sınıflandırılırlar.

4.7.1.1 İleri beslemeli ağlar

İleri beslemeli bir ağda nöronlar genellikle katmanlara ayrılmışlardır. İşaretler, giriş katmanından çıkış katmanına doğru tek yönlü bağlantılarla iletilir. Herhangi bir katmanda yer alan bir nöron, giriş bağlantılarını kendinden önceki katmanla kurar, çıkış bağlantılarını ise kendinden sonraki katmanla kurar ve bulunduğu katmandaki nöronlarla bağlantı kurmaz. İleri beslemeli YSA'da, bir katmandaki nöronların çıkışlarıyla sonraki katmanda yer alan nöronların girişleri birbirine ağırlıklar yoluyla bağlanır. Ağırlıklar bağlantının görece kuvvetini ifade eder.

Giriş katmanı, dış ortamdan aldığı bilgileri değişikliğe uğratmadan orta (gizli) katmandaki hücrelere iletir. Bilgi, orta ve çıkış katmanında işlenerek ağ çıkışı

belirlenir. Bu yapısı ile ileri beslemeli ađlar, doğrusal olmayan statik bir işlevi gerçekleştirir. İleri beslemeli 3 katmanlı YSA'nın, orta katmanında yeterli sayıda nöron olmak kaydıyla, herhangi bir sürekli fonksiyonu istenilen doğrulukta yaklaştılabileceđi gösterilmiştir. Geriye yayılım öğrenme algoritması, bu tip YSA'ların eğitiminde etkin olarak kullanıldığından, bu ađlara geriye yayılım ađları da denmektedir. İleri beslemeli ađlara örnek olarak çok katmanlı perseptron (Multi Layer Perseptron-MLP) ve LVQ (Learning Vector Quantization) ađları verilebilir. Şekil 4.7'de giriş, orta ve çıkış katmanı olmak üzere 3 katmanlı ileri beslemeli YSA yapısı verilmiştir.

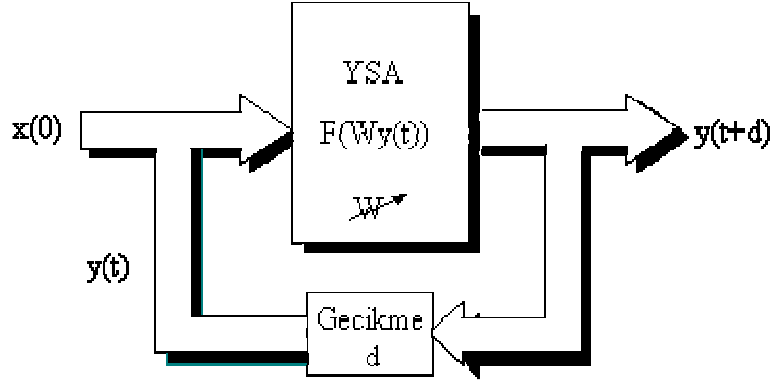


Şekil 4.7 : Bir gizli katmanı olan YSA.

İleri beslemeli YSA'lar, en genel anlamıyla giriş ve çıkış uzayı arasında statik haritalam yapar. Herhangi bir andaki çıkış, sadece o andaki girişin bir fonksiyonudur [9].

4.7.1.2 Geri beslemeli ađlar

Bir geri beslemeli YSA, herhangi bir katmandaki çıkışların, giriş katmanına veya önceki ara katmanlara geri beslendiđi bir ađ yapısıdır. Böyle bir yapıyla, girişler hem ileri yönde hem de geri yönde aktarılmış olur. Şekil 4.8'de bir geri beslemeli ađ görülmektedir. Bu çeşit sinir ađlarının dinamik hafızaları vardır ve bir andaki çıkış hem o andaki hem de önceki girişleri yansıtır. Bundan dolayı, özellikle önceden tahmin uygulamaları için uygundur. Geri beslemeli ađlar çeşitli tipteki zaman serilerinin tahmininde oldukça başarı sağlamışlardır. Bu ađlara örnek olarak Hopfield, SOM (Self Organizing Map), Elman ve Jordan ađları verilebilir.



Şekil 4.8 : Geri beslemeli ağın blok şeması [9].

Geribeslemeli YSA'da, en az bir nöronun çıkışı kendisine ya da önceki katmandaki diğer nöronlara giriş olarak verilir ve genellikle geribesleme bir geciktirme elemanı üzerinden yapılır. Geribesleme, bir katmandaki nöronlar arasında olduğu gibi katmanlar arasındaki nöronlar arasında da olabilir. Bu yapı ile geribeslemeli YSA, doğrusal olmayan dinamik bir davranış gösterir. Dolayısıyla, geribeslemenin yapılış şekline göre farklı yapıda ve davranışta geribeslemeli YSA yapıları elde etmek mümkündür.

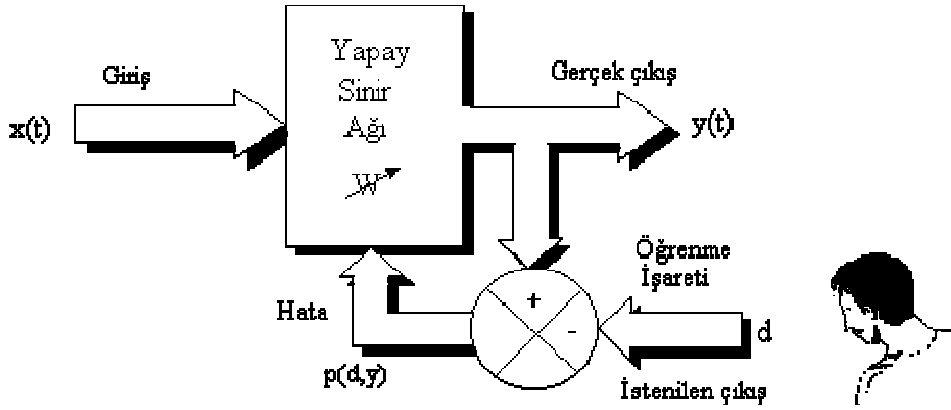
4.7.2 YSA'ların öğrenme algoritmalarına göre sınıflandırılması

Öğrenme edimi, biyolojik varlıkların gözlem ve eğitim yoluyla davranışlarını yeni şartlara uyarlayabilme yeteneği olarak tanımlanır. YSA'ya da biyolojik bir yapı gibi öğrenme yeteneği kazandırmak için, eğitim ve gözleme dayanan bir takım kurallar ve metotlar yoluyla ağıdaki ağırlıkların yenilenmesi sağlanmalıdır. Bunun için genel olarak iki öğrenme metodundan ve bunların uygulandığı değişik öğrenme kurallarından söz edilebilir. Bu öğrenme kuralları aşağıda açıklanmaktadır.

4.7.2.1 Danışmanlı öğrenme

Bu tip öğrenmede, elde YSA'nın kullanıldığı probleme ilişkin giriş-çıkış örnek kümeleri vardır. Danışmanlı öğrenme işlemi, elde bulunan örnek giriş-çıkış kümesinin YSA'ya öğretilmesi işlemidir. Bu öğrenmede, ağı ürettiği çıktılar ile örnek kümesindeki gerçek çıktılar arasındaki fark hata olarak ele alınır ve bu hata minimize edilmeye çalışılır. Nöronlar arasındaki bağlantı ağırlıkları adım adım yenilenecek, gerçek çıktılarla YSA'nın çıktıları arasındaki fark minimize edilmeye çalışılır. Şekil 4.9'da danışmanlı öğrenme yapısı gösterilmiştir. Widrow-Hoff tarafından geliştirilen delta kuralı ve Rumelhart ve McClelland tarafından geliştirilen genelleştirilmiş delta kuralı veya geriye yayılma (back propagation) algoritması danışmanlı öğrenme algoritmalarına örnek olarak verilebilir. Danışmanlı öğrenme,

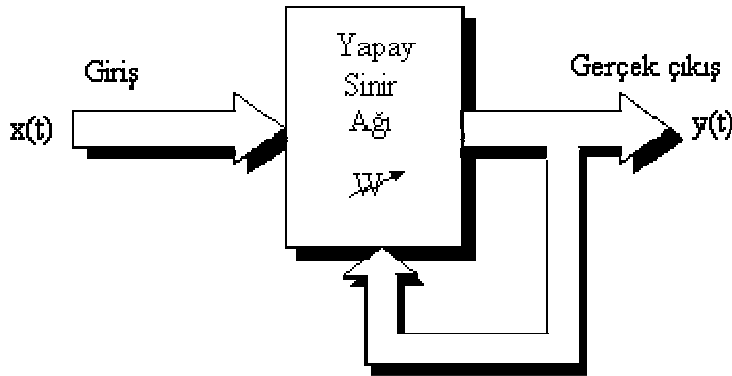
on-line veya off-line olarak uygulanabilir. Off-line öğrenmede, belirlenen bir YSA, kullanılacağı ortamın dışında, deneylerle oluşturulan giriş-çıkış örnek kümesiyle eğitilir. Eğitim tamamlandıktan sonra elde edilen YSA kullanılacağı ortama aktarılır ve kullanım esnasında ağırlıklar yenilenmez, öğrenme olmaz. Öğrenme sadece kullanılacağı ortamın dışında gerçekleşir ve orada biter. On-line öğrenmede ise öğrenme, ağı kullanıldığı ortamda gerçekleşir, eğitim için ayrı bir işlem ortamı kullanılmaz. Başka bir deyişle, on-line öğrenme gerçek zamanda (real time) gerçekleşir [10].



Şekil 4.9 : Danışmanlı öğrenme yapısı [9].

4.7.2.2 Danışmansız öğrenme

Danışmansız öğrenmede ağı sadece girdiler verilir. Ağın ulaşması gereken hedef (gerçek) çıktılar verilmaz. YSA, girişe verilen örnekten çıkış bilgisini ortaya çıkaran sınıflandırma kurallarını kendi kendine geliştirir. Ağ daha sonra bağlantı ağırlıklarını aynı özellikleri gösteren örüntüler oluşturmak üzere ayarlar. Şekil 4.10'da danışmansız öğrenme yapısı gösterilmiştir. Grossberg tarafından geliştirilen ART (Adaptive Resonance Theory) veya Kohonen tarafından geliştirilen SOM (Self Organizing Map) öğrenme kuralı danışmansız öğrenmeye örnek olarak verilebilir.



Şekil 4.10 : Danışmansız öğrenme yapısı [9].

4.7.3 Çeşitli YSA yapıları ve öğrenme algoritmaları

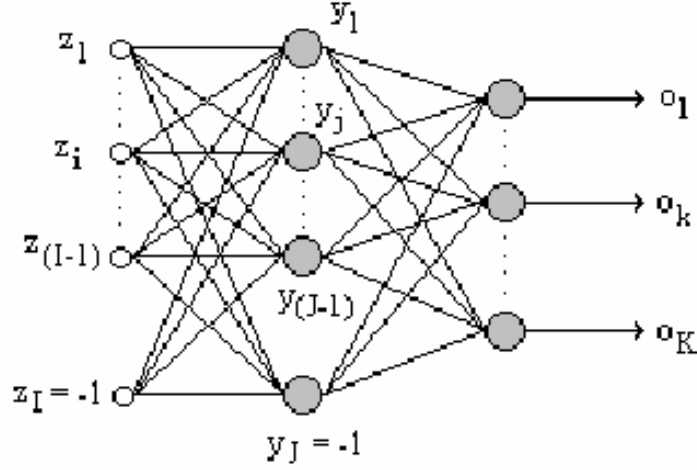
4.7.3.1 Çok katmanlı perseptronlar

Minsky, basit perseptron modelinin lineer olmayan problemlere çözüm getirmedeğini göstermiş, örneğin XOR'un basit perseptron ağı ile gerçekleştirilemeyeceğini kanıtlamıştır. Bunun sonucunda, YSA'ların doğrusal olmayan problemlerin çözümünde kullanılamayacağını konjektür olarak ortaya atmıştır. Bu durum, YSA ile ilgili yapılan bilimsel araştırmaların pek çoğunun durmasına neden olmuştur, çünkü günlük hayatta karşılaşılan problemlerin hemen hepsi doğrusal olmayan nitelikler taşımaktadır. Bazı araştırmacılar çalışmalarına devam ederek YSA'nın doğrusal olmayan problemlerin de çözümünde kullanılabileceğini göstermeleri ile YSA ile ilgili çalışmalar tekrar hız kazanmıştır.

XOR gibi doğrusal olmayan problemleri çözmek için yapılan çalışmalar sonucunda çok katmanlı perseptron modeli geliştirilmiştir. Rumelhart ve arkadaşları tarafından geliştirilen bu modele hata yayma algoritması veya geriye yayılım algoritması (backpropagation network) de denilmektedir. Çok katmanlı ağ (ÇKA) modeli yapay sinir ağlarına olan ilgiyi çok hızlı bir şekilde arttırmış ve YSA tarihinde yeni bir dönem başlatmıştır. Bir çok öğrenme algoritmasının bu ağı eğitmede kullanılabilir olması, bu modelin yaygın kullanılmasının sebebidir.

Bir ÇKA modeli, şekil 4.11'deki gibi bir giriş katmanı, bir veya daha fazla ara katman ve bir de çıkış katmanından oluşur. Bir katmandaki bütün işlem elemanları bir üst katmandaki bütün işlem elemanlarına bağlıdır. ÇKA ileri beslemeli bir ağ modelidir. Giriş katmanında herhangi bir bilgi işleme yapılmaz. Buradaki işlem elemanı sayısı tamamen uygulanan problemlerin giriş sayısına bağlıdır. Ara katman sayısı ve ara katmanlardaki işlem elemanı sayısı ise, deneme yanılma yolu ile bulunur. Çıkış katmanındaki eleman sayısı ise probleme dayanılarak belirlenir. Delta öğrenme kuralını kullanan bu ağ modeli, özellikle sınıflandırma, tanıma ve genelleme yapmayı gerektiren problemler için çok önemli bir çözüm aracıdır.

ÇKA modelinin temel amacı, ağın beklenen çıktısı ile ürettiği çıktı arasındaki hatayı en aza indirmektir. Çok katmanlı perseptronlar, danışmanlı öğrenme yöntemiyle eğitilirler. Ağın görevi her girdiye karşılık gelen çıktıyı en az hatayla üretmektir. Örnekler giriş katmanına uygulanır, ara katmanlarda işlenir ve çıkış katmanından da çıkışlar elde edilir. Kullanılan eğitim algoritmasına göre, ağın çıkışı ile gerçek çıkış arasındaki hata tekrar geriye doğru yayılarak, hata belirlenen bir değer altına düşüncüye kadar ağın ağırlıkları değiştirilir.



Şekil 4.11 : Çok katmanlı perseptron modeli.

4.7.3.2 Radyal tabanlı sinir ağı

Çok katmanlı YSA'nın tasarımında eğitici geriye yayılım öğrenme algoritması bir en iyileme uygulamasıdır. Ara katmandaki nöronlar, girişlerin ağırlıklandırılmış şeklini kullanmamakta ve ara katmandaki işlemci elemanların çıkışları, YSA girişleri ile temel fonksiyonun merkezi arasındaki uzaklığa göre belirlenmektedir. Radyal tabanlı fonksiyon ağı (RTFA) tasarımı ise çok boyutlu uzayda eğri uydurma yaklaşımıdır ve bu nedenle RTFA'nın eğitimi, çok boyutlu uzayda eğitim verilerine en uygun bir yüzeyi bulma problemine dönüşür. RTFA'nın genellemesi ise test verilerini interpolate etmek amacıyla eğitim sırasında bulunan çok boyutlu yüzeyin kullanılmasına eşdeğerdir. Radyal tabanlı fonksiyonlar, sayısal analizde çok değişkenli interpolasyon problemlerinin çözümünde kullanılmış ve YSA'nın gelişmesi ile birlikte bu fonksiyonlardan YSA tasarımında yararlanılmıştır. RTFA, ileri beslemeli YSA yapılarına benzer şekilde giriş, orta ve çıkış katmanından oluşur; ancak giriş katmanından orta katmana dönüşüm, radyal tabanlı aktivasyon fonksiyonları ile doğrusal olmayan sabit bir dönüşümdür. Orta katmandan çıkış katmanına ise uyarlamalı ve doğrusal bir dönüşüm gerçekleştirilir.

Burada, ara katman işlemci elemanları, lineer yapıdaki bir çıkış katmanına bütünüyle bağlantılıdır.

$$I_k = \|X - c_k\| = \sqrt{\sum_{i=1}^N (X_i - c_{ki})^2} \quad (4.5)$$

ile ifade edilir.

Nöronun çıkış değeri ise,

$$V_k = e^{\frac{I_k}{\sigma_k^2}} \quad (4.6)$$

formülü ile hesaplanır.

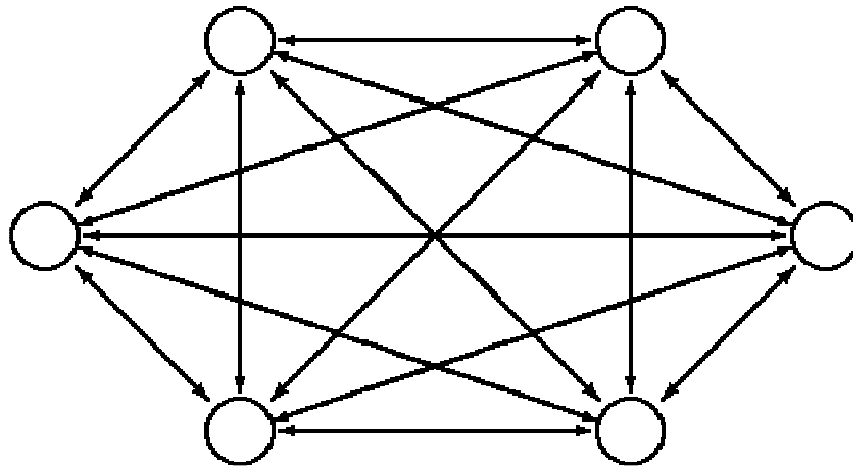
C_k : Ağırlıklar

σ_k : Genişlikler

RTFA'da uyarlanabilecek serbest parametreler; merkez vektörleri, radyal fonksiyonların genişliği ve çıkış katman ağırlıklarıdır. Çıkış katmanı doğrusal olduğundan ağırlıklar, eğim düşme (gradient descent, gradyan azalan) ya da doğrusal en iyileme yöntemleri ile kolayca bulunabilir. Merkezler, girişler arasından rastgele ve sabit olarak seçilebilmekle birlikte RTFA'nın performansını iyileştirmek amacıyla merkez vektörlerinin ve genişliğin uyarlanması için çeşitli yöntemler geliştirilmiştir. Merkez vektörleri, eğim düşme yöntemine göre eğitici öğrenme algoritması ile uyarlanarak, dik en küçük kareler yöntemi ile, ya da kendiliğinden düzenlemeli yöntemle giriş örneklerinden öbekleme yapılarak belirlenebilir.

4.7.3.3 Hopfield ağı

Hopfield sinir ağı, beyine benzer bir şekilde belirli hafızaları veya örüntüleri depolayan basit bir yapay sinir ağıdır [10]. Hopfield ağ topolojisi diğer ağlardan farklıdır. Farklı katmanlar yoktur; her birim diğer tüm birimlere bağlıdır. Ayrıca, bağlantılar çift yönlüdür (bilgi her iki yönde akar) ve simetriklerdir. Bu ağ genellikle ikili (0 veya 1) ve bipolar (+1 veya -1) girişler kabul eder. Tek katman işlemci elemanları vardır ve her işlemci eleman bir diğerine bağlanmıştır. Şekil 4.12'de bir Hopfield ağının yapısı gösterilmiştir.



Şekil 4.12 : Hopfield ağı.

Hopfield ağı, bir geriyayılımlı (BackPropagation) ağın eğitildiği gibi eğitilmez. Bunun yerine, örnek örüntü grupları seçilir, ağın ağırlıklarının başlangıç değerlerini saptamak için kullanılır. Bu bir kere yapıldıktan sonra herhangi bir örüntü ağı sunulur ve bu da giriş örüntüsüne en çok benzeyen örnek örüntülerden biriyle sonuçlandırılır. Çıkış örüntüsü, birimlerin durumlarına bakılarak ağdan okunabilir.

Hopfield Ağı, bir nöronun matematiksel basitleştirilmiş halinin nasıl büyük sinir ağlarının davranış analizlerine izin verdiğini açıklamaktadır. Bütün bir sinirsel mimarinin bir özelliği üzerindeki birimlerin aktivasyonlarındaki değişikliklerin matematiksel olarak etkisini karakterize etmeye enerji denir, Hopfield (1982) lokal etkileşimler ile global davranışlar arasında önemli bağlantıyı sağlamaktadır.

Ağ, N birime sahip olsun ve bunlar 1 den N'e numaralanmış olsun. i. birimden j. birime olan bağlantının ağırlığı $1 \leq i, j \leq N$ için w_{ij} 'dir. Tüm $i=1, \dots, N$ için hiçbir birimin kendine olan bağlantısı yoktur, $w_{ii} = 0$. i. birimden j. birime olan bağlantının ağırlığı, j. birimden i. Birime olan bağlantının ağırlığına eşit olduğundan, $w_{ij} = w_{ji}$.

Her birimin onunla bütünleşmiş bir eşik değeri vardır. Eşik, diğer birimlerden olan girişlere cevabında birimin durumundaki değişimi belirler i. birimin eşikini Q_i ile gösteririz.

M tane örnek örüntümüz olsun: p_1, \dots, p_m . Her birinin N elemanı vardır, böylelikle $i=1, \dots, M$ için $p_i = (p_i^1, \dots, p_i^N)$ 'dir. Ağırlıkları $1 \leq i, j \leq N, i \neq j$ için $w_{ij} = \sum_{k=1}^M p_k^i p_k^j$ formülü

ile hesaplanır. t anında i. birimin durumu $\mu_i(t)$, $i=1, \dots, N$ olsun. (x^1, \dots, x^N) 'in giriş örüntüsü olsun. $\mu_i(0) = x^i$, $i=1, \dots, N$ eşitliğini sağlayarak ağa giriş örüntüsü yüklenir. Ağ çalıştırmak için, birbirini izleyen anlarda denklem (4.7) kullanılarak birimlerin durumları hesaplanır.

$$\mu_i(t+1) = f\left(\sum_{j=1, j \neq i}^M w_{ij} \mu_j(t) - \theta_i\right) \quad (4.7)$$

$i=1, \dots, N$ ve $t=1, 2, \dots$

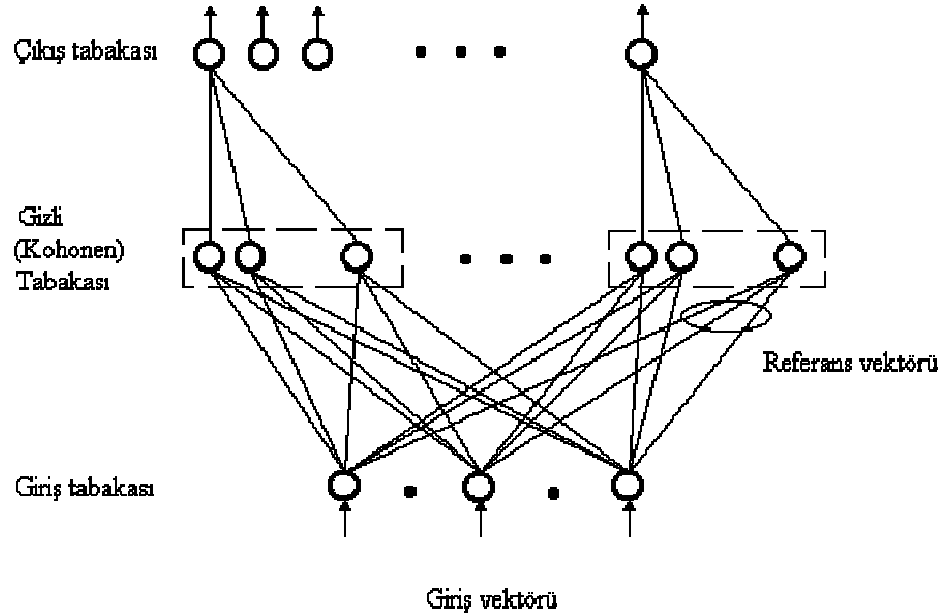
birimlerin ikili çıkışlara sahip olduğu durumlarda, f basamak fonksiyonudur: eğer $x < 0$ ise $f(x) = -1$ ve eğer $x > 0$ ise $f(x) = 1$.

Eğer, $\left(\sum_{j=1, j \neq i}^N w_{ij} \mu_j(t) - \theta_i\right) = 0$ ise birimin durumu değişmez. Sürekli durumda f, [a,b]

sınırlı bir sigmoid fonksiyondur [9].

4.7.3.4 Vektör kuantalama (learning vector quantization ,LVQ)

Şekil 4.13, üç katmandan oluşan LVQ ağını göstermektedir. Bu katmanlar, giriş katmanı, gizli katman ve çıkış katmanıdır. LVQ ağı, giriş katmanı ile gizli katman birbirleriyle tamamen, gizli katman ile çıkış katmanı da kısmen bağlıdır. Her çıkış nöron düğümü farklı bir gizli nöron düğümü kümesine bağlıdır. Gizli ve çıkış nöronları arasındaki ağırlıklar "1"e sabitlenmiştir. Girişler ile gizli katmandaki nöronların bağlantılarının ağırlıkları "referans" vektörlerinin elemanlarını oluşturur. Ağın öğretilmesi aşamasında bunlar yeniden değerler alırlar. Hem gizli katman nöronları hem de çıkış katmanı nöronları ikili (binary) çıkışa sahiptir. Ağa bir giriş örüntüsü verildiğinde referans vektörü giriş örüntüsüne en yakın olan gizli nöron kümesi "1", diğerleri "0" üretir. "1" üreten çıkış nöronu giriş isaretini sınıflar ve her nöron ayrı bir sınıfa atanmıştır [9, 10, 12].



Şekil 4.13 : LVQ ağı [9].

En basit LVQ öğrenme prosedürü şu sıra ile açıklanabilir:

1. Referans vektörlerinin ağırlıklarının başlangıç değerlerinin belirlenmesi.
2. Ağa giriş örüntüsünün uygulanması.
3. Giriş örüntüsü ile her referans vektörü arasındaki öklit (Euclidian) uzaklığının hesaplanması.
4. Giriş örüntüsüne en yakın referans vektörünün ağırlıklarının yeniden düzenlenmesi. Eğer bu gizli katman nöronu, o çıkış katmanı nöronunun bağlı olduğu gizli katman nöron kümesine bağlı ise referans vektörünü giriş

örüntüsüne daha yakın hale getirir. Aksi takdirde referans vektörünü uzaklaştırır.

5. 2 numaralı adıma yeni bir giriş örüntüsü ile dönülür ve işlemlere sınıflandırılması istenen giriş örüntüsü kalmayana dek devam edilir.

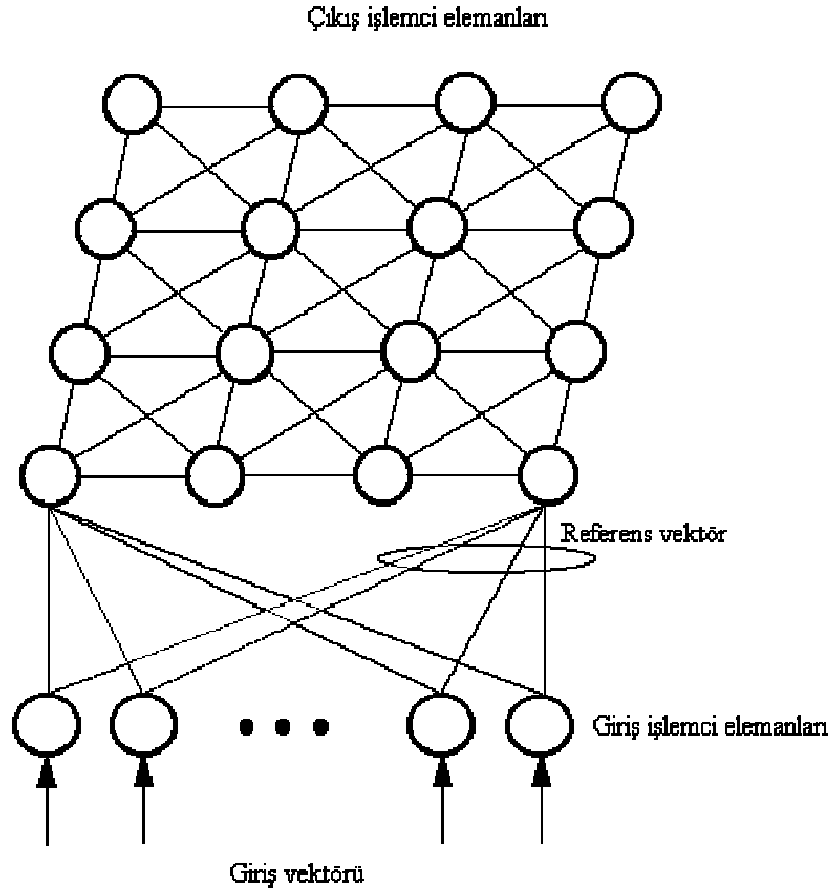
4.7.3.5 Kohonen ağı

Şekil 4.14'de gösterildiği gibi Kohonen ağı, bir giriş katmanı ve bir de çıkış katmanı olmak üzere iki katmandan oluşur. Çıkış katmanındaki işlemci elemanlar (İE) genellikle düzenli iki boyutlu aralıklar olarak düzenlenir. Çıkıştaki her işlemci eleman, bütün giriş işlemci elemanlarına bağlıdır. Bağlantıların ağırlıkları verilen çıkış işlemci elemanı ile ilgili olan referans vektörünün elemanlarını oluşturur.

Kohonen ağının öğrenme adımları aşağıdaki gibidir.

1. Çıkış İE'lerin bütün referans vektörlerini küçük rastgele değerlere çekilir,
2. Bir giriş örüntüsü alınır,
3. Kazanan çıkış işlemci elemanı belirlenir. Bu yapılırken, Kohonen katmanındaki bütün proses elemanlarının NET girdileri hesaplandıktan sonra bunların en büyüğüne sahip olan proses elemanı yarışmayı kazanan eleman olarak belirlenmiş olur. Bu elemanın çıktısı 1, diğerlerinininki 0 olarak atanır. Kazanan işlemci giriş örüntüsüne en yakın referans vektörüne sahip işlemci elemandır. Referans vektörü ile giriş vektörü arasındaki öklit uzaklığı genellikle uzaklık ölçüsü olarak alınır.
4. Kazanan işlemci elemanının ve onun komşularının referans vektörü güncelleştirilir. Bu referans vektörleri giriş vektörüne yaklaştırılır. Bu yaklaştırma (ayarlama) kazanan işlemci elemanı için en fazla olur ve bu işlemci elemanından uzaklaştıkça azalır. Öğrenme ilerledikçe komşuların sayısı azalmakta ve öğrenme sonunda sadece kazanan işlemci elemanın referans vektörü ayarlanmaktadır.

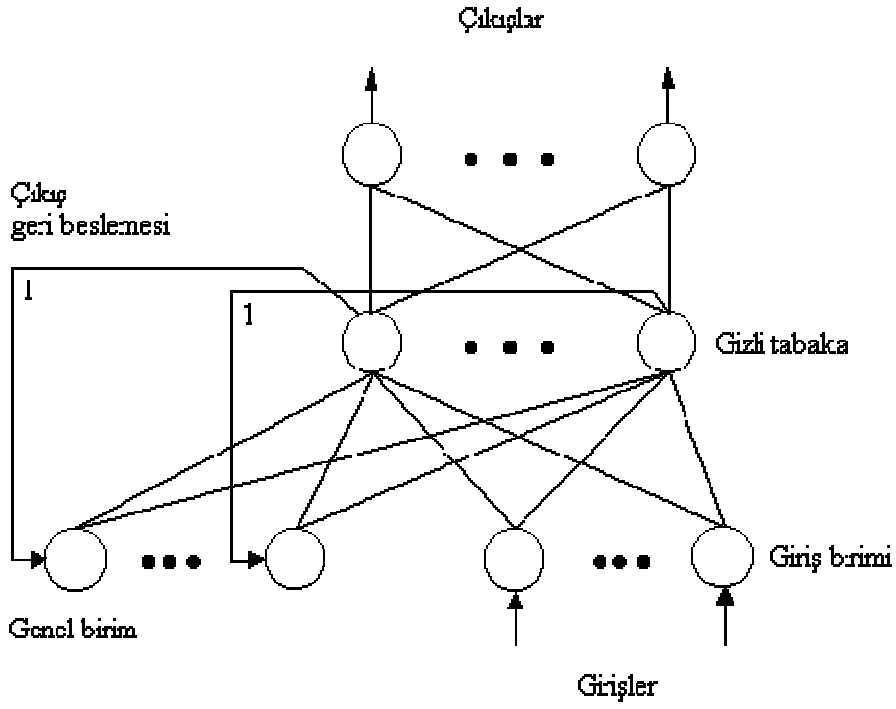
İyi öğrenmiş bir Kohonen ağında birbirine yakın çıkış işlemci elemanlarının referans vektörleri vardır. Öğrenmeden sonra bir etiketleme işlemine bağlanır. Bilinen sınıfların giriş örüntüleri ağa giriş olarak verilir ve bu giriş örüntüleri tarafından aktif hale gelen çıkış işlemci elemanlarına bu sınıf etiketleri verilir. LVQ ağında ise bir çıkış işlemci elemanı eğer yarışmayı diğer çıkış işlemci elemanlarına karşı kazanırsa bir giriş örüntüsü tarafından aktif hale getirildiği hatırlatılır.



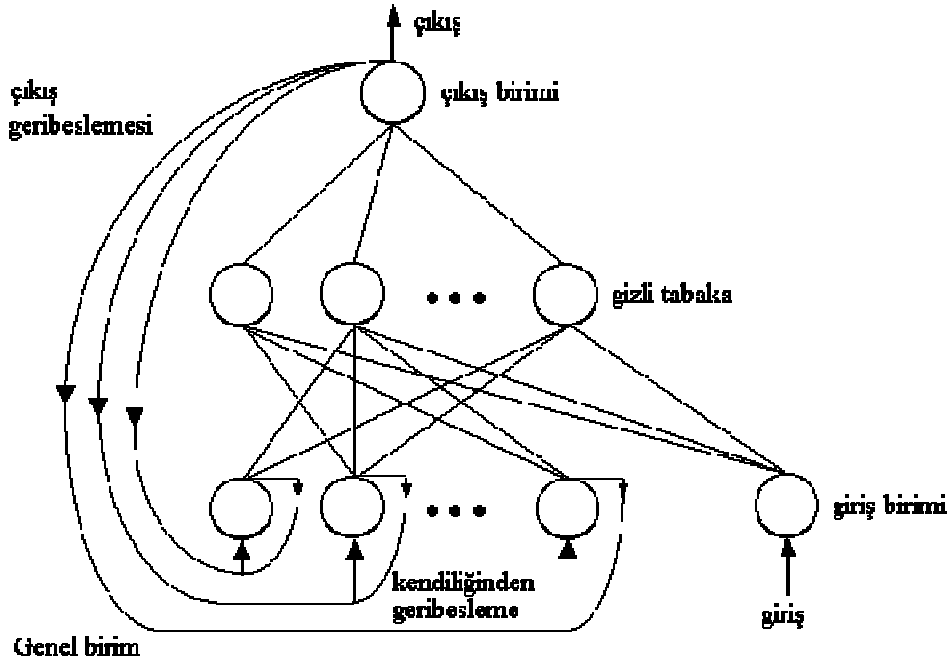
Şekil 4.14 : Kohonen ağı [9].

4.7.3.6 Elman ve Jordan ağları

Şekil 4.15a ve şekil 4.15b Elman ve Jordan ağlarını göstermektedir. Bu ağlar, MLP'ye benzer bir yapıdadırlar. Her iki ağda da gizli katmana ek olarak bir "durum" katmanı denilen özel bir gizli katman daha bulunmaktadır. Bu katman gizli katmandan veya çıkış katmanından geri besleme işaretleri alır. Jordan ağının aynı zamanda durum katmanındaki her nöronun kendisine bağlantıları vardır. Her iki ağda da durum katmanındaki nöronların çıkışları ileriye doğru gizli katmana verilmektedir. Eğer sadece ileri doğru bağlantılar göz önüne alınır ve geri besleme bağlantılarına sabit değerler verilirse, bu ağlar sıradan ileri beslemeli ağlar haline gelir.



(a)



(b)

Şekil 4.15 : a) Elman ağı ve b) Jordan ağı [9].

4.8 Geri Yayılım Algoritması

Bir çok uygulamada kullanılan en yaygın öğretim algoritması geri yayılım algoritmasıdır. Anlaşılması kolay ve matematiksel olarak ispatlanabilir olmasından dolayı en çok tercih edilen öğretim algoritmasıdır. Bu algoritma, hataları geriye doğru çıkıştan girişe azaltmaya çalışmasından dolayı geri yayılım ismini almıştır.

Geriye yayılım algoritması, sadece ileri beslemeli ağlar için kullanılabilir. Klasik çok katmanlı geri yayılım ağı, daima; bir giriş katmanı, bir çıkış katmanı ve en az bir gizli katmana sahiptir. Gizli katmanların sayısında teorik olarak bir sınırlama yoktur, fakat genel olarak bir veya iki tane bulunur. Genelde, gizli katman sayısının artması ağın eğitiminin çok yavaş olmasına neden olabilir.

Geriye yayılım algoritmasının eğitimi sırasında, giriş örüntüleri gizli katmanlardan geçirilir ve çıkış katmanında sonuç üretilir. Ağın çıkışında elde edilen sonuç ile beklenen sonuç karşılaştırılır ve çıkış katmanının hataları bulunur. Hataların bulunmasından sonra, hataların sinaptik bağlantılara göre gradyanı çıkış katmanından geriye doğru gizli katmana geçirilir. Algoritmanın bu özelliğinden dolayı geriye yayılım algoritması denilmiştir [12].

Her çıkış ve gizli katmandaki nöronların hata değerleri bulunduktan sonra, her nöron kendi hedef çıkışına yaklaşmak için sinaptik ağırlıklarını değiştirir. Ağırlık değiştirme denklemleri, ağın performans fonksiyonunu minimum yapacak şekilde düzenlenir. Standart geriye yayılım algoritması performans fonksiyonu olarak, ortalama hatanın karesi kullanılır. Standart geriye yayılım algoritmasının bazı eksiklerini gidermek için, ağırlık güncelleme denklemlerinde veya performans fonksiyonunda değişiklikler yapılarak, yeni geriye yayılım algoritmaları türetilmiştir.

4.8.1 Delta kuralı

Delta kuralı eğitimci ileri beslemeli YSA'larda sık kullanılan bir yöntemdir. Bu tez çalışmasında da yük kestirimini başarmak için delta kuralından yararlanılmıştır. Bu sebeple delta kuralına ilişkin bazı algoritmalar sonraki bölümlerde açıklanmıştır.

YSA uygulamalarının amacı giriş/çıkış ilişkiler serilerinin öğrenilmesidir. Bunun için, ağ tasarlandıktan sonra temsili veri kümesi ile eğitilir. Geri yayılım ağlar eğitildiğinde, amaç ağırlıklar kümesinin elde edilmesidir. Şöyle ki, genel sistem hatasının (arzu edilen çıkış ile gerçek ağ çıkışı arasında) minimum olması hedeflenir. Matematiksel olarak toplam ağ hatası (hataların kareleri toplamı) şu şekilde ifade edilir.

$$E = \frac{1}{2} \sum_{k=1}^P \sum_{j=1}^K (y_{kj} - O_{kj})^2 \quad (4.8)$$

Burada; E , toplam hata; P , eğitim kümesindeki örüntü sayısı; K , ağ çıkışlarının sayısı; y_{kj} , k. örüntünün hedef çıkışı ve O_{kj} , k. örüntünün j. çıkışıdır. Bu nedenle toplam hata, P hatalarının toplamıdır.

Hatanın minimizasyonu, ağırlıkların negatif gradyan yönünde değiştirilmesi gerektiğinden, Δw_{ij} gradyan vektörü ile orantılıdır. Bu yüzden $\Delta w_{ij} \approx -\nabla E$, burada ∇E gradyan vektörüdür. Geri yayılım ağlarının amacı, ağırlıkları denklem (4.9)'daki bağıntıya göre değiştirmektir.

$$\nabla w_{ij} \approx -\frac{\partial E}{\partial w_{ij}} \quad (4.9)$$

Hata fonksiyonunu minimize etmek için ağırlıklar (4.10)'a göre güncellenmelidir:

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} - \eta \frac{\partial E_k}{\partial w_{ij}} \quad (4.10)$$

$\eta > 0$, öğrenme hızı parametresidir. Öğrenme hızı, sabit bir değer olabileceği gibi, hata yüzeyinin yapısına göre uyarlanabilecek şekilde de olabilir.

$$\frac{\partial E_k}{\partial w_{ij}} = \frac{\partial E_k}{\partial \text{net}_i^k} \frac{\partial \text{net}_i^k}{\partial w_{ij}} = -(y_i^k - o_i^k) x_j^k \quad (4.11)$$

$\text{net}_i^k = w_{i1} x_1^k + \dots + w_{in} x_n^k$ olmak üzere,

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} + \eta (y_i^k - o_i^k) x_j^k \quad j=1, \dots, n \quad (4.12)$$

ağırlık değiştirme fonksiyonu tanımlanmış olur.

i. çıkış nöronu tarafından üretilen hata işareti δ_i^k , delta, şu şekilde tanımlanır:

$$\delta_i^k = -\frac{\partial E_k}{\partial \text{net}_i^k} = (y_i^k - o_i^k) \quad (4.13)$$

Lineer çıkışlar için δ_i^k , hedef çıkış ile ağın çıkışı arasındaki farktan başka birşey değildir. O halde delta öğrenme kuralı şu şekilde yazılabilir:

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} + \eta \delta_i^k x_j^k \quad (4.14)$$

$i=1, \dots, m$ ve $j=1, \dots, n$.

4.8.1.1 Sigmoidal aktivasyon fonksiyonuyla delta kuralı

Pratikte birçok durumda linner aktivasyon fonksiyonları yerine lineer olmayan aktivasyon fonksiyonları kullanılır. Dolayısıyla lineer olmayan aktivasyon fonksiyonlarının türevi, delta kuralıyla ağırlıklar güncellenebilmesi için hesaplanması gerekir. Bölüm 6'da YSA uygulamasında kullanılan ağ modelinde sigmoidal fonksiyon kullanıldığı için bu fonksiyonun geri yayılım algoritmasında kullanımı incelenmiştir.

Unipolar sigmoidal aktivasyon fonksiyonu $f(t) = \frac{1}{1 + \exp(-t)}$ 'nin türevi:

$$f'(t) = f(t)(1 - f(t)) \quad (4.15)$$

eşitliğini sağlar.

Burada sigmoidal fonksiyonu çıkış fonksiyonuna uyarladığımızda:

$$o(\langle w, x \rangle) = \frac{1}{1 + \exp(-w^T x)} \quad (4.16)$$

w ağırlık vektörü, x ise giriş vektörüdür. Ağırlıkları güncelleme, gradyan azalan yöntemi ile hata fonksiyonunu minimum yapmak için şu şekilde kullanılır:

$$w^{(t+1)} = w^{(t)} - \eta E'_k(w) \quad (4.17)$$

Hata fonksiyonu E_k 'nin w noktasında gradyan vektörü şu şekilde hesaplanır:

$$E'_k(w) = \frac{d}{dw} \left(\frac{1}{2} \cdot \left[y^k - \frac{1}{1 + \exp(-w^T x^k)} \right]^2 \right) \quad (4.18a)$$

$$E'_k(w) = -(y^k - o^k) o^k (1 - o^k) x^k \quad (4.18b)$$

$o^k = 1/(1 + \exp(-w^T x^k))$ olmak üzere.

Dolayısıyla katsayıları güncelleme kuralı şu şekilde yazılabilir:

$$w^{(t+1)} = w^{(t)} + \eta (y^k - o^k) o^k (1 - o^k) x^k \quad (4.19a)$$

veya

$$w^{(t+1)} = w^{(t)} + \eta \delta_k o^k (1 - o^k) x^k \quad (4.19b)$$

Elde edilen denklemler (4.19a) ve (4.19b) ile gizli katmanı olmayan bir ileri beslemeli YSA'nın geri yayımlı öğrenme algoritması şu şekilde yazılabilir:

Eğitim kümesi $\{(x^1, y^1), \dots, (x^K, y^K)\}$ olmak üzere, $x^k = (x_1^k, \dots, x_n^k)$ ve $y^k = (y_1^k, \dots, y_m^k)$, $k=1, \dots, K$.

- Adım 1: $\eta > 0, E_{\max} > 0$ değerleri seçilir.
- Adım 2: Başlangıç ağırlıkları, w , rastgele küçük değerler olarak seçilir.
- Adım 3: Öğrenme bu adımda başlar. Giriş x^k ile çıkış o hesaplanır:

$$o = o(\langle w, x \rangle) = \frac{1}{1 + \exp(-w^T x)} \quad (4.20)$$

- Adım 4: Ağırlıklar güncellenir

$$w^{(t+1)} = w^{(t)} + \eta(y^k - o^k) o^k (1 - o^k) x^k \quad (4.21)$$

- Adım 5: Kümülatif döngü hatası, anlık hatanın E 'ye eklenmesiyle hesaplanır

$$E^{(t+1)} = E^{(t)} + \frac{1}{2}(y - o)^2 \quad (4.21)$$

- Adım 6: Eğer $k < K$ ise $k := k + 1$, bir sonraki eğitim verisine geçilir. Adım 3'e geri dönülür ve eğitime devam edilir. Eğer $k = K$ ise Adım 7'ye geçilir.
- Adım 7: Öğrenme döngüsü tamamlanmıştır. Eğer $E < E_{\max}$ ise eğitim biter; arzu edilen çıkışlar öğrenilmiştir. Eğer $E > E_{\max}$ ise E sıfır yapılır ve Adım 3'e dönülerek yeni bir öğrenme adımı başlatılır [7].

Gradyan yöntemiyle en uygun ağırlıklar yukarıdaki yöntemle bulunmuş olur. Ancak ara katmanları olan bir YSA'da türevleri hesaplamak basit değildir; genelde hata eğrisinin tek minimumu olmaz. Dolayısıyla, eğitimin yerel bir minimumda takılıp kalma riski vardır.

4.8.2 Genelleştirilmiş delta kuralı

Elde edilen delta kuralının ileri beslemeli çok katmanlı YSA'lar için genelleştirilmesi bu bölümde incelenecektir. Genelleştirilmiş delta kuralı ileri beslemeli ağırları eğitmek için en çok kullanılan yöntemdir. Basitleştirmek için tek gizli katmanı olan bir yapay sinir ağının delta öğrenme kuralı ele alınmıştır.

Giriş-çıkış eğitim örüntüsü (x^k, y^k) olsun, (x^k, y^k) örüntüsünün hata fonksiyonu şu şekilde benzer biçimde tanımlanmış olsun:

$$E_k(W, \omega) = \frac{1}{2}(y^k - O^k)^2 \quad (4.22)$$

O^k , ağın hesaplanan çıkışıdır. Hatanın genel değerlendirilmesi şu şekilde yapılır:

$$E(W, \omega) = \sum_{k=1}^K E_k(W, \omega) \quad (4.23)$$

Eğitim kümesinden x^k verisi YSA'na girdiğinde çıkış

$$O^k = \frac{1}{1 + \exp(-W^T o^k)} \quad (4.24)$$

olur. o^k , gizli katmanın çıkış vektörüdür.

$$o_{\ell}^k = \frac{1}{1 + \exp(-\omega_{\ell}^T x^k)} \quad (4.25)$$

ω_{ℓ} , ℓ . Gizli nöronun ağırlık vektörüdür, $\ell = 1, \dots, L$.

Gradyan metoduyla ağırlıkların güncellenip kuadratik hata fonksiyonunun minimize edilmesi için şu iteratif yöntem kullanılır:

$$W^{(t+1)} = W^{(t)} - \eta \frac{\partial E_k(W, \omega)}{\partial W}, \quad (4.26)$$

$$\omega_{\ell}^{(t+1)} = \omega_{\ell}^{(t)} - \eta \frac{\partial E_k(W, \omega)}{\partial \omega_{\ell}} \quad (4.27)$$

$\ell = 1, \dots, L$ ve $\eta > 0$ için.

Zincir kuralından (chain rule) yararlanarak türevler şu şekilde hesaplanır:

$$\frac{\partial E_k(W, \omega)}{\partial W} = \frac{1}{2} \frac{\partial}{\partial W} \left[y^k - \frac{1}{1 + \exp(-W^T o^k)} \right]^2 \quad (4.28a)$$

$$\frac{\partial E_k(W, \omega)}{\partial W} = -(y^k - O^k) O^k (1 - O^k) o^k \quad (4.28b)$$

Böylece, çıkış bağlantıları için ağırlık değiştirme kuralı şu şekilde olur:

$$W^{(t+1)} = W^{(t)} + \eta (y^k - O^k) O^k (1 - O^k) o^k = W^{(t)} + \eta \delta_k o^k \quad (4.29a)$$

yani,

$$W_{\ell}^{(t+1)} = W_{\ell}^{(t)} + \eta \delta_k o_{\ell}^k \quad (4.29b)$$

Denklemdaki δ_k yerine $\delta_k = (y^k - O^k)O^k(1 - O^k)$ yazılır. E_k 'nin ω_ℓ 'ye göre kısmi türevi şu şekilde hesaplanır:

$$\frac{\partial E_k(W, \omega)}{\partial \omega_\ell} = -O^k(1 - O^k)W_\ell o_\ell^k(1 - o_\ell^k)x^k \quad (4.30)$$

denklem 4.29b ve 4.30 ile gizli katmanların ağırlıkları değiştirme kuralı şu şekilde olur:

$$w_\ell^{(t+1)} = w_\ell^{(t)} + \eta \delta_k W_\ell o_\ell^k(1 - o_\ell^k)x^k, \ell = 1, \dots, L \quad (4.31)$$

ve

$$w_{\ell j}^{(t+1)} = w_{\ell j}^{(t)} + \eta \delta_k W_\ell o_\ell^k(1 - o_\ell^k)x_j^k, j = 1, \dots, n \quad (4.31b)$$

İleri beslemeli tek gizli katmanı olan YSA'nın öğrenme algoritması şu şekilde yazılabilir:

- Adım 1: $\eta > 0, E_{\max} > 0$ değerleri seçilir.
- Adım 2: Başlangıç ağırlıkları, w , rastgele küçük değerler olarak seçilir E sıfır seçilir.
- Adım 3: Öğrenme bu adımda başlar. Giriş x^k ile çıkış O hesaplanır:

$$O = \frac{1}{1 + \exp(-W^T o_\ell)} \quad (4.32)$$

o_ℓ gizli katmanın çıkış vektörüdür,

$$o_\ell = \frac{1}{1 + \exp(-\omega_\ell^T x)} \quad (4.33)$$

- Adım 4: Çıkış katmanının ağırlıkları güncellenir

$$W^{(t+1)} = W^{(t)} + \eta \delta o \quad (4.34)$$

- Adım 5: Gizli katmanların ağırlıkları güncellenir

$$w_\ell^{(t+1)} = w_\ell^{(t)} + \eta \delta W_\ell o_\ell(1 - o_\ell)x \quad (4.36)$$

- Adım 6: Kümülatif döngü hatası, anlık hatanın E 'ye eklenmesiyle hesaplanır

$$E^{(t+1)} = E^{(t)} + \frac{1}{2}(y - o)^2 \quad (4.37)$$

- Adım 7: Eğer $k < K$ ise $k := k + 1$, bir sonraki eğitim verisine geçilir. Adım 3'e geri dönülür ve eğitime devam edilir. Eğer $k = K$ ise Adım 7'ye geçilir.
- Adım 8: Bir öğrenme döngüsü tamamlanmıştır. Eğer $E < E_{max}$ ise eğitim biter; arzu edilen çıkışlar öğrenilmiştir. Eğer $E > E_{max}$ ise E sıfır yapılır ve Adım 3'e dönülerek yeni bir öğrenme adımı başlatılır.

Birden fazla ara katmanı olan YSA'nın geri yayılım ile ağırlık güncelleme yöntemi yukarıda anlatılan bir ara katmanı olan yapıya benzer şekilde, kısmi türevler alınırken zincir kuralından yararlanılarak bulunur.

Yukarıda adım adım anlatılan yöntemde ağırlıklar her (x^k, y^k) çiftinden sonra güncellenmektedir. Bu tip öğrenmeye on-line öğrenme denir. Ancak gradyan azalan (gradient descent) yönteminde, her öğrenme adımında hatanın kesin olarak küçülmesi için, ağırlıklar kümülatif hata hesaplandıktan sonra yani bir periyodun (epoch) sonunda güncellenmelidir. Bu tip öğrenmeye batch-mode öğrenme denir. Fakat, güncelleme bu şekilde olursa öğrenme çok yavaş olur. On-line öğrenmede, eğer öğrenme adımı yeterince küçük seçilirse, öğrenme esnasında hatanın osilasyon yapması engellenmiş olur [7].

4.8.3 Delta-Bar-Delta

Delta-Bar-Delta (DBD) çok katmanlı perseptronlarda bağlantı ağırlıklarının yakınsama hızını arttırmak için kullanılan bir sezgisel yaklaşımdır. Deneysel çalışmalar, hata yüzeyinin ağırlık uzayının her boyutunun değişimi (gradyanı) için farklı olabileceğini göstermiştir. Hata yüzeyindeki değişimleri açıklamak için, ağırlık her bağlantısı kendi öğrenme katsayısına sahip olmalıdır. Tek ağırlık boyutu için uygun olan adım büyüklüğü, diğer tüm ağırlık boyutları için uygun olmayabilir. Bununla birlikte, her bir bağlantıya bir öğrenme oranı atanır ve bu öğrenme oranının zamanla değişmesine izin verilirse, yakınsama zamanını azaltmak için daha çok serbestlik derecesi sağlanmış olur.

İleri beslemeli YSA yapıları çoğu zaman karmaşıktır. Ağdaki her bağlantı için en uygun öğrenme katsayıları kümesini belirlemek oldukça zaman alıcı olabilir. Eğimin geçmişteki değerlerini kullanarak, yerel hata yüzeyinin eğriliğini çıkarmak için sezgisellik uygulanabilir. Bir bağlantı için ağırlık değişimlerinin işareti, birkaç ardışık zaman adımları sırayla değiştiği zaman, bağlantı için öğrenme oranı azalmalıdır. Bağlantı ağırlık boyutu hata yüzeyi ile ilgili büyük bir eğriliğe sahiptir. Bağlantı ağırlık değişimleri bir kaç ardışık zaman adımları için aynı işarete sahip olduğundan bağlantı için öğrenme oranı arttırılmalıdır. Ağırlık boyutu hata yüzeyi ile ilgili küçük

bir eğime sahiptir ve yüzey önemli bir mesafe için aynı doğrultudaki eğime göre devam eder [12].

Standart geri yayılım algoritmasında eğitim bileşeni aşağıdaki şekilde verilir.

$$\delta(k) = \frac{\partial E(k)}{\partial w(k)} \quad (4.38)$$

Burada $E(k)$, k anındaki hata değerini, $w(k)$ bağlantı ağırlığını ve $\delta(k)$ ise ağırlık değişiminin eğitim bileşenini göstermektedir. Standart geri yayılım algoritmasında bağlantı ağırlığı,

$$w(k+1) = w(k) + \alpha\delta(k) \quad (4.39)$$

olarak güncelleştirilir. Burada α sabit bir öğrenme oranıdır. DBD öğrenme kuralında, her bağlantı için değişken öğrenme oranı $\alpha(k)$ atanır ve bağlantı ağırlığının güncelleşmesi

$$w(k+1) = w(k) + \alpha(k)\delta(k) \quad (4.40)$$

şeklinde yapılır. Her bağlantıda öğrenme hızının azalması ve artması için sezgiselliğin gerçekleştirilmesinde, $\delta(k)$ eğitim bileşeninin ağırlıklı ortalamasını kullanılmıştır. Bu ağırlıklı ortalama $\bar{\delta}(k)$ 'dir. Özellikle, ağırlıklı ortalama aşağıdaki şekilde gösterilmiştir.

$$\bar{\delta}(k) = (1-\theta)\delta(k) + \theta\bar{\delta}(k-1) \quad (4.41)$$

Burada, θ konveks ağırlık faktörüdür. Sezgisel açıdan, önceki eğitim bileşeninin üstel artması ve şu anki eğitim bileşeni aynı işaretli ise, öğrenme oranı K sabiti ile artan ağırlıkla birleştirilir. Mevcut eğitim bileşeni üstel ortalamadan farklı işaretli ise, öğrenme oranı mevcut değeri ile orantılı olarak azalır. Bu öğrenme oranının güncelleştirilmesi aşağıdaki eşitlikle tanımlanmıştır.

$$\Delta\alpha(k) = \begin{cases} K & \bar{\delta}(k-1)\delta(k) > 0 \\ -\phi\alpha(k) & \bar{\delta}(k-1)\delta(k) < 0 \\ 0 & \text{aksi halde} \end{cases} \quad (4.42)$$

Burada, K öğrenme katsayısı artma faktörü, ϕ öğrenme katsayısı azaltma faktörü, $\alpha(k)$ k anındaki öğrenme oranını göstermektedir. Denklem (4.42)'den görülebileceği gibi algoritma, öğrenme katsayılarını lineer olarak artırmakta, fakat geometrik olarak azaltmaktadır [9].

4.8.4 Genişletilmiş Delta-Bar-Delta

Genişletilmiş Delta-Bar-Delta (GDBD) öğretim algoritması, eğitimde belirli bir anda epoch (periyot) genel hatayı hesaplayarak hata düzeltme özelliğine sahiptir. Belirli bir andaki (ya da herhangi bir anda) ölçülen hata, bir önceki anda bulunan hatadan büyük ise ağırlıkların bir önceki ağırlık değerlerini alır. Daha iyi sonuçlar elde etmek için hata düzeltme özelliğine dayanıklılık (patience) faktörü eklenmiştir. Her periyottaki (epoch) hatayı test etmek yerine daha önce iyileştirilme işlemi yapıldığından hata, n periyotta test edilir.

EDBD'nin DBD'den farkı aşağıda özetlenmiştir.

- Sezgisel momentum kullanılır.
- Ağırlık uzayında büyük atlamalar ortadan kaldırılmıştır.
- Geometrik azaltma büyük amacı aşan atlamaları engellemede yeterince hızlıdır.

GDBD'de, her bir bağlantıdaki momentum hızı $\mu(k)$ ve öğrenme hızı $\alpha(k)$ zamanla değişmektedir. DBD ile karşılaştırmak için standart delta kuralı ile verilen momentum için bağlantı ağırlıklarının değiştirilmesi

$$\Delta w(k+1) = \alpha \delta(k) + \mu \Delta w(k) \quad (4.43)$$

değerinin ağırlıklara katkısıyla

$$w(k+1) = w(k) + \Delta w(k+1) \quad (4.44)$$

elde edilebilir. Burada α ve μ sabittir. Bunun yanında EDBD için değişken öğrenme ve değişken momentum hızları aşağıdaki gibi elde edilebilir. Ağırlıklar,

$$\Delta w(k+1) = \alpha(k) \delta(k) + \mu(k) \Delta w(k) \quad (4.45)$$

değerinin bir önceki ağırlığa eklenmesiyle

$$w(k+1) = w(k) + \Delta w(k+1) \quad (4.46)$$

bulunur. Burada, $\mu(k)$ k zamanındaki bağlantı momentum hızıdır. Momentum hızı $\mu(k)$ ve öğrenme hızı $\alpha(k)$ aşağıdaki kurallara göre benzer şekilde ayarlanır. İlk önce, $\bar{\delta}$ 'nin hesaplanması DBD algoritmasına göre belirlenir.

$$\bar{\delta}(k) = (1 - \theta) \delta(k) + \theta \bar{\delta}(k-1) \quad (4.47)$$

Burada konveks ağırlık faktörüdür. EDBD için öğrenme hızı değişimi:

$$\Delta\alpha(k) = \begin{cases} K_\alpha \exp(-\gamma_\alpha |\bar{\delta}(k)|) & \bar{\delta}(k-1)\bar{\delta}(k) > 0 \\ -\varphi_\alpha \alpha(k) & \bar{\delta}(k-1)\bar{\delta}(k) < 0 \\ 0 & \text{aksi halde} \end{cases} \quad (4.48)$$

Formülünden elde edilir. Burada \exp , üstel fonksiyonunu, K_α , sabit öğrenme hızı skala faktörünü φ_α , sabit öğrenme hızı azaltma faktörünü ve γ_α , sabit öğrenme hızı üstel faktörünü göstermektedir.

Benzer şekilde, k zamanındaki momentum hızı değişimi,

$$\Delta\mu(k) = \begin{cases} K_\mu \exp(-\gamma_\mu |\bar{\delta}(k)|) & \bar{\delta}(k-1)\bar{\delta}(k) > 0 \\ -\varphi_\mu \alpha(k) & \bar{\delta}(k-1)\bar{\delta}(k) < 0 \\ 0 & \text{aksi halde} \end{cases} \quad (4.49)$$

ifadesinden bulunabilir. Burada, K_μ , sabit momentum hızı skala faktörünü φ_μ , sabit momentum hızı azaltma faktörü ve γ_μ , sabit momentum hızı üstel faktörünü göstermektedir.

Şuna da dikkat edilmelidir ki öğrenme ve momentum hızları, azalmayı ve artmayı kontrol eden farklı sabitlere sahiptir. $\bar{\delta}(k)\delta(k)$ 'nin işareti sezgisel olarak artma mı yoksa azalma mı olduğunu tespit etmek için kullanılır. Azalmanın ayarlanması DBD'nin kurallarına göre belirlenir. Ancak öğrenme ve momentum oranlarının artırılması $|\bar{\delta}(k)|$ 'nin üstel azalan fonksiyonu olarak değiştirilir. Böylece, küçük eğimli bölgelerde, büyük eğimli bölgelere nazaran daha büyük artırımlar yapılabilecektir. Bu, atlama problemi için kısmi bir çözümdür.

Ağırlık uzayında, osilasyonları ve aşırı atlamaları engellemek için, üst sınırlar, her bir bağlantı öğrenme hızları ve momentum hızları üzerinde belirlenir. Matematiksel ifadesi, tüm bağlantılar için aşağıdaki gibidir.

$$\alpha(k) \leq \alpha_{\max} \quad (4.50)$$

$$\mu(k) \leq \mu_{\max} \quad (4.51)$$

Burada, α_{\max} öğrenme katsayısının üst sınırı ve μ_{\max} ise momentum katsayısının üst sınırıdır. Sonuç olarak, GDBD algoritmasına düzeltme özellikli bir hafıza yerleştirilmiştir. Bu kullanıldığı zaman, her iterasyondan sonra hata değerlendirilir. Hata, $E(k)$, bir önceki minimum hatadan küçük ise ağırlıkları o anki en iyiler olarak

hafızada saklanır. Düzeltme tolerans parametresi λ , düzeltme sürecini kontrol eder. O andaki hata önceki minimum hatayı aşarsa, yani

$$E(k) > E_{\min} \lambda \quad (4.52)$$

olursa, tüm bağlantı ağırlıkları, hafızada saklı olan en iyi değerlerle değiştirilir. Ayrıca, öğrenme ve momentum hızları, düzeltmeyi başlatmak için azaltılır[9].

4.8.5 Hızlı yayılım

Hızlı yayılım (Quickprop) algoritması, Scott Fahlman tarafından Carnegie Mellon Üniversitesi'nde geliştirilen ve Newton metoduna dayanan, çok katlı perseptronların eğitilmesi için kullanılan bir sezgisel öğretim algoritmasıdır.

Sezgisel algoritmalar, tecrübeye ve eğitilmiş tahmine dayalı kurallar içermektedir. Bu algoritmalar optimum çözümü garanti edemeyip sadece ona yakın çözümü bulmayı garanti etmektedir.

Sezgisel metotları kullanmanın sağladığı bazı avantajlar şöyle özetlenebilir.

1. Sezgisel metotlar karar verici mekanizma için sadeleştirici olabilir.
2. Sezgisel metot, herhangi bir tam yöntemin parçası olarak öğrenme amacıyla kullanılabilir.
3. Gerçek dünya problemleri için her zaman matematiksel formülasyon kurmak çok kolay olmayabilir. Bu basitleştirme sonucu oluşan hata, bir sezgisel metodun sağladığı optimale yakın çözümün sahip olduğu hatadan daha büyük olabilir.

Problem uzayında mümkün olan en kısa sürede, bir çözüm bulmak için ağırlık uzayında eğimi (gradyan) son derece küçük azaltmak gerekebilir. Bunun yerine çözümden fazla uzaklaşmaksızın mümkün olan en büyük adımlarla kısa sürede çözüme ulaşmak arzu edilir. Hata fonksiyonunun eğim derecesi biliniyorsa, güvenli bir şekilde ağırlık uzayında daha büyük adımla çözüme ulaşılabilir. Bu algoritmada iki geleneksel yaklaşım birleştirilerek çözülmüştür. Bunlar;

- Hesaplamanın geçmişteki durumu hakkında bazı sezgilere dayanan (genel veya her bir ağırlık için ayrı) dinamik olarak ağırlıkların ayarlanması.
- Her bir ağırlığa göre hatanın ikinci türevinin belirgin kullanımı
- Hızlı yayılım algoritması bir ikinci dereceli metoddan daha sezgiseldir ve çok az da olsa Newton-metoduna dayanan bu metoddaki kabul kullanılır.
- Her bir ağırlık için, ağırlık hata eğrisi kolları yukarı doğru açık olan bir parabol ile yaklaştırılabilir.

- Hata eğrisinin eğimindeki değişim, diğer tüm ağırlıkların aynı andaki değişiminden etkilenmez.

Yukarıda bahsettiğimiz parabol, birbirinden bağımsız her bir ağırlık için şimdiki ve önceki hata eğimleri ve bu eğimlerin ölçüldüğü noktalar arasındaki değişim kullanılarak belirlenir. Daha sonra, algoritma doğrudan bu parabolün minimum noktasına atlar.

Genellikle hızlı yayılım algoritmasının performans testleri diğer tekniklerle karşılaştırıldığında oldukça iyidir. Bu algoritma, özellikle gürültü seviyesi az olan dataya sahip problemlerde, EDBD kadar iyi sonuç vermektedir. Hızlı yayılım algoritması gürültü seviyesi fazla olan problemlerde çok iyi çalışır ancak gürültülü dataların bulunduğu problemlerde EDBD kullanılmalıdır.

Bu algoritmada ağırlık hızlandırma ve kırılması ihmal edilerek $(t-1)$ 'den t 'ye kadar ağırlıklardaki değişim,

$$\Delta w(t) = \eta L(t) + \alpha Q(t) \quad (4.53)$$

formülü ile verilir. Burada η öğrenme katsayısı ve α ise momentum katsayısıdır.

$$L(t) = \begin{cases} h(t) & h(t)h(t-1) \geq 0 \text{ ise} \\ 0 & \text{aksi halde} \end{cases} \quad (4.54)$$

ve

$$Q(t) = \begin{cases} \mu \Delta w(t-1) & h(t) \left(h(t) - \left(\frac{\mu}{\mu+1} \right) h(t-1) \right) \geq 0 \text{ ise,} \\ \Delta q(t) & \text{aksi halde} \end{cases} \quad (4.55)$$

eşitlikleri ile verilir. Bu ifadelerde μ momentum büyüme faktörü, $h(t) = \partial E / \partial w(t)$

eğimi ve $\Delta q(t) = \frac{\Delta w(t)h(t)}{h(t-1) - h(t)}$ minimum adım miktarını göstermektedir. Buradan

ağırlık fonksiyonunu güncelleştirmek için, delta ağırlık fonksiyonu ve ağırlık hızlandırma katsayısı işleme katılır.

$$w(t) = (1 - \delta)w(t-1) + \Delta w(t) \quad (4.56)$$

Burada δ hızlandırma katsayısıdır. Son olarak ağırlık çok küçük ise, 0 alınarak kırılır.

$$|w(t)| < K \text{ ise } w(t) = 0 \quad (4.57)$$

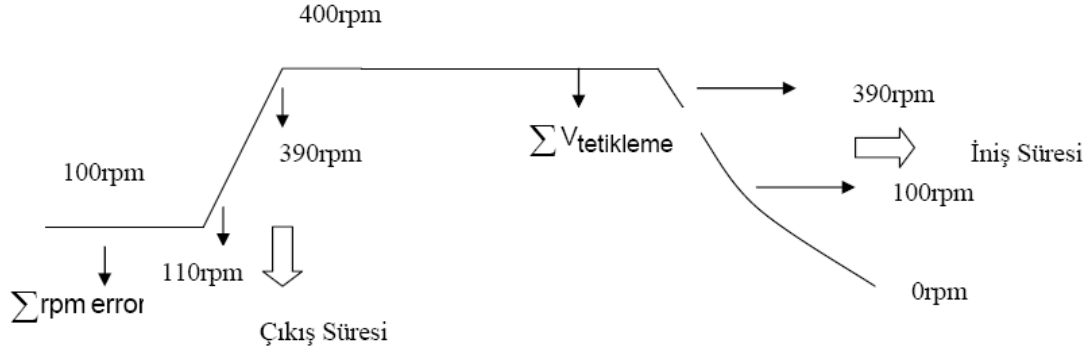
burada, K ağırlık kırılma faktörüdür [9].

ÇAMAŞIR YÜKÜ TAHMİN MODELİ

5.1 Çamaşır Yükünü Belirleme Profili

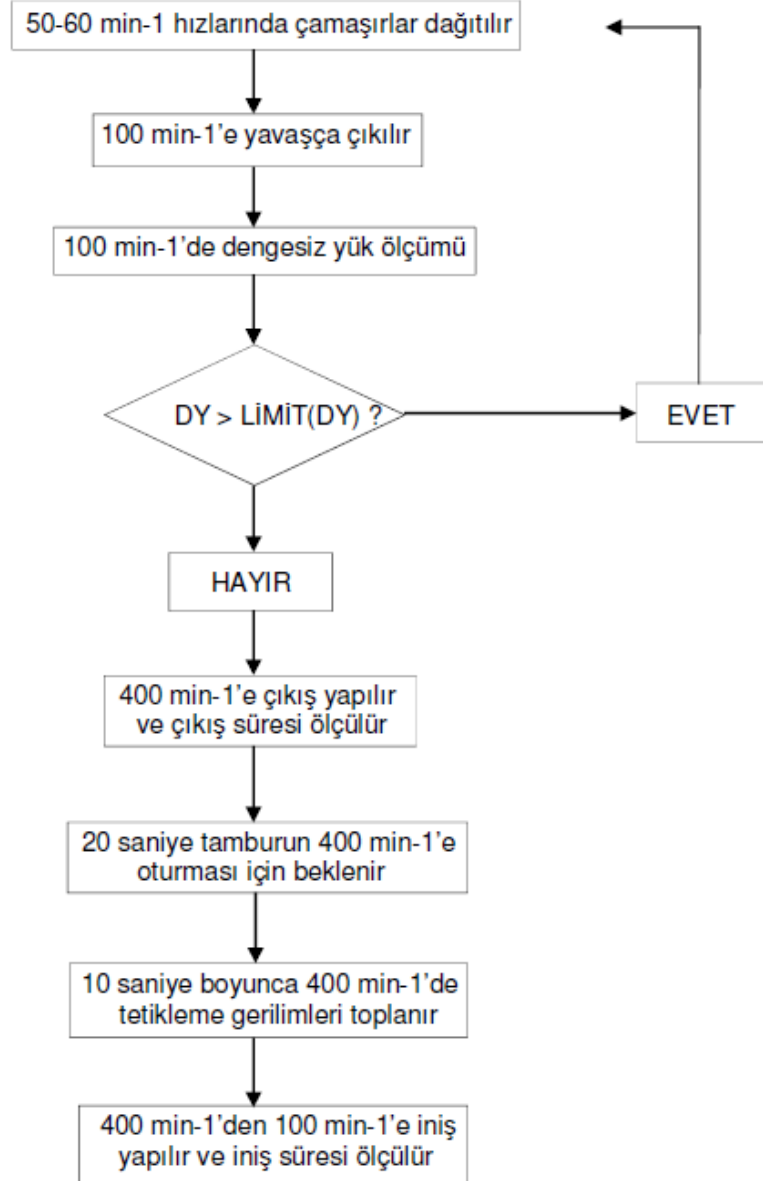
Çamaşır makinesinin içindeki çamaşır yükünün kestirimi için veri kümesine ihtiyaç vardır. Bu veri kümesi oluşturulurken çamaşır yükü ile bu kümenin elemanları arasında yüksek korelasyon olmasına dikkat edilmelidir. Çamaşır, makine içinde tamburun dönüşü ile hareket ettiğinden, tambura belirli bir hareket profili uygulamak, çamaşır makinesi içindeki çamaşır yükü miktarını kestirmede kullanılacak verilerin elde edilmesini sağlar. Bu tez çalışmasında kullanılan veriler, böyle bir hareket profilinden elde edilmiştir.

Şekil 5.1 ve Şekil 5.2 sırasıyla yük ve dengesiz yük belirleme profili ve akış şeması verilmiştir. Çamaşır, 50-60 min^{-1} olan yıkama hızından, yaklaşık 10 saniyede 100 min^{-1} 'e çıkarılarak dengeli dağılma ve yapışma sağlanmaya çalışılır (çamaşırın tamburun duvarlarına yapıştığı 70 min^{-1} hızının üstündeki hızlara sıkma hızları, altındaki hızlara yıkama hızları denir). 100 min^{-1} 'de dengesiz yük ölçümü yapılır, 0 kg dengeli yük bazında 400 min^{-1} 'e dahi çıkmayı tehlikeli gösteren bir dengesiz yük algılanırsa, tekrar çamaşır dağıtma adımına dönülür, birkaç tekrardan sonra eğer hala 400 min^{-1} 'e çıkılamıyorsa hiç sıkma yapılmaz. Böyle bir dengesizlik yoksa, 5 saniye gibi kısa bir rampa süresi ile 400 min^{-1} 'e çıkılır, ve 170 min^{-1} civarı olan rezonans bölgesi hızlı geçilmeye çalışılır. 400 min^{-1} az tehlikeli, düşük bir sıkma hızı olduğundan ortalama gerilim ve iniş zamanı ölçümlerinin etkin çalışacağı bir hız seviyesidir. 400 min^{-1} , yeterli çözünürlükte iniş zamanı ölçümü için uygundur. Tambur hızı 400 min^{-1} 'e oturduktan sonra 20 saniyelik sıkmanın 15. ve 20. saniyeleri arası motora uygulanan gerilim ölçülüp, ortalaması hesaplanır. 20. saniyede tamburu döndüren elektrik motoruna verilen enerji kesilir ve 400 min^{-1} 'den 100 min^{-1} 'e iniş zamanı ölçümü başlatılır. Tambur hızının 100 min^{-1} 'e indiği tespit edildiğinde iniş zamanı ölçümü bitirilir ve kaydedilir.



Şekil 5.1 : Yük kestirim profili.

ÇAMAŞIR YÜKÜNÜ BELİRLEME PROFİLİNİN AKIŞ DİYAGRAMI

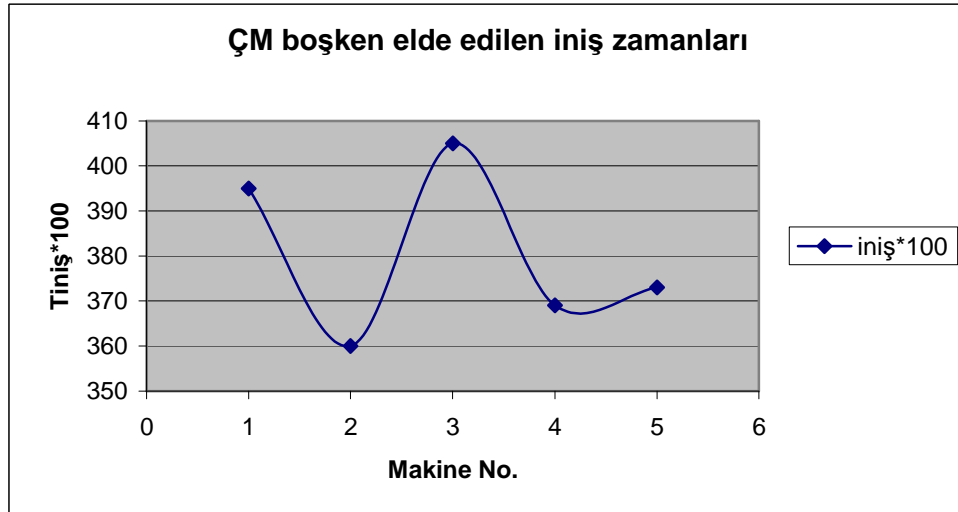


Şekil 5.2 : Yük belirleme profili akış diyagramı.

5.2 amařır Makinesinde Yk Kestirimi iin Kullanılan Verilerin İncelenmesi

Yk algılama zerine yapılan alıřmalar, alınan patentlerin sıklığına bakıldıėında anlaşılacağı zere daha ok sensrsz yntemler zerinde durulmuřtur. Sensrsz yntemlerde, amařır makinesi belirli bir algoritmaya gre belirli hareketler yapar ve bunun sonucunda bazı veriler elde edilir. Bu veriler, yk kestirim algoritmasında girdi olarak kullanılır ve yk kestirim algoritmasından yk tahmin sonucu ıkar.

2. blmde bahsedilen patentlerin bir kısmında, amařırın eylemsizliėine dayanarak elde edilen verilerin, tablo ile deėerlendirilerek yk tahmini yapıldığından bahsedilmektedir ve bu patentlerde genelde sadece bir veya iki veri deėerlendirilmektedir. Ancak, sadece bir veya iki veriyi deėerlendirmek, her makinede ve her kořulda en iyi sonucu vermeyecektir. rneėin, farklı makineler ele alındığında, srtnmeler farklı olacaktır; dolayısıyla ıkıř-iniř zamanları aynı ykler iin farklılıklar gsterecektir. Őekil 5.3'te farklı makineler iin tambur bořken kaydedilmiř iniř zamanları incelenirse, bu fark gzlenebilir (iniř sresi tambur bořken lldėi iin dengesiz daėılmıř ykn olmadığı varsayılabilir). Ayrıca, aynı makineye farklı zamanlarda aynı amařır yk konmuř olsa dahi, dengesiz yk miktarlarının farklı olması durumunda, yine iniř-ıkıř zamanlarında nemli farklar gzlenecektir. Bu sebeplerle, amařır ykn kestirmek iin uygulanabilir, ticari deėeri olabilecek bir yntem geliřtirmek iin her nemli veri dikkate alınmalıdır. Yntemin performansı aynı tip makineler arasında farklılık gstermemelidir.



Őekil 5.3 : amařır makineleri bořken elde edilen iniř sreleri.

Yntem belirlenirken, tamburun iindeki yk miktarıyla iliřkili olabilecek deėiřkenler ve kullanılacak veriler de belirlenmelidir. Bu verilerin, amařır makinesi tarafından

herhangi ek bir sensör gerekmeden, ya da genel olarak herhangi ek bir donanım gerekmeden elde edilebiliyor olması, ek maliyet getirmeyeceği için avantaj sağlayacaktır.

Yükü kestirmek için sadece bir veriyi değerlendirmek yetmeyecektir. Az hatayla yükü tahmin edebilmek için birkaç veriden aynı anda yararlanmak gerekir. Bu tez çalışmasında çamaşır yükünün kestirimini başarmak için kullanılan veriler ve gözlenen durumlar şunlardır:

1. Dengesiz yük miktarı: tambur sabit hızda (100 min^{-1}) dönerken hız salınım miktarı
2. Çıkış zamanı: tamburun 100 min^{-1} 'den 400 min^{-1} 'e çıkış süresi
3. Sürtünme: tambur 400 min^{-1} dönerken çektiği akım
4. İniş zamanı: tamburun 400 min^{-1} 'den 100 min^{-1} 'e iniş süresi
5. Şebeke gerilimi: çamaşır makinesi motorunun giriş gerilimi

Bu veriler dışında, 100 min^{-1} 'den 400 min^{-1} 'e çıkarken motora verilen enerjiyi yansıtan uygulanan gerilimin integrali, ya da faz kontrollü bir devreyle motor AC gerilimle sürülüyorsa tetikleme açılarının toplamı kullanılır. Bu tezde kullanılan çamaşır makineleri motorları AC gerilimle sürülmektedir. Elektrik motoru üniversal motordur.

Yukarıda bahsi geçen verileri kullanarak çamaşır miktarını kestirecek olan yöntem, kullanılan verilerin özelliğinden de anlaşılacağı gibi çamaşırın tambur içindeki eylemsizliğine dayanan bir yöntemdir. Bunun sebebi, çıkış ve iniş süreleri doğrudan çamaşır yükünün eylemsizliği ile ilişkilidir. Dengesiz yük miktarı, sürtünme ve şebeke gerilimine ilişkin veriler ise, eylemsizlikle birlikte çıkış-iniş zamanını etkileyen diğer faktörler oldukları için kullanılmaktadır.

Yük kestirim algoritması, çamaşırın eylemsizliğine dayandığına göre; çamaşırın kütlesiyle, çamaşırın tambur içindeki eylemsizliği arasındaki ilişkiyi değerlendirmek gerekir. Bu ilişkiyi açıklamak için bazı ön kabuller yapılabilir. Şekil 5.4'de gösterildiği gibi sıkma hızlarında çamaşır homojen (özkütlesi değişmeyen) ve düzgün dağılmış olarak kabul edilirse, çamaşırın kütlesiyle eylemsizliği arasındaki bağıntı şu şekildedir:

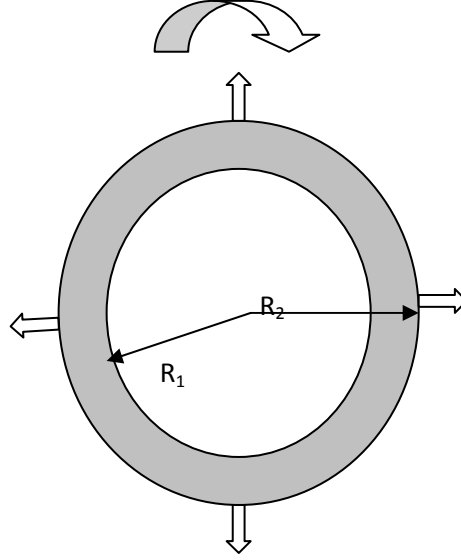
$$J = \frac{M}{2} * (R_1^2 + R_2^2) \quad (5.1)$$

J : Eylemsizlik

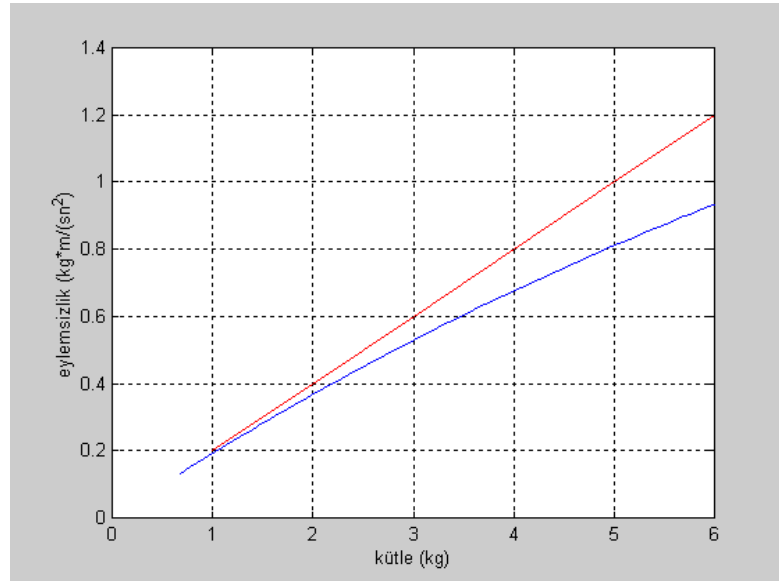
M : Çamaşırın kütlesi

R_1 : Tambur merkezinin çamaşıra olan en yakın uzaklığı

R_2 : Tamburun yarı çapı



Şekil 5.4 : Homojen dağıldığı varsayılan çamaşırın basit 2 boyutlu modeli.



Şekil 5.5 : Tambur içindeki çamaşırın kütle-eylemsizlik ilişkisi (mavi eğri).

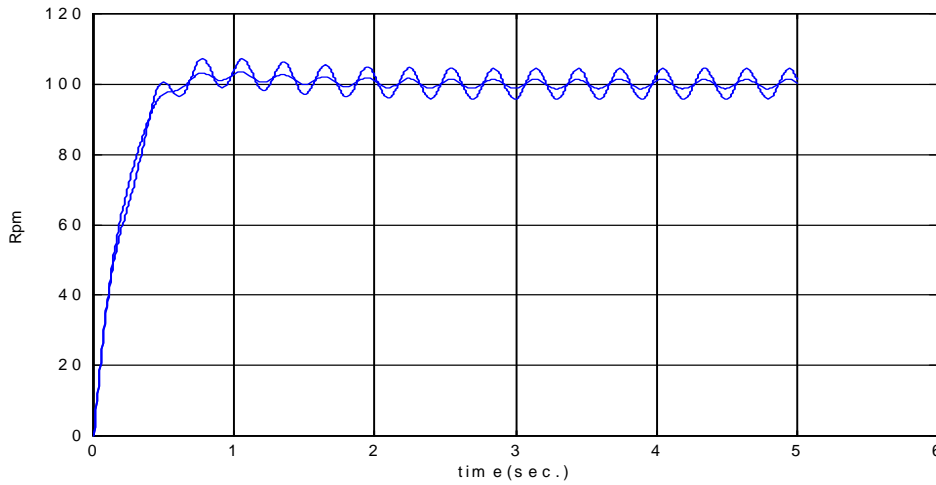
Denklem (5.1)'deki kütle-eylemsizlik ilişkisini gösteren şekil 5.5'te de görüldüğü gibi kütleyle eylemsizlik arasında lineer olmayan bir ilişki vardır. Lineer olmama durumu, kütle arttıkça daha da belirginleşmektedir. Örneğin, 5kg çamaşır ile 6kg çamaşır arasındaki eylemsizlik farkı, 1kg çamaşır ile 2kg çamaşır arasındaki eylemsizlik farkından daha azdır.

Bu bölümün devamında, gözlenen durumların çamaşır makinesindeki yükü tahmin edebilmek için neden gerekli oldukları incelenmiştir.

5.2.1 Dengesiz yük

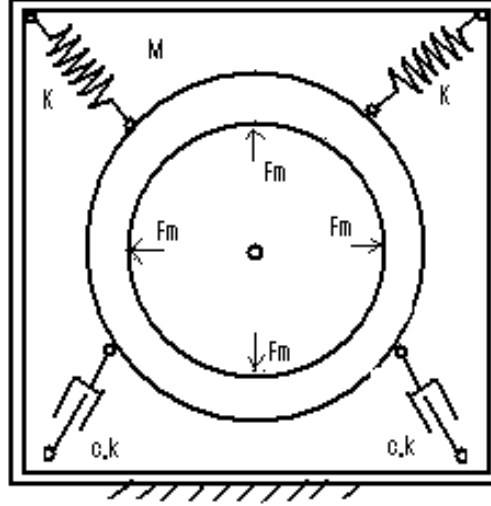
Çamaşır makinesinde dengesizlik, tamburun ağırlık merkezinin dönme eksenine ile çakışmaması durumunda ortaya çıkar. Gevşek bağlı yatak elemanları, mil kaçıklığı, millerdeki eğilme, malzemenin homojen olmaması ve en baskın sebep olarak da tambur içindeki çamaşır yükünün eşit dağılmaması, çamaşır makinesinde dengesizliğe neden olan etkenlerdir. Sıkma işlemi sırasında, çamaşır makinesi tamburu içinde, düzgün dağılmamış çamaşırlar bulunması durumunda oluşan dengesiz yük, bir merkezkaç kuvveti oluşturur. Bu kuvvetin şiddeti, dengesiz yükün konumu, miktarı ve dönme hızıyla doğru orantılıdır.

Çamaşır makinesi motorunun hız kontrolü, klasik ayırık zamanda PI (orantı-integral) kontrolü ile yapılır. Bu kontrol yönteminde, sistem davranışı sürekli hale geçtiğinde hata (referans – ölçülen hız) integral etkisinden dolayı sifıra yakınsar. Çamaşır makinesinde hız kontrolünde de bu durum gözlenir. Ancak çamaşır dağılımında dengesizlik varsa, şekil 5.6'da gösterildiği gibi, tamburun hızı sürekli halde referans hızın etrafında sinüzoidal salınım yapar [14].



Şekil 5.6 : Farklı dengesiz yüklerde sabit hız kontrolünde salınım.

Çamaşır makinesinin fiziksel modelinin şeması, simetrik yapısından dolayı şekil 5.7'deki gibi iki boyutlu düzlemde verilebilir.



Şekil 5.7 : Çamaşır makinesinin iki boyutlu düzlemde fiziksel modeli [14].

Şekildeki sistemin, k direngenlik ve c viskoz sönümlenme miktarına sahip iki adet amortisör ile gövdeye bağlandığı düşünülmüştür. Modelde M dinamik titreşime maruz kalan toplam kütle, m dönen dengesiz kütle miktarını ve Ω tamburun açılma hızını ifade etmektedir.

Şekil 5.7'de sistemde dönme eksenine ile birlikte yatay ve dikey doğrultularda hareket serbestliğinden bahsedilebilir. Gövdenin zemine rijit bir şekilde bağlı olduğu varsayılarak dikey ve yatay doğrultudaki hareket denklemleri sırasıyla aşağıdaki gibi elde edilir.

$$(M + m_{\text{dengeli}} + m_{\text{dengesiz}}) \ddot{x} + c_x \dot{x} + F_s \text{sign}(\dot{x}) + k_x x = m_{\text{dengesiz}} e \Omega^2 \cos(\Omega t) \quad (5.2)$$

$$(M + m_{\text{dengeli}} + m_{\text{dengesiz}}) \ddot{y} + c_y \dot{y} + F_s \text{sign}(\dot{y}) + k_y y = m_{\text{dengesiz}} e \Omega^2 \sin(\Omega t) \quad (5.3)$$

(5.2) ve (5.3) denklemlerinde

m_{dengeli} : dengeli çamaşır miktarını

m_{dengesiz} : dengesiz çamaşır miktarını

r : dengesiz yük'ün dönme eksenine olan uzaklığını

e : dengesiz çamaşırın dönme ekseninden uzaklığını

c_x, c_y : sönümlenme miktarını

F_D : kuru sürtünmeden kaynaklanan sürtünme kuvvetini

k_x, k_y : direngenlik miktarını

$m\Omega^2$: merkezkaç kuvvetinin büyüklüğünü

ifade etmektedir [14].

Dengesiz yükün yatay ve düşey düzlemdeki harekete etkisini inceleyebilmek için denklemler lineerize edilirse, sistemde, dengesiz çamaşırın dönmesinin yarattığı merkezkaç kuvveti dolayısıyla oluşan deplasman miktarı, açısal tambur hızının bir fonksiyonu olarak bulunabilir.

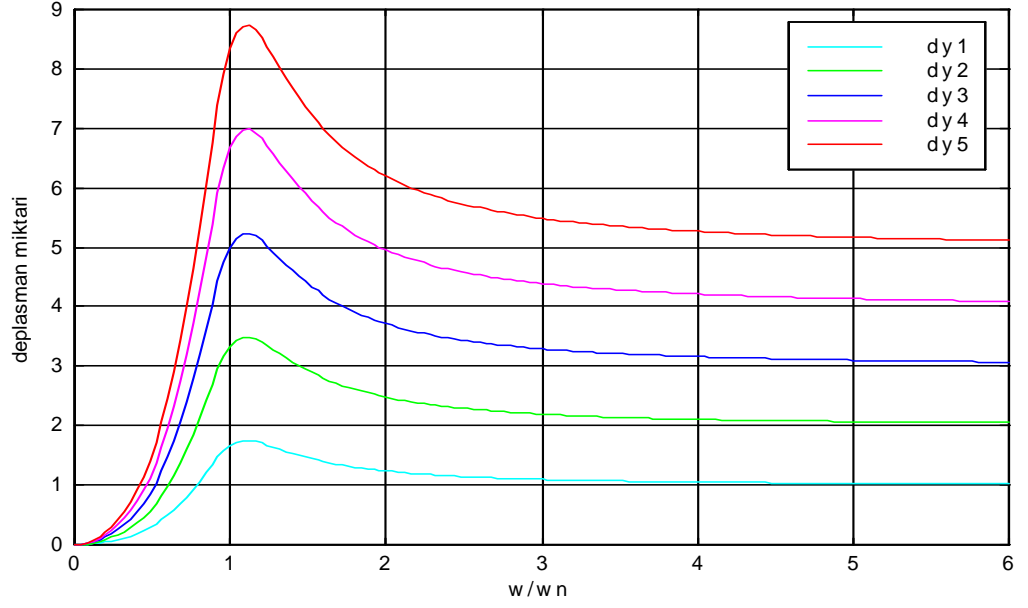
$$|H(\Omega)| = \frac{1}{\sqrt{\left(1 - \frac{\Omega^2}{\Omega_n^2}\right)^2 + \left(2\zeta \frac{\Omega}{\Omega_n}\right)^2}} \quad (5.4)$$

$$e = \frac{m_{\text{dengesiz}} \Gamma}{M} \quad (5.5)$$

$$\text{titresimgenligi} = \frac{m_{\text{dengesiz}} e}{k} |H(\Omega)| \quad (5.6)$$

(5.5) ve (5.6) denklemlerinden görüldüğü üzere titreşim genliği dengesiz yük miktarı ile doğru orantılı olarak artmakta, dengeli yük miktarının artışıyla sabit dengesiz yükte azalmaktadır [14].

Şekil 5.8'de sabit dengeli yükte, farklı dengesiz yük miktarları için açısal hıza bağlı deplasman miktarları verilmiştir. Sabit hızlarda dengesiz yükün miktarıyla doğru orantılı davranan deplasman miktarı, herhangi bir dengesiz yük miktarında rezonans hızda kendi maksimumundan geçer. Dengesiz yükün hasarlarından özellikle sakınılması gereken noktaların başında da rezonans hız gelir.



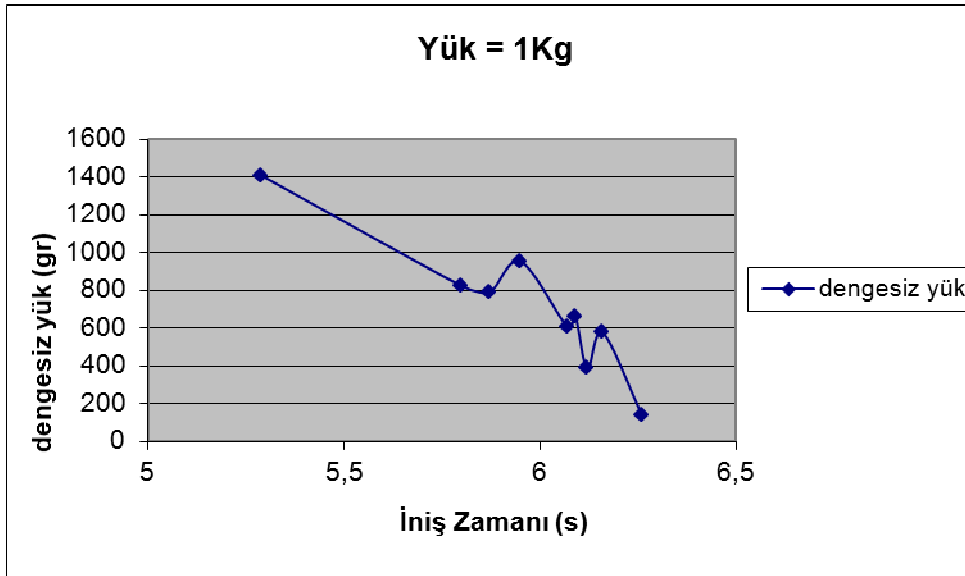
Şekil 5.8 : Deplasman miktarının farklı dengesiz yüklerle açısal hıza göre değişimi [14].

Dengesiz yükün varlığı, çamaşır makinesi motorunun hız ve akım şekillerinde gözlemlenebilir. Tambur içinde bir bölgede yoğunlaşmış dengesiz yük, tamburun aşağıya doğru hareketi esnasında yerçekimi kuvvetinin etkisinde potansiyel enerjisini kaybederek hızlanmakta ve dengesiz yükün dönme merkezinden uzaklığı ile doğru orantılı olarak dönüş yönünde bir moment üretmektedir. Dengesiz yük yukarıya doğru çıkarken ise potansiyel enerji kazanarak yavaşlamakta ve dönüş yönüne ters bir moment üretmektedir. Motordan çekilen akım veya akımın karesinin üretilen momentle doğru orantılı olduğu düşünüldüğünde, motor hızında ve akımında dengesiz yük miktarıyla, genliği orantılı, frekansı dönüş devri ile belirlenen ideal durumda sinüzoidal dalgalanmalar olur. Sistemin bu özelliğinden dolayı dengesizlik ölçümü hız dalgalanması gözlenerek gerçekleştirilmiştir. Dengesiz yük ölçümü için, tambur hızı çamaşırın tambur çeperine yapıştığından garanti edilebileceği ve rezonans hızdan yeterince uzak olan 100 min^{-1} 'de kapalı çevrim kontrol edilmeye çalışılır. Dengesiz yükün miktarıyla doğru orantılı olan hız salınımları, hız salınımlarının standart sapması değerlendirilir ve dengesiz yük miktarı tahmin edilir.

Ölçüm öncesinde, tambur hızının 100 min^{-1} 'de oturması beklenilir. Daha sonra da yeterli istatistiği sağlayacak tur sayısı kadar tambur döndürülerek, hız salınımları ölçülür ve 100 min^{-1} 'de standart sapma bulunur. Motor kontrolü için kullanılan mikrokontrollör, standart sapmadan çok daha kolay şekilde hız hatalarının mutlak

değerlerin toplamını bulabildiği için standart sapma yerine, daha basit olan bu toplam, dengesiz yük göstergesi olarak pratikte kullanılmıştır.

Yük kestirimi için dengesiz yük ölçümünün gerekliliği şundandır ki, çamaşır yükünün kestirimi esas olarak dengeli ve dengesiz dağılmış çamaşır yükünün toplam eylemsizliğine dayanır. Eylemsizlik de dolaylı olarak zaman sabiti ölçümü ile, çıkış ve iniş zamanlarının ölçümüyle elde edilir. Dengesiz yükün varlığı, zaman sabitinin doğrudan eylemsizlikle olan ilişkisini değiştirir, yukarıda belirtilen (5.2) ve (5.3) denklemlerinin sağ tarafında yer alan dengesiz yükün oluşturduğu kuvvet ifadesi gözardı edilemez hale gelir. Dengesiz yükün varlığında sistemin dinamik davranışı değişir. Tambur kendi eksen etrafında dönerken, kazan sistemi de dengesiz yükün ve dönme hızının büyüklüğüne bağlı olarak ikinci bir eksen etrafında döner. Bu ilişki (5.2), (5.3), (5.4), (5.5) ve (5.6) numaralı denklemlerde verilmiştir ve şekil 5.8'de gösterilmiştir. Kazanın da bir eksen etrafında dönmesi, bağlı olduğu amortisör ve yay grubunda belli miktarda enerjinin sönmülmesine neden olur. Dengesizlik arttıkça sönmülen enerji de büyüyecektir. Sonuç olarak, aynı miktarlarda yüklerin dengesiz yük dağılımları farklı ise iniş-çıkış zamanlarında büyük farklılıklar gözlenebilir. Aşağıda şekil 5.9'da, 1 kg. yükün farklı deneylerde elde edilen hız hatası toplamları (dengesiz yüke karşılık gelir) ile iniş zamanı arasındaki ilişki gösterilmiştir.



Şekil 5.9 : Sabit yükte dengesiz yük miktarı ile iniş zamanı arasındaki ilişki

Dengesiz yük durumunun tespiti iki nedenle önemlidir. Birincisi, dengesiz yük olduğunda ve büyüklüğü bilinmediğinde, ölçülen iniş zamanı normalden daha az olacaktır. İkincisi ise, tambur 100 min⁻¹'den 400 min⁻¹'e hızlanırken çıkış zamanı

normalden daha uzun olacaktır. Bunların nedeni kazanın da bir eksen etrafında dönmesi ve amortisörlerde enerji kaybının olmasıdır. Yük kestirim yönteminde çamaşır yükünün kestirimi, çamaşır yükünün eylemsizlik-kütle ilişkisi kullanılarak yapıldığı için dengesiz yük bilgisi iniş-çıkış zamanlarının kompanzasyonu için gereklidir.

5.2.2 Çıkış zamanı

Tamburun 100 min^{-1} 'den 400 min^{-1} 'e ne kadar sürede çıktığı bilgisi kaydedilerek, çamaşırın eylemsizliğine ilişkin bir veri elde edilir. Eylemsizliği büyük olan çamaşır yükü daha uzun sürede 400 min^{-1} 'e çıkarken, eylemsizliği küçük olan çamaşır yükü daha kısa sürede bu hıza ulaşır. Bunu daha iyi açıklamak için şekil 5.4'te gösterildiği gibi ideal olarak tasarlanmış dengeli çamaşır yükünün dinamik hareket denklemleri ele alınabilir.

$$J \cdot \frac{d\Omega}{dt} + B \cdot \Omega + T_L = T_{\text{motor}} \quad (5.7)$$

$$\Omega(t) = \left(\Omega_i - \frac{T_{\text{motor}} - T_L}{B} \right) e^{-t/\tau} + \frac{T_{\text{motor}} - T_L}{B} \quad (5.8)$$

zaman sabiti, $\tau = \frac{J}{B}$ 'dir.

$K = \frac{T_{\text{motor}} - T_L}{B}$ yerine konursa,

$$\Delta t = -\tau \cdot \ln\left(\frac{\Omega_f - K}{\Omega_i - K}\right) \quad (5.9)$$

elde edilir.

Ω : tamburun açısal hızı

Ω_i : ilk hız

Ω_f : son hız

J: toplam eylemsizlik (tambur, çamaşır)

B: motor + tambur sisteminin toplam eşdeğer vizkoz sürtünme katsayısı

T_L : yük torku (coulomb sürtünmesi, statik sürtünme ve diğer etkiler)

T_{motor} : çamaşır makinesi motorunun sisteme verdiği ortalama mekanik tork

Δt : çıkış zamanı

Yukarıda açıklanan parametrelerden B , T_L ve T_{motor} 'un zamanla değişmediği varsayılırsa, denklem (5.9)'dan anlaşılacağı gibi Δt çıkış zamanının, zaman sabiti τ ile lineer bir ilişkisi vardır. Zaman sabiti de zamanla değişmediği varsayılan vizkoz sürtünme, B , ve çamaşır makinesine yerleştirilen çamaşır miktarıyla değişen eylemsizliğe, J , bağlıdır. Burada B sabit, J değişken olarak düşünülebilir. Bu durumda,

$$\tau \propto J \Rightarrow \Delta t \propto J \quad (5.10)$$

olur. Çıkış zamanı ile eylemsizlik arasında kurulan bu ilişkiye, denklem (5.7) ve (5.8)'den yola çıkılarak varılmıştır. Denklem (5.7) ve (5.8), yük dağılımının ideal olduğu, yani dengesiz yükün olmadığı ya da küçük olduğu, sistemin sabit bir B vizkoz sürtünme katsayısı olduğu ve vizkoz sürtünme torkunun hızla lineer değiştiği varsayımından hareketle oluşturulmuştur. Ancak pratikte yük dağılımında dengesizlik olabilir, sürtünmeler sıcaklığa ve hıza göre değişim gösterebilir.

Çıkış zamanı, giriş geriliminin büyüklüğü ile de ilişkisi vardır. Sistemin mekanik torku, tambura kayış ve kasnakla bağlı bir elektrik motoru tarafından sağlanmaktadır. Çamaşır makinelerinde kullanılan elektrik motorları çok çeşitli olabilir. Ancak çamaşır makinesinde en çok kullanılan motor üniversal motordur (UM/AC). Fırçasız doğru akım motoru, asenkron motor da kullanılan diğer motorlardır. UM/AC motorlu bir tasarım, daha ucuz olduğu için tercih sebebi olmaktadır. Bu tez çalışmasında da veri toplamak için ve deney yapmak için UM/AC motoru olan çamaşır makineleri kullanılmıştır.

UM/AC motorun hız ve tork denklemleri şu şekildedir:

$$T_e = k \cdot I^2 \quad (5.11)$$

$$E_m = k \cdot I \cdot \Omega \quad (5.12)$$

$$V_{in} = I \cdot (R + j \cdot \omega \cdot L) + k \cdot I \cdot \Omega \quad (5.13)$$

$$I = \frac{V}{R + k \cdot \Omega + j \cdot \omega \cdot L} \quad (5.14)$$

$$I_{rms}^2 = \frac{V_{rms}^2}{(R + k \cdot \Omega)^2 + (\omega \cdot L)^2} \quad (5.15)$$

$$T_e = k \cdot \frac{V_{rms}^2}{(R + k \cdot \Omega)^2 + (\omega \cdot L)^2} \quad (5.16)$$

Denklem (5.16)'dan motorun ürettiği ortalama elektriksel torkun giriş geriliminin karesiyle ilişkili olduğu ortaya çıkar. Elektriksel torkun, yaklaşık olarak motorun mekanik tork çıkışına eşit olduğu kabul edilebilir.

$$T_e \cong T_{\text{motor}} \quad (5.17)$$

Bu durumda denklem (5.7) yeniden şu şekilde yazılabilir:

$$J \cdot \frac{d\Omega}{dt} + B \cdot \Omega + T_L = T_e \quad (5.18)$$

dolayısıyla,

$$J \cdot \frac{d\Omega}{dt} + B \cdot \Omega + T_L = k \cdot \frac{V_{\text{rms}}^2}{(R + k \cdot \Omega)^2 + (\omega \cdot L)^2} \quad (5.19)$$

olur. Denklem (5.19)'dan, sistem dinamiğinin gerilimin karesiyle ilişkili olduğu görülür. Sistem, lineer olmayan bir sistemdir.

5.2.3 Sürtünme

Çamaşır makinesinin hareketli birimleri; motor, tambur ve bunları birbirine bağlayan kayış-kasnak düzeneğinden oluşmaktadır. Dengesiz yükün varlığında, bu hareketli sistemlere kazan ve kazanı taşıyan yay ve amortisör grubu da dahil edilebilir.

Tamburun mili ve milin oturduğu yatak ile motorun mili ve rotor milinin oturduğu yatak, sistemin başlıca sürtünme karakteristiğini belirleyen yapılardır. Bu kayıplar yatakların bilezikli, bilyalı veya rulmanlı olmasına göre değişiklik gösterir. Bölüm 5.2.2'de verilen basit modelden farklı olarak vizkoz sürtünme torku ile hız arasındaki ilişki genellikle lineer değildir [15]. Hava sürtünmesi ve vantilayon kayıpları, yük kestiriminin yapıldığı hızlarda ihmal edilebilir düzeydedir.

Dönen sistemin zaman sabiti, $\frac{J}{B}$, sürtünmeyle ters orantılıdır. Sürtünme sabitleri, makineden makineye ve ortam sıcaklığına bağlı olarak değişebilir. Belirlenecek yük kestirim algoritmasının performansı ortam şartlarından etkilenmemeli ve makineden makineye farklılık göstermemeli. Bu sebeplerle, dönen sistemin sürtünme sabitiyle ilgili bilgi elde edilmesi gerekir. Bu bilgiye dolaylı olarak ulaşılabilir.

Denklem (5.7)'de, $J \cdot \frac{d\Omega}{dt}$ terimi hız sabitken sıfır olur ve sabit hızda denklem,

$$B \cdot \Omega + T_L = T_{\text{motor}} \quad (5.20)$$

haline gelir. Denklem (5.19) ve (5.17)'den, sabit hızda motorun çektiği akımla veya akımın karesiyle sistemin sürtünmesinin ilişkili olduğu görülmektedir. Yük kestirimi için yapılan deneylerde kullanılan UMAC motorlu çamaşır makinelerinin motor kontrol kartlarında akım algılanmamaktadır. Ancak akım bilgisine dolaylı olarak ulaşmak mümkündür. UMAC motor kontrolü, triyakla AC gerilim kıyılarak yapılmaktadır. Kontrol yöntemi olarak kapalı çevrim PI hız kontrolü yapılmaktadır. Motor sabit hızda sürülürken çektiği akım ile triyağın tetikleme açısı arasındaki bağıntı şu şekildedir:

$$i(\omega t) = \frac{1}{\omega L} \int_{\alpha}^{\omega t} [v_{in}(\zeta) - E] d\zeta \quad (5.21)$$

ω şebeke frekansı, L motorun eşdeğer endüktansı, α tetikleme açısı, v_s şebeke gerilim, E motorun emk'sıdır. Elektrik motorunun eşdeğer seri direnci bu denklemde ihmal edilmiştir.

Sistemin sürtünmesi ile ilgili veri elde etmek için tambur 400 min^{-1} 'de sürülürken, her şebeke yarı periyodunda ateşlenen triyağın tetikleme açısı (alfa) belirlenmiş bir süre boyunca toplanır. Bu toplam, dengesiz yükün olmadığı durumda çamaşır makinesinin vizkoz sürtünme katsayısı ile orantılıdır.

Dengesiz yük varsa, enerji kaybı sadece tambur ve motor sürtünmelerinden kaynaklanmaz, kazan da hareket ettiği için damperlerde de enerji kaybı olur. Dolayısıyla dengesiz yük varken sistemin tork ihtiyacı artar, akım artar, elde edilen alfa toplamı artar. Dengesiz yükün varlığı, sürtünmeye ilişkin doğru verinin elde edilmesini zorlaştırır.

5.2.4 İniş zamanı

Bu tez çalışmasında, yük kestirimi için kullanılan verilerden biri de tambur 400 min^{-1} dönerken enerjinin kesilip tamburun 100 min^{-1} 'e serbest iniş süresidir. Bu veri, çıkış zamanı gibi sistemin zaman sabitiyle ilişkilidir. Sistemin basit modeli denklem (5.7)'yle aynıdır, tek fark $T_{motor} = 0$ olmasıdır. Denklem serbest iniş için tekrar şu şekilde yazılabilir:

$$J \cdot \frac{d\Omega}{dt} + B \cdot \Omega + T_L = 0 \quad (5.22)$$

İniş zamanı bilgisi çıkış zamanından farklı olarak şebeke geriliminden, motor parametrelerinden bağımsızdır. İniş zamanını etkileyen temel şeyler, dönen yapıların toplam eylemsizliği, sürtünmeler ve dengesiz yük miktarıdır.

5.2.5 Gerilim

Gerilim faktörünün yük kestirimindeki etkisi doğrudan değildir. Çıkış zamanı ölçümü ve 400 min^{-1} 'de motorun çektiği akımla ilişkili olan tetikleme açısının toplamı verileri, gerilimle ilişkili oldukları için, gerilim verisi diğer verilerin kompanzasyonu için kullanılabilir. Yani bu parametreler birbirinden bağımsız değildir. Dolayısıyla, yük kestirim algoritmasında gerilim kullanılması gereken önemli bir veridir.

6. ÇAMAŞIR MAKİNESİNDE ÇAMAŞIR YÜKÜ KESTİRİMİNİN GERÇEKLENMESİ

6.1 Giriş

Bu bölümde, çamaşır yükünün kestirimi için kullanılacak iki yöntemin uygulaması ele alınacaktır. Bu iki yöntem, bölüm 3 ve bölüm 4 'te ana hatlarıyla anlatılan YSA ile kestirim ve en küçük kareler regresyonu yöntemiyle kestirimdir. Şekil 5.2'de tarif edilen çamaşır yükünü belirleme profiliyle elde edilen veriler, önce lineer en küçük kareler yöntemiyle değerlendirilecektir. Daha sonra aynı veriler, çamaşır yükünü kestirmek için önerilen bir YSA modelinde ve öğrenme algoritmasında kullanılacaktır. Kullanılan yöntemlerin karşılaştırması yapılacaktır.

Çamaşır makinesinde yük kestirimi çalışmalarında kullanılan donanım şunlardır:

- Bilgisayar (PC): Intel® Pentium® 4 CPU 2.60 GHz, 1.00 GB RAM
- MPLAB® ICE 2000 In-Circuit Emulator
- Motor kontrol kartı
- 6 kg.lık çamaşır makinesi

Yapay sinir ağları ve lineer en küçük kareler kestirimi algoritmalarının bilgisayar ortamında benzetim çalışmaları, Matlab 6.5 programında m-file dosyaları oluşturularak, program yazılarak yapılmıştır.

Yük kestirim modelleri MATLAB ortamında hazırlanıp tatmin edici bir performans elde edildikten sonra, kestirim algoritmaları mikrodenetleyici tabanlı bir sistemde, motor kontrol kartının mikrodenetleyicisiyle gerçek zamanda gerçekleşmiştir. Mikrodenetleyici için yazılım geliştirme ve emülasyon ortamı olarak MPLAB IDE 7.2 ortamı, emülatör olarak MPLAB® ICE 2000 kullanılmıştır. Mikrodenetleyici yazılımı için C dili kullanılmıştır. Emülatör, motor kontrol kartı, ve çamaşır makinesi şekil 6.1 ve şekil 6.2'de görülebilir.

Bu çalışmada kullanılan çamaşır makinesinde yük kestirim verileri, 5 adet makineden elde edilmiştir. Makinelerin hepsi aynı model olup, 6 kg çamaşır kapasiteleri vardır. Veri toplama işlemi iki aşamada gerçekleştirilmiştir. İlk önce, sabit gerilimde 0 kg.'dan başlayarak 6 kg.'a kadar, her yük miktarı için deney 4 defa tekrarlanmıştır, 5 makine için veriler toplanmıştır. Buna ek olarak, 3 makineyle farklı

giriş gerilimleri için 1 kg, 3 kg ve 5 kg yüklerinde veriler toplanmış ve her gerilim değeri için deney 3 defa tekrarlanmıştır. Toplamda 415 deney yapılmış ve veri kümesi oluşturulmuştur. Sonuçta, farklı yüklerde ve farklı gerilimlerde 5 farklı makinede yük kestirimi yapabilmek için yeterli miktarda veri elde edilmiştir.



Şekil 6.1 : Motor kontrol kartı ve MPLAB® ICE 2000 In-Circuit Emulator



Şekil 6.2: Deney ortamının genel görünümü

6.2 Lineer En Küçük Kareler ile Yük Kestirimi

Bu yöntemde, çamaşır makinesinin mikrodenetleyicisiyle elde edilebilen tüm veriler parametre matrisinin bulunması için kullanılmıştır. Bölüm 5'te tanıtilan prediktörler (veriler), EKK yöntemiyle yük tahmin hatasını minimum yapacak katsayı vektörü β 'yı bulmak için kullanılmıştır. Prediktörleri tekrar hatırlamak gerekirse:

- 100 min⁻¹'de hız hatasının mutlak değer toplamı, $E_{hız} = \sum_{k=1}^K |E_k|$
- 100 min⁻¹'den 400 min⁻¹'e çıkış süresi, $T_R = \Delta t_r$
- 400 min⁻¹'de motora uygulanan tetikeleme açılarının toplamı, $\text{alfa} = \sum_{\ell=1}^L \alpha_{\ell}$
- 400 min⁻¹'den 100 min⁻¹'e iniş süresi, $T_F = \Delta t_f$
- motora uygulanan şebeke gerilimi, V_{in}

Denklem 3.2, kullanılan verilere göre,

$$\hat{Y} = \beta_0 + \beta_1 W + \beta_2 T_R + \beta_3 \text{alfa} + \beta_4 T_F + \beta_5 V_{in} \quad (6.1)$$

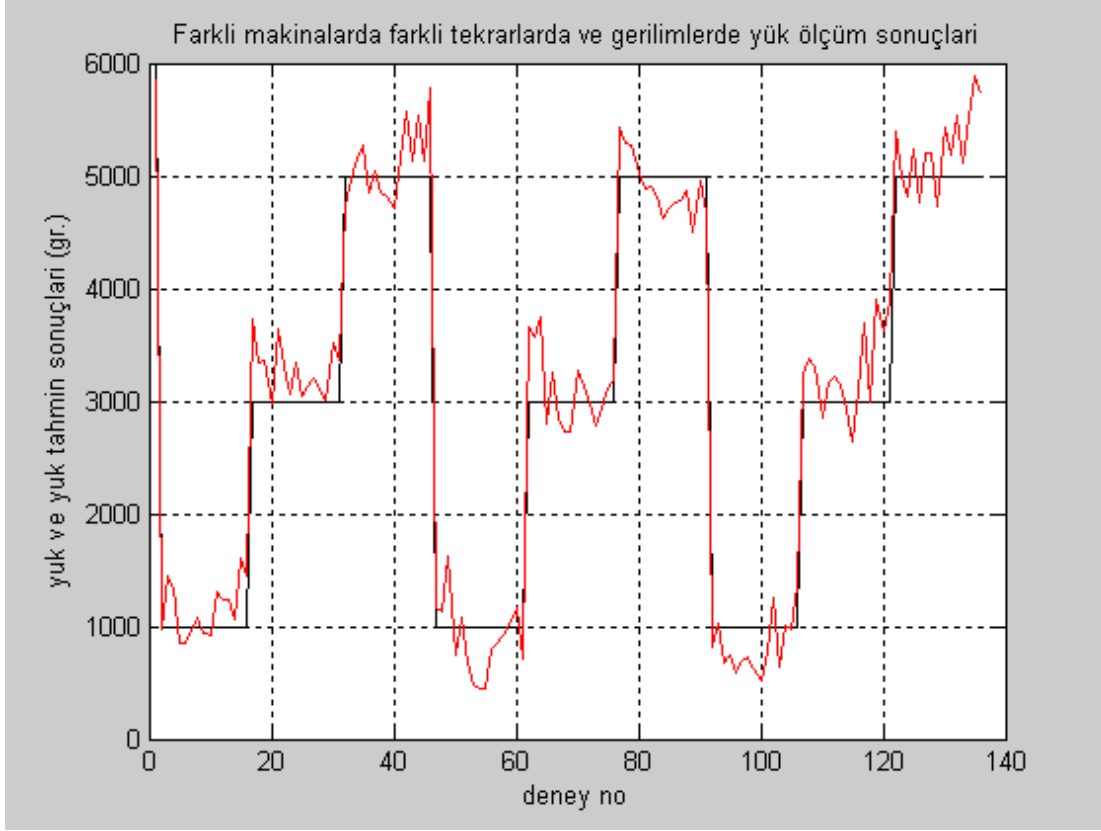
şeklinde yeniden yazılabilir, \hat{Y} kestirim sonucudur.

Toplanan verilerle parametre vektörünü bulmak için denklem (3.11a) kullanılır:

$$\hat{\beta} = (X^T X)^{-1} X^T y \quad (6.2)$$

Burada X, veri matrisidir. Bütün deneylerden elde edilen veriler, matris formuna getirildikten sonra denklem (6.2)'de yerine konulunca parametre vektörü $\hat{\beta}$ bulunmuştur.

Lineer en küçük kareler kestirimi algoritması MATLAB'ta m-file oluşturularak gerçekleştirilmiştir. Elde edilen sonuçlar, şekil 6.3 ve 6.4 ile çizelge 6.1 ve 6.2'de görülebilir. Veriler elde edilirken deneylerde kullanılan farklı gerilim değerleri sırasıyla şunlardır: 200 V, 210 V, 220 V, 230 V, 240 V. Farklı gerilim deneyleri 3 makinede, 1 kg., 3 kg. ve 5 kg.'da yapılmıştır. Her gerilim, her ağırlıkta 3 defa tekrar edilmiştir. Farklı gerilim deneylerinin kestirim sonuçları şekil 6.3 ve çizelge 6.1'de görülebilir.

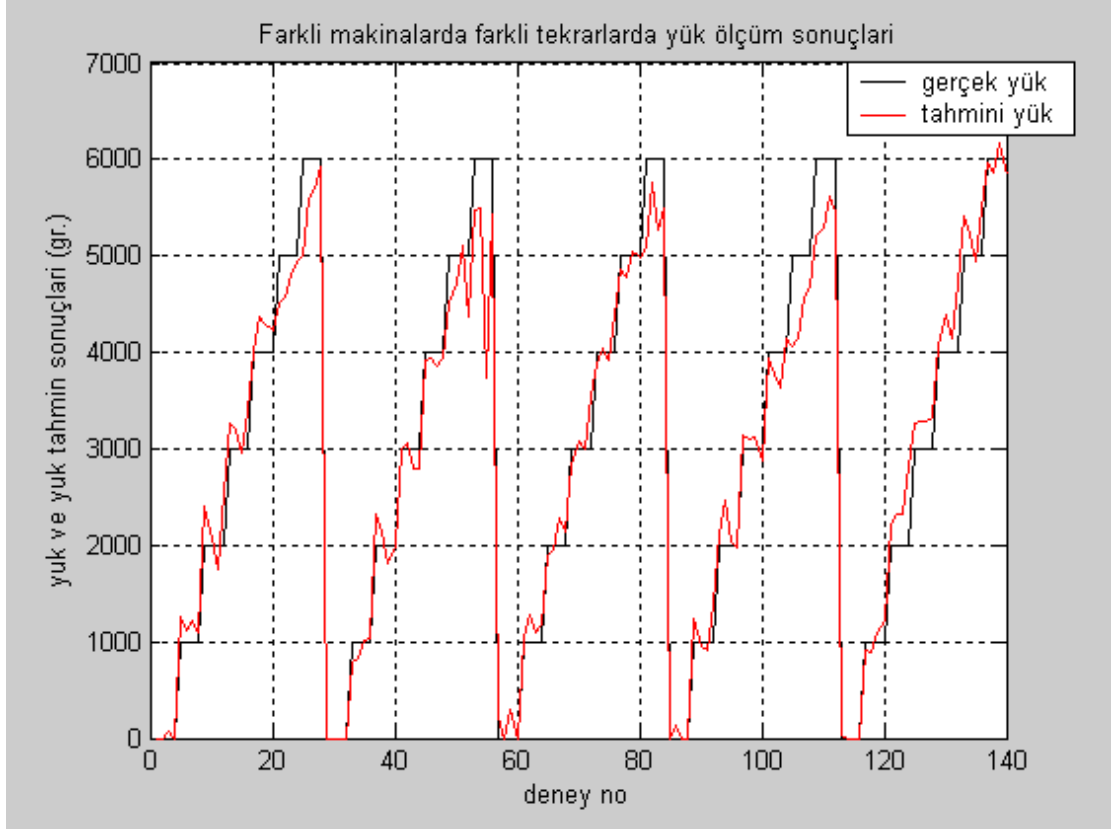


Şekil 6.3 : Değişen gerilimde EKK ile yük kestirim sonuçları; siyah eğri gerçek yük miktarını, kırmızı eğri ise EKK ile yük kestirimini göstermektedir.

Çizelge 6.1 : Değişen gerilimde kestirim ortalamaları ve standart sapmaları

Gerçek Çamaşır Yüğü	EKK Kestirim Ortalamaları	EKK Kestirim Standart Sapmaları
1000 gr.	945.63 gr.	301.30 gr.
3000 gr.	3209.39 gr.	317.14 gr.
5000 gr.	5092.42 gr.	337.85 gr.

Sabit gerilimde yapılan deneyler 5 makineyle 0 kg., 1 kg., 2kg., 3 kg., 4kg., 5 kg. ve 6 kg.'lık yüklerle yapılmıştır. Şekil 6.4'te ve çizelge 6.2'de sabit gerilimde(230V) yapılan deneylerin kestirimleri görülebilir.



Şekil 6.4 : Sabit gerilimde EKK ile yük kestirim sonuçları; siyah eğri gerçek yük miktarını, kırmızı eğri ise EKK ile yük kestirimini göstermektedir.

Çizelge 6.2 : Sabit gerilimde kestirim ortalamaları ve standart sapmaları

Gerçek Çamaşır Yükü	EKK Kestirim Ortalamaları	EKK Kestirim Standart Sapmaları
0 gr.	64.86 gr.	123.98 gr.
1000 gr.	1000.52 gr.	271.07 gr.
2000 gr.	2178.93 gr.	231.65 gr.
3000 gr.	3192.38 gr.	269.34 gr.
4000 gr.	4029.78 gr.	291.70 gr.
5000 gr.	4995.09 gr.	356.21 gr.
6000 gr.	5501.29 gr.	433.82 gr.

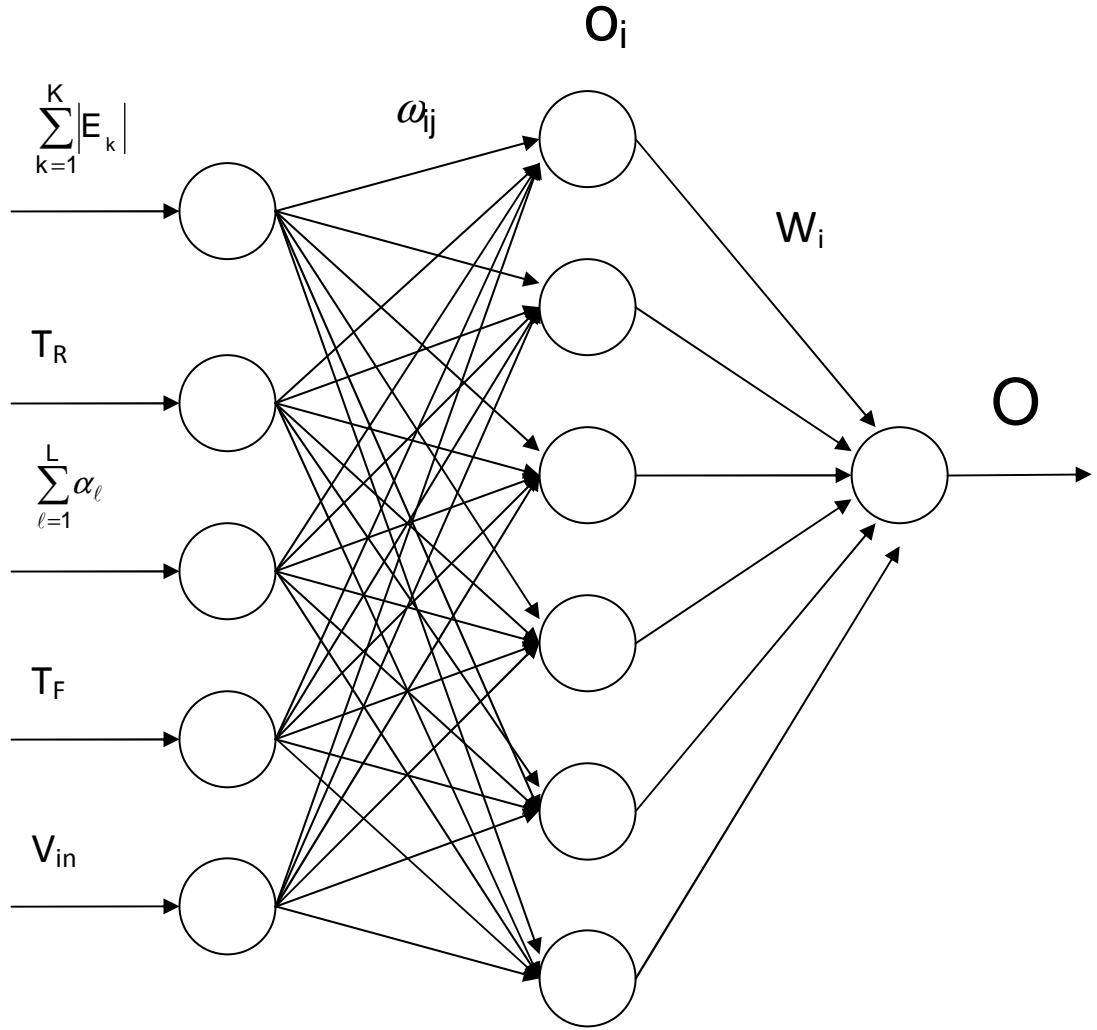
Lineer EKK ile bulunan $\hat{\beta}$ parametre vektörü, uygulamada herhangi bir zorlukla karşılaşmadan kullanılabilir. Yapılması gereken, çamaşır makinesindeki mikrodenetleyicinin program hafızasına denklem (6.1)'yi yazmaktır. Yük kestirimi yapmak için mikrodenetleyicide yapılacak işlemler sadece 5 adet toplama ve 5 adet çarpmadır.

6.3 İleri Beslemeli Geri Yayılım Ağı ile Yük Kestirimi

Lineer en küçük kareler kestirimi ile çamaşır yükü tahmini için kullanılan veriler, MATLAB'ta oluşturulan YSA algoritması için de kullanılmıştır. Deneylelerden elde edilen verilerin 1/3'ü, oluşturulan ileri beslemeli geri yayılım ağını eğitmek için kullanılmıştır. Eğitim için, 45 değişen gerilimli ve 93 sabit gerilimli olmak üzere toplam 138 deney verisi giriş vektörü olarak kullanılmıştır. Elde edilen ağın başarı ölçütü için test kümesi olarak eğitimde kullanılmayan veriler kullanılmıştır.

YSA ile çamaşır yükü tahminini gerçekleştirmek için bir tane gizli katmanı olan ileri beslemeli perseptron ağı kullanılmıştır. Ağın giriş katmanında 5 tane nöron, gizli katmanda 6 tane nöron ve çıkış katmanında kestirim sonucunu veren 1 tane nöron vardır. Öğrenme algoritması olarak geri yayılım algoritması ve delta öğrenme kuralı kullanılmıştır. Bölüm 4.8.2'de anlatılan 8 adımlık yöntem ile eğitim için ayrılmış deney verileriyle çıkış örüntüleri YSA'ya öğretilmiştir.

İleri beslemeli çok katmanlı perseptron ağının tercih edilemesinin nedeni, giriş vektörü ile çıkış vektörü çiftinin elimizde olmasıdır. Yani, giriş-çıkış veri kümesinin olduğu bir yapıyı, eğitici bir algoritmayla YSA'na çıkış örüntülerini öğretmek daha uygundur. Çok katmanlı perseptron ağı ise, yapısının ve öğrenme algoritmasının basitliği, hızlı öğrenme özellikleri nedeniyle tercih edilmiştir. Bu özellikleriyle, çok katmanlı perseptron ağı uygulamada da yaygın olarak kullanılan bir YSA'dır. Bu tez çalışmasında çamaşır yükü kestirimi için kullanılan YSA, sembolik olarak şekil 6.5'te gösterilmiştir.



Şekil 6.5 : Çamaşır yükü kestiriminde kullanılan 5x6x1'lik çok katmanlı perseptron ağı

Delta öğrenme algoritmasında, gradyan azalan, $-\eta \frac{\partial E_k}{\partial w_{ij}}$, terimine momentum terimi de eklenmiştir. Momentumlu öğrenmede, hatanın daha hızlı istenen değerin altına indiği gözlemlenmiştir.

6.3.1 Verilerin hazırlanması

Veri hazırlanmasında son aşamada; YSA'nın girişleri için verinin normalizasyonu gerekmektedir. İleri beslemeli geri yayılım ağında 0 ile 1 aralığında veri tercih edilir, çünkü kullanılan sigmoid fonksiyonunun çıkış aralığı (0-1) sınıridir. Sigmoid fonksiyonlarının hemen doymaya girmemesi için normalizasyon şarttır, eğitimin hızı için önemlidir. Çizelge 6.3'te örneği görülen girişler normalize edilerek, YSA'nın girişine uygulanmaya hazır hale getirilmelidir. Çizelge 6.3'teki veriler mikrodenetleyicide kayıtlı oldukları halleridir. Yani mikrodenetleyicinin A/D portundan

okunan değerler ile zamanlayıcı (timer) biriminden elde edilen değerlerdir. Örneğin, şebekede ölçülen gerçek gerilimin rms değeri 220 V ise mikrodenetleyici bu değeri ölçeklendirerek 185 olarak kaydetmektedir. Amaç YSA'yı mikrodenetleyicide yazılımla gerçekleştirmek olduğu için, doğrudan bu veriler eğitimde kullanılmıştır. Bu verilerin normalizasyonu şu şekilde gerçekleştirilmiştir [16]:

$$r = \frac{A_{\max} - A_{\min}}{X_{\max} - X_{\min}} \quad (6.3)$$

olmak üzere;

$$A = r(X - X_{\min}) + A_{\min} \quad (6.4)$$

Yukarıdaki formüllerde;

X_{\max} : Gerçek değerlerin en büyük olanı,

X_{\min} : Gerçek değerlerin en küçük olanı,

A_{\max} : En büyük normalizasyon değeri,

A_{\min} : En küçük normalizasyon değeri,

X : Normalize edilecek değer,

A : Normalize edilmiş değerdir.

Normalizasyon sınırları uygulamada maksimum için 0,9, minimum için 0,1 seçilmiştir.

Çizelge 6.3 : Yük kestirim profili sonucunda bir makineden elde edilen veri örnekleri

Çamaşır Makinesi 1	$E_{\text{hız}} = \sum_{k=1}^K E_k $	T_R	$\sum_{\ell=1}^L \alpha_{\ell} = \text{alfa}$	T_F	V_{in}
1 kg.	910	280	5058	458	168
3 kg	1538	223	4681	538	187
5 kg	1090	216	4432	641	195

6.3.2 YSA algoritmasının bilgisayar ile benzetimi

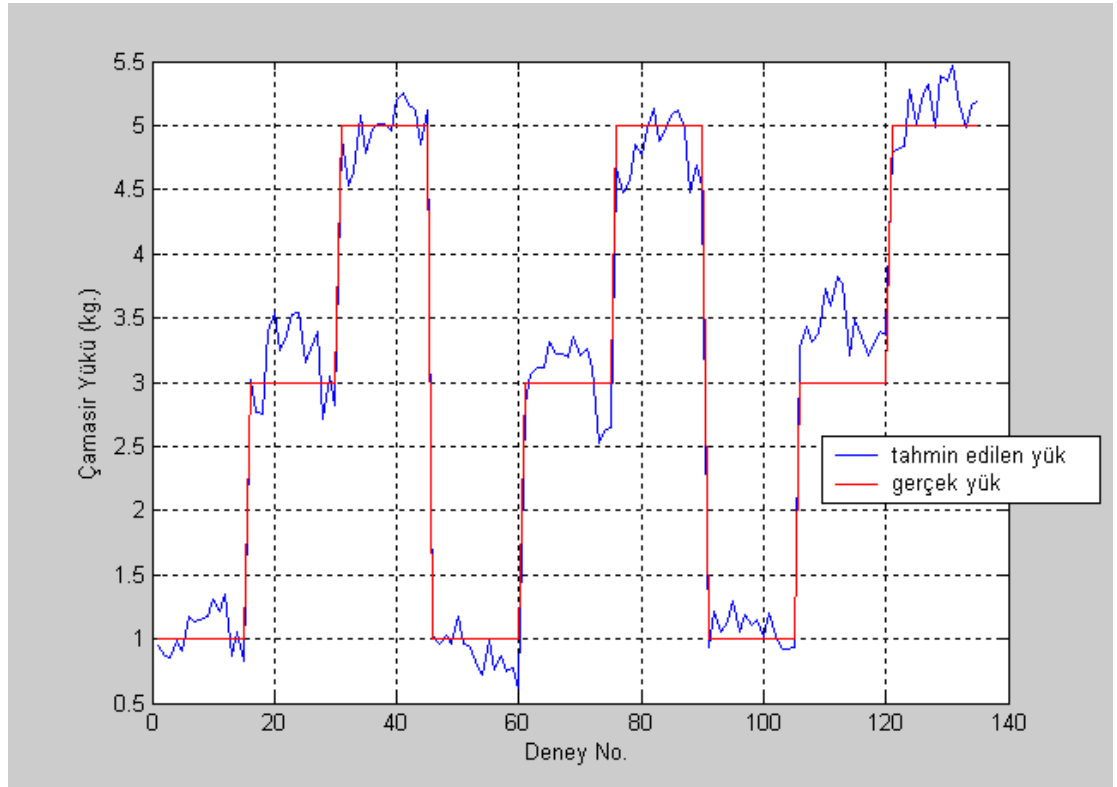
Bölüm 4.8.2'de anlatılan 8 adımlık algoritma 20.000 periyot (epoch) uygulandıktan sonra arzu edilen sonuçlar elde edilmiştir. Ara katman ve çıkış katmanı ağırlıklarının istenen sonuçları verecek şekilde değişimleri sağlanmıştır. Aktivasyon fonksiyonu olarak unipolar sigmoid fonksiyonu kullanılmıştır. Prediktörlerle çıkış arasındaki ilişki

lineer olmadığı için ve kullanılan öğrenme algoritmasında türev alınabilir bir fonksiyon gerektiği için, aktivasyon fonksiyonu sigmoid seçilmiştir. MATLAB'ta gerçekleştirilen YSA ile elde edilen sonuçlar şekil 6.6 ve 6.7 ile çizelge 6.4 ve 6.5'te verilmiştir. Öğrenme algoritması, MATLAB m-file olarak EK A1'de yer almaktadır. Eğitimde kullanılan temel öğrenme parametreler şunlardır:

İterasyon sayısı: 20000 döngü

Öğrenme terimi η : gizli katman için 0.03, çıkış katmanı için 0.05

Momentum terimi α : 0.1



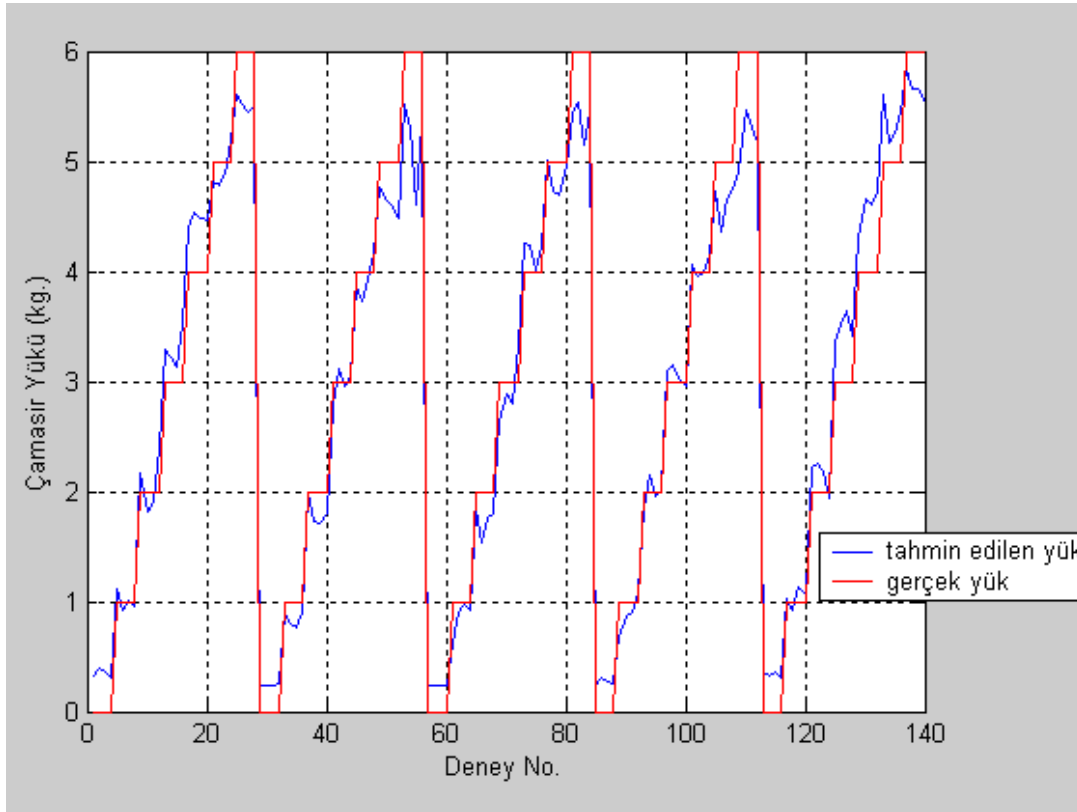
Şekil 6.6 : Değişen gerilimde YSA ile yük tahmin sonuçları; kırmızı eğri gerçek yük miktarını, mavi eğri ise YSA ile yük kestirimini göstermektedir.

Kullanılan YSA algoritmasının, değişen gerilim deneylerinden elde edilen giriş-çıkış örüntülerini tanımada başarılı olduğu görülebilir. Çizelge 6.4'te değişen gerilimde yükü tahmin etme başarısına ilişkin ortalama ve standart sapma verileri vardır.

Çizelge 6.4 : Değişen gerilimde kestirim ortalamaları ve standart sapmaları

Gerçek Çamaşır Yüğü	YSA Kestirim Ortalamaları	YSA Kestirim Standart Sapmaları
1000 gr.	1006.78 gr.	168.02 gr.
3000 gr.	3220.63 gr.	301.61 gr.
5000 gr.	4971.42 gr.	250.19 gr.

Sabit gerilimde 5 makinede elde edilen yük kestirim sonuçları, çizelge 6.5 ile şekil 6.7'de verilmiştir.



Şekil 6.7 : Sabit gerilimde YSA ile yük tahmin sonuçları; kırmızı eğri gerçek yük miktarını, mavi eğri ise YSA ile yük kestirimini göstermektedir.

Çizelge 6.5 : Sabit gerilimde kestirim ortalamaları ve standart sapmaları

Gerçek Çamaşır Yüğü	YSA Kestirim Ortalamaları	YSA Tahmin Standart Sapmaları
1000 gr.	974.66 gr.	150.47 gr.
2000 gr.	1970.00 gr.	224.23 gr.
3000 gr.	3186.75 gr.	284.62 gr.
4000 gr.	4172.88 gr.	312.70 gr.
5000 gr.	4952.823 gr.	263.51 gr.
6000 gr.	5364.47 gr.	278.76 gr.

YSA algoritmasının başarılı şekilde genelleme yapması için, eğitim periyodu çok uzun tutulmamalıdır, ya da eğitimi durduracak limit hata değeri çok küçük seçilmemelidir ve gizli katmandaki nöron sayısı çok fazla olmamalıdır. Öğrenme periyodunun ne uzunlukta olması gerektiği ya da hata limitinin ne olması gerektiği hakkında kesin yargılarla konuşmak güçtür. Ancak, iyi sonuç veren bir YSA elde etmek için dikkat edilmesi gereken bu sınırlayıcı öğeler, eğitim verisinin miktarına ve yapısına göre şekillenir. YSA'nın yapısına, eğitim periyoduna, hata limiti gibi öğelerin ne olacağına birçok deneme-yanılma aşamasından sonra karar verilir.

Eğitim periyodunun çok uzun olması ya da eğitimi durduran limit hata değerinin çok küçük seçilmesi halinde, YSA iyi genelleme yapamaz. Eğitim sonucunda, eğitim kümesinde çok küçük bir hata elde edilmiş olabilir; fakat elde edilen YSA, test kümesine uygulandığında öğrenmenin istenen düzeyin altında kaldığı görülebilir. Bunun nedeni, uzun eğitimin sonucunda YSA, giriş-çıkış örüntüsünü öğrenmek yerine ezberlemeye başlamasıdır. Ölçümlerde bulunan gürültü ve hataları, yani öğrenmemesi gereken şeyleri de öğrenmeye başlar. Bundan dolayı eğitim kümesini çok iyi öğrenmiş olabilir ancak eğitim kümesinin dışında kalan örüntüleri genellemede başarısız olur.

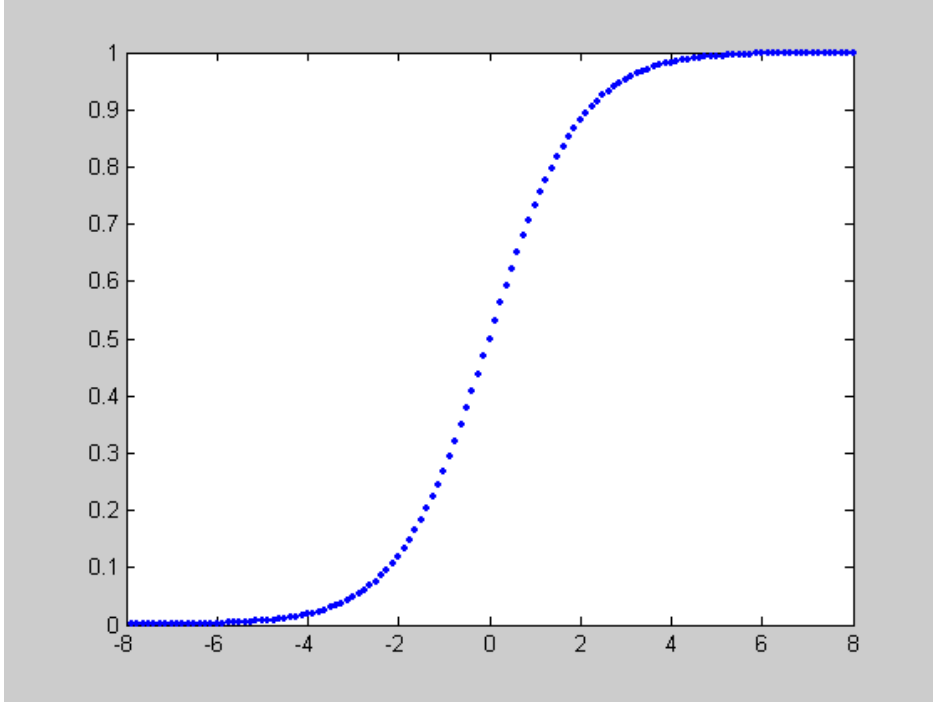
6.3.3 YSA'nın mikrodenetleyici ile gerçekleştirilmesi

EKK ile kestirim algoritmasının mikrodenetleyiciyle uygulanması basittir. Yapılması gereken, MATLAB'ta β parametre vektörü bulunduktan sonra denklem (6.1)'de yerine konarak, mikrodenetleyicinin çamaşır makinesi yazılımına denklemi eklemektir.

YSA'nın mikrodenetleyiciyle uygulaması ise EKK kadar kolay değildir. Bunun nedenleri, lineer olmayan aktivasyon fonksiyonlarının kullanılması ve çok sayıda sinaptik bağlantı nedeniyle çok fazla işlem içermesidir. Yük kestirimi için 36 tane sinaptik bağlantı vardır ve her işlemci biriminin (nöron) aktivasyon fonksiyonu sigmoiddır.

Mikrodenetleyicide uygulanan YSA'nın öğrenmesi durmuştur. Öğrenme MATLAB'ta tamamlanıp ağırlıklar öğrenildikten sonra eğitim durdurulur. Eğitimi tamamlanmış YSA mikrodenetleyiciye yazılımla gömülerek uygulamada kullanılabilir, fakat YSA'nın uygulamada kullanılabilmesi için yukarıda bahsedilen pratik zorlukların aşılması gerekir.

Lineer olmayan aktivasyon fonksiyonu tablo oluşturularak mikrodenetleyicide kullanılabilir. Bunun için 128 noktalık (byte) bir tablo yeterli olmuştur. Sigmoidal aktivasyon fonksiyonu tablosunun grafik gösterimi şekil 6.8'de yer almaktadır.



Şekil 6.8 : Sigmoid fonksiyonu tablosu

128 noktalık bir tablo yerine 64 noktalık tablo da kullanılabilir, fakat çözünürlük daha düşük olacağı için tahmin sonuçları daha kötü olacaktır. EK A2'de MATLAB'ta elde edilen YSA'nın PIC 16C76 mikrodenetleyicisine C yazılım diliyle gerçekleştirilmesi verilmiştir. 128 noktalık tablolulu algoritma önce MATLAB'ta çalıştırılmış ve aşağıda çizelge 6.6 ve çizelge 6.7'deki sonuçlar ortaya çıkmıştır.

Çizelge 6.6 : Sigmoid fonksiyonu için 128 noktalık tablo kullanılan YSA ile sabit gerilimde tahmin ortalamaları ve standart sapmaları

Gerçek Çamaşır Yüğü	YSA Kestirim Ortalamaları	YSA Kestirim Standart Sapmaları
1000 gr.	972.2 gr.	161.0 gr.
2000 gr.	1965.1 gr.	272.0 gr.
3000 gr.	3172.0 gr.	317.7 gr.
4000 gr.	4185.2 gr.	311.7 gr.
5000 gr.	4942.5 gr.	289.0 gr.
6000 gr.	5345.2 gr.	290.3 gr.

Çizelge 6.7 : Sigmoid fonksiyonu için 128 noktalık tablo kullanılan YSA ile değişen gerilimde tahmin ortalamaları ve standart sapmaları

Gerçek Çamaşır Yüğü	YSA Kestirim Ortalamaları	YSA Kestirim Standart Sapmaları
1000 gr.	1006.8 gr.	168.0 gr.
3000 gr.	3220.6 gr.	301.6 gr.
5000 gr.	4971.4 gr.	250.2 gr.

Mikrodenetleyici algoritması MATLAB'ta uygulanarak elde edilmiştir.

Çizelgelerdeki sonuçlar, 128 noktalık sigmoid tablosu kullanılarak mikrodenetleyici ile gerçekleşmesinin tatmin edici olduğunu gösteriyor.

Yazılım, mikrodenetleyiciye aktarılmış ve çalıştırılmıştır. Algoritmanın MATLAB'takine yakın bir performansla çalıştığı görülmüştür. 16 MHZ'lik kristalle çalışan PIC 16C76, verileri aldıktan 4,7 ms sonra kestirim sonucunu bildirmektedir. Mikrodenetleyicilerin çalışma hızları düşünüldüğünde bu uzun bir süredir. Ancak, mikrodenetleyiciler işlem ağırlıklı çalışan yapılar olmadıkları için çarpma-toplama işlemleri uzun sürmektedir. Ayrıca algoritma tablo içerdiği için ve karmaşık olduğu için program hafızasında 1Kbyte'a yakın yer kaplamaktadır.

6.4 Sonuçlar ve Karşılaştırmalar

Bu bölümde, çamaşır makinesinde yıkama başlamadan önce çamaşır miktarını belirlemek amacıyla, momentumlu genelleştirilmiş delta öğrenme kuralını kullanan, çok katmanlı perseptron ağı ile lineer en küçük kareler regresyonu uygulamaları anlatılmıştır. Bilindiği gibi lineer olmayan sistemleri modellemede, giriş-çıkış ilişkisini tahmin etmede YSA, lineer olan yöntemlerden daha başarılıdır.

Sabit gerilimde YSA ve EKK ile elde edilen kestirim ortalamaları çizelge (6.5) ve çizelge (6.2) ile verilmiştir. Bu çizelgelerden, iki yöntemle elde edilen tahmin ortalamalarının ve standart sapmaların birbirine yakın oldukları görülmektedir, ancak çamaşır yükü arttıkça EKK ile yapılan kestirimin standart sapmalarında artma gözlenmektedir. Buradan eğer şebeke gerilimi 220 V'tan çok sapma göstermiyorsa iki yöntemle elde edilen sonuçların birbirine yakın olduğu, yük 6 kg.'a yaklaştıkça YSA ile kestirimde hatanın biraz artmakta olduğu ancak standart sapmanın EKK yöntemiyle karşılaştırıldığında daha düşük kaldığı gözlenmektedir.

YSA ile EKK arasındaki fark değişen gerilimde daha net ortaya çıkmaktadır. İki yöntemle değişen gerilimde elde edilen sonuçlar, yük tahmin ortalamaları ve standart sapmaları çizelge (6.8) ve çizelge (6.9)'da verilmiştir.

Çizelge 6.8 : Değişen gerilimde YSA ve EKK ile kestirim ortalamaları

Gerçek Çamaşır Yüğü	YSA Kestirim Ortalamaları	EKK Kestirim Ortalamaları
1000 gr.	1006.78 gr.	945.63 gr.
3000 gr.	3220.63 gr.	3209.39 gr.
5000 gr.	4971.42 gr.	5092.42 gr.

Çizelge 6.9 : Değişen gerilimde YSA ve EKK ile kestirim ortalamalarının standart sapmaları

Gerçek Çamaşır Yüğü	YSA Kestirim Standart Sapmaları	EKK Kestirim Standart Sapmaları
1000 gr.	168.02 gr.	301.30 gr.
3000 gr.	301.61 gr.	317.14 gr.
5000 gr.	250.19 gr.	337.85 gr.

Çizelge (6.8)'de farklı gerilimlerde YSA ile kestirim ve EKK ile kestirim ortalamaları birbirine yakın görünmektedir. Ancak aralarındaki fark, çizelge (6.9)'da gösterilen standart sapmalara bakıldığında anlaşılmaktadır. Değişen gerilimde, çok katmanlı perseptron ağı ile elde edilen kestirimin daha başarılı olduğu söylenebilir, çünkü bu yöntemde EKK ile karşılaştırıldığında standart sapma üç yük tipinde de daha düşük çıkmıştır. Standart sapma bir kestirim algoritmasının başarısını değerlendirmek için çok önemli bir kriterdir. Standart sapma, yapılan tahminin ortalamadan ne kadar sapabileceğini gösterdiği için önemlidir. Standart sapmanın yüksek olması, ortalama kestirimden sapma ihtimalinin yüksek olacağını da belirtisidir. Bu nedenle, kestirim başarı kriteri olarak tahmin ortalamalarının yanında tahmin standart sapmaları da göz önünde bulundurulmalıdır.

EKK ile kestirim fazla işlem içermediği için ve lineer olduğu için mikrodenetleyiciyle uygulaması kolaydır. ANN ise lineer olmamasından, çok sayıda işlem içermesinden ve daha karmaşık bir algoritmaya sahip olmasından dolayı uygulaması daha zordur.

İki yöntemde de kütle ile eylemsizlik arasındaki ilişkinin lineer olmayan karakteristiğinin belirginleştiği 6 kg. yükü diğer yüklere oranla daha büyük hatalarla tahmin edilmiştir.

7. SONUÇ VE ÖNERİLER

Bu çalışmada, aynı model 5 adet çamaşır makinesine ait hız salınımı, çıkış zamanı, sabit hızda uygulanan tetikleme açıları toplamı, iniş zamanı ve giriş gerilimi verileri kullanılarak, geriye yayılım algoritması ile eğitilmiş yapay sinir ağı ve en küçük kareler regresyonu ile yük kestirimi yapılmıştır. Bu iki yöntem karşılaştırılmış ve uygulanabilirlikleri tartışılmıştır.

Tezin giriş bölümünde, tezin amacından ve çamaşır makinesinde çamaşır yükünün belirlenmesinin değerinden bahsedilmiştir. İkinci bölümde çamaşır makinesinde yük tahmini ile ilgili yapılmış çalışmalardan bahsedilmiştir. Sensörlü ve sensörsüz yük kestirim algoritmaları tanıtılmıştır. Üçüncü bölümde yük kestirimi için bu tezde kullanılan yöntemlerden biri olan lineer regresyon analizi tanıtılmıştır. Dördüncü bölümde tezin ana amaçlarından biri olan lineer olmayan bir yöntemle regresyon analiz yönteminin, yapay sinir ağlarının, teorik alt yapısı sunulmuştur. Beşinci bölümde çamaşır yükünün kestirimi için kullanılan veriler ve verileri elde etmek için önerilen bir yük kestirim profili tanıtılmıştır. Altıncı bölümde, önceki bölümlerde tanıtılan lineer ve lineer olmayan kestirim algoritmalarının hem benzetim ortamında (MATLAB) hem de çamaşır makinesindeki uygulamaları ve sonuçları tartışılmıştır.

Yapay sinir ağları gibi öğrenme yöntemleri örneklerden öğrenmeye dayanmaktadır. Örneklerden öğrenmenin temel felsefesi bir olay hakkındaki gerçekleşmiş örnekleri kullanarak olayın girdi ve çıktıları arasındaki ilişkileri öğrenmek ve bu öğrenilen ilişkilere dayanarak yeni karşılaşılan örneklerin mümkün olduğu kadar doğru bir şekilde çıktılarını belirlemektir. Öğrenmenin arzu edildiği şekilde başarılması için ağın eğitimi için kullanılacak örnek kümesi çok çeşitli olmalıdır. Bu tez çalışmasında kullanılan eğitim ve test verileri, aynı model 5 farklı makineden oluşmuştur ve verilerin bir kısmı farklı gerilim ortamında elde edilmiştir.

Bu tezde bahsi geçen YSA ile çamaşır yükü kestirimi algoritmasının nihai olarak bir çamaşır makinesi ürününde gerçekleştirilebilir olması için çok daha fazla sayıda makine ile çalışma yapılmalı ve algoritma, gerçek veriler ile belirli bir süre sınanmalıdır.

Çok katmanlı perseptron ağının geri yayılım algoritmasıyla elde edilen kestirim sonuçlarının, lineer en küçük kareler yöntemiyle elde edilenlere kıyasla, özellikle

gerilimin deđiřtiđi durumlarda, biraz daha iyi olduđu gözlenmiřtir. Her iki yöntemde de 6 kg. yük tahmini yaklaşık 0,5 kg. az yapılmıřtır. Bunun nedeni, 5. bölümde gösterildiđi gibi çamařır miktarı arttıka, eylemsizliđin belirli bir ađırlıktan sonra doyma eğilimine girmesidir.

Çamařırın ađırlık-eylemsizlik iliřkisi, dengesiz yükün sistem dinamiđine kattıđı lineer olmayan iliřki ve gerilim deđiřiminin diđer prediktörlerle lineer olmayan iliřkisi düşünöldüđünde çamařır yükünü tahmin etmek için kullanılacak yöntemin de lineer olmayan sistemleri modelleyebilecek, genelleyebilecek bir yöntem olması gerekir. Lineer en küçük kareler yöntemi ile lineer sistemlerde tatmin edici kestirim sonuçları elde etmek mümkündür; fakat YSA ile lineer olmayan sistemlerde, sistemin modellenmesinin çok güç olduđu durumlarda iyi sonuçlar elde edilebilir. Yük tahmin ortalamalarının yanı sıra, tahmin ortalamalarının standart sapmaları da kestirim algoritmalarının başarı ölçütü olarak deđerlendirildiđinde, bu çalışmada YSA ile çamařır yükü kestirimi, EKK ile çamařır yükü kestiriminden biraz daha başarılı olmuřtur.

Yük kestirim algoritmasının uygulamada kullanılmasını etkileyen faktörlerden biri algoritmanın başarısıysa diđer algoritmanın gerçeklenebilirliđidir. Gerçeklenebilirlik kriteri göz önüne alındıđında, en küçük kareler yöntemi basitliđi sebebiyle YSA'ya göre daha avantajlıdır. Uygulanacak almortmaya karar verilirken kestirim başarısıyla uygulanabilirlik birlikte düşünölmelidir.

Çamařır yükünün çamařır makinesi tarafından algılanması-kestirilmesi, gün geçtikçe önemi artan bir konudur. Çamařır yükünün belirlenmesi enerji veriminin, su veriminin, hatta yıkama performansının artışına katkı sađlayacađı için önemlidir.

KAYNAKLAR

- [1] Sümer, İ. T., 1996. Yıkama hızı ve hızlanma süresinin çamaşır makinesi yıkama performansı üzerindeki mekanik etkileri, *Arçelik ARGE Raporu*, İstanbul.
- [2] Yang, S. B., Yeul, K., Hong, C. H., 2003. Apparatus and method for detecting laundry weight in washing machine employing sensorless BLDC motor, *US Patent*, No: US 6,842,928 B2
- [3] Byung, K. C., Dong, W. K., Dong, S. L., 2006. Drum type washing machine and method of detecting laundry weight thereof, *US Patent*, No: US 2006/0179584A1
- [4] Yong, H. K., Lyu, C., 2006. Method for detecting dewatering load in washing machine and washing machine control method using the same, *US Patent*, No: US2004/0194226A1
- [5] Beier, D., Koschnike, J., Krause, D., 2005. Method of determining the weight of laundry in a washing machine drum, *US Patent*, No: US2005022315
- [6] Faraway, J., Julian., 2002. Practical Regression and Anova using R, <http://cran.r-project.org/doc/contrib/Faraway-PRA.pdf>, alındığı tarih: 01.12.2006.
- [7] Fuller, R., 1995. Neural Fuzzy Systems, Lecture Notes, Abo Akademi University. Retrieved November 15, 2009, from <http://users.abo.fi/rfuller/robert.html>
- [8] Uri-1 <<http://www.sinancanan.net/bilimsel/mss.htm>>, alındığı tarih 11.11.2006.
- [9] Uri-2 <<http://batitrakya.dostweb.com/yapaysiniraglari/ysa.htm>>, alındığı tarih 13.11.2006.
- [10] Haykin, S., 1999. *Neural Networks: A Comprehensive Foundation*, Second Edition, Prentice Hall, New Jersey.
- [11] Uri-3 <<http://www.backpropagation.net>>, alındığı tarih 14.11.2006.
- [12] Bolat, S., 2003. Yapay Sinir Ağı ile Elektrot ve İzolatör Biçim Optimizasyonu, *Yüksek Lisans Tezi*, İ.T.Ü., F.B.E., İstanbul
- [13] Şengör, N. S., 2001. *Yapay Sinir Ağlarına Giriş*, Ders Notları, İTÜ, İstanbul.
- [14] Erenay, H. K., Yıldırım, Y., 2001. Çamaşır Makinesinde Sıkma İşleminde Dengeli ve Dengesiz Yük Ölçümü, *Arçelik Raporu*, İstanbul.

- [15] **Yılmaz, M.**, 1999. Üniversal Motorun Benzetişim ve Tasarımı, *Yüksek Lisans Tezi*, İ.T.Ü., F.B.E., İstanbul.
- [16] **Ceylan, G.**, 2004. Yapay Sinir Ağları ile Kısa Dönem Yük Tahmini, *Yüksek Lisans Tezi*, İ.T.Ü., F.B.E., İstanbul.

EKLER

EK A1

ÇAMAŞIR MAKİNESİNDE YÜK KESTİRİMİ İÇİN KULLANILAN YSA ALGORİTMASININ MATLAB PROGRAMI

```
clear all;
load deneyler.txt
%%%%NORMALİZASYON KATSAYISI%%%%%%%%
K1=0.8/(max(deneyler(:,6))-min(deneyler(:,6)));
K2=0.8/(max(deneyler(:,2))-min(deneyler(:,2)));
K3=0.8/(max(deneyler(:,7))-min(deneyler(:,7)));
K4=0.8/(max(deneyler(:,5))-min(deneyler(:,5)));
K5=0.8/(max(deneyler(:,3))-min(deneyler(:,3)));
%%%%NORMALİZASYON KATSAYISI_SON%%%%%%%%
row=1;
column=1;
for row=1:415
    kalan=mod(row,3); % Deneylerin 1/3'ü eğitim için ayrılıyor
    % kalan2=mod(row,30);
    if(kalan==1)% | kalan==2)% & kalan2>15)
        trainset(1,column)=K1*(deneyler(row,6)-min(deneyler(:,6)))+0.1;
        trainset(2,column)=K2*(deneyler(row,2)-min(deneyler(:,2)))+0.1;
        trainset(3,column)=K3*(deneyler(row,7)-min(deneyler(:,7)))+0.1;
        trainset(4,column)=K4*(deneyler(row,5)-min(deneyler(:,5)))+0.1;
        trainset(5,column)=K5*(deneyler(row,3)-min(deneyler(:,3)))+0.1;

        y(column)=deneyler(row,1)*0.15;
        column=column+1;
    end
end

nu=0.05;
Wout=[ 1
        1
        -1
        -1
        0
        2
        1
        %-2
        1

]; %5x6
Wout2=0;
Wout3=0;
Whidden=[ -1 2 3 -1 1
```

```

2 3 1 -1 -2
2 2 1 1 0
0 0 1 0 -2
-1 -2 -1 1 1
1 3 3 2 2
1 4 1 0 0

```

```

]; %5x4
Whidden2=zeros(size(Whidden));
Whidden3=zeros(size(Whidden));
e=0;
e1=1;
training_size=size(trainset);
training_counter=1;
l=1;
q=1;
t=0;
n=0;
tsim=30000;
earr=zeros(1,(floor(tsim/2000)));
tmod=0;

Out=[ zeros(size(y))
];
hidden_layer=[ 1
1
1
1
1
1
1
% 1
-1
];
while(t<tsim)
    while(training_counter<(training_size(2)+1))
        while(l<8)
            hidden_layer(l)=1/(1+exp(-(Whidden(l,:)*trainset(:,training_counter))));
            l=l+1;
        end
        l=1;
        while(q<2)
            Out(training_counter)=1/(1+exp(-(Wout(:,q)*hidden_layer)));
            Wout(:,q)=Wout(:,q)+0.03*(y(training_counter)-
            Out(training_counter))*Out(training_counter)*(1-
            Out(training_counter))*hidden_layer+0.1*(Wout2-Wout3);
            Wout3=Wout2;
            Wout2=Wout;
            q=q+1;
        end
        q=1;
    while(l<5)
        Whidden(l,:)=Whidden(l,:)+nu*(y(training_counter)-
        Out(training_counter))*Out(training_counter)*(ones(1,1)-
        Out(training_counter))*Wout(l,:)*hidden_layer(l)*(ones(1,1)-

```

```

        hidden_layer(l))*trainset(:,training_counter)'+0.1*(Whidden2(l,:)-
        Whidden3(l,:));
        Whidden3(l,:)=Whidden2(l,:);
        Whidden2(l,:)=Whidden(l,:);
        l=l+1;
        n=n+1;
    end
    l=1;

        e=e+1/2*(y(training_counter)-Out(training_counter))^2;
    e1=e;
    tmod=mod(t,200);
    if(tmod==0)
        earr(floor(t/200)+1)=e;
    end
        training_counter=training_counter+1;
    end
    if(e1>0.075)
        e=0;
        training_counter=1;
    end
    t=t+1;
end

```

EK A2

ÇAMAŞIR MAKİNESİNDE YSA İLE YÜK KESTİRİM ALGORİTMASININ C# KULLANARAK MİKROKONTROLÖRDE GERÇEKLENMESİ

```

#include <pic.h>
#include <stdio.h>
#include <math.h>
double oll;
double ol;
int olint;
double nt;
double bank1 qq[4];
double owl[5];
double OO;
double O;
double tt;
double output;
double bank1 out;
char l;
char k;
unsigned int zaman;
int ntt;
int ttt;
double f[4]={falltime, activation, vspinum, risesum
};
const double ss[128]={0.0018329, 0.0020253, 0.0022378,
0.0024726,
0.0027319, 0.0030184, 0.0033348, 0.0036842,
0.0040701, 0.0044962, 0.0049668, 0.0054862,
0.0060598, 0.0066928, 0.0073915, 0.0081625,
0.0090132, 0.0099518, 0.0109869, 0.0121284,
0.0133869, 0.0147740, 0.0163024, 0.0179862,

```

```

0.0198403,    0.0218812,    0.0241270,    0.0265969,
0.0293122,    0.0322954,    0.0355711,    0.0391657,
0.0431072,    0.0474258,    0.0521535,    0.0573241,
0.0629733,    0.0691384,    0.0758581,    0.0831726,
0.0911229,    0.0997504,    0.1090968,    0.1192029,
0.1301084,    0.1418510,    0.1544652,    0.1679816,
0.1824255,    0.1978161,    0.2141650,    0.2314752,
0.2497398,    0.2689414,    0.2890504,    0.3100255,
0.3318122,    0.3543436,    0.3775406,    0.4013123,
0.4255574,    0.4501660,    0.4750208,    0.5000000,
0.5249791,    0.5498339,    0.5744425,    0.5986876,
0.6224593,    0.6456563,    0.6681877,    0.6899744,
0.7109495,    0.7310585,    0.7502601,    0.7685247,
0.7858349,    0.8021838,    0.8175744,    0.8320183,
0.8455347,    0.8581489,    0.8698915,    0.8807970,
0.8909031,    0.9002495,    0.9088770,    0.9168273,
0.9241418,    0.9308615,    0.9370266,    0.9426758,
0.9478464,    0.9525741,    0.9568927,    0.9608342,
0.9644288,    0.9677045,    0.9706877,    0.9734030,
0.9758729,    0.9781187,    0.9801596,    0.9820137,
0.9836975,    0.9852259,    0.9866130,    0.9878715,
0.9890130,    0.9900481,    0.9909867,    0.9918374,
0.9926084,    0.9933071,    0.9939401,    0.9945137,
0.9950331,    0.9955037,    0.9959298,    0.9963157,
0.9966651,    0.9969815,    0.9972680,    0.9975273,

0.9977621,    0.9979746,    0.9981670,    0.9983411
};
const double W[5] ={
0.4918771,
6.3114931,
2.4076605,
-4.0566471,
1.7718957
};
const double w1[4][4] ={
-1.3046,    2.5708,    2.7683,    -1.0736,
0.7129,    7.3869,    3.6558,    4.0280,
3.8344,    3.6155,    0.3187,    -0.8538,
0.1393,    -2.7486,    -0.7953,    -3.4617};
}
//*****
***/
void yuk_hesapla(void);
const long double polyk_kuruyuk_cikis[4]={-44.707e+03,21.567e-
02,6.350e+01,-27.987e-01};
void main(void)
{
    OPTION = 0b10000111;
    T1CON = 0X31;                // HIZ OKUMA interrupt lari
    TMR1IE = 1;
    GIE = 1;
    PEIE = 1;
    TMR1ON=0;
    f[0]=f[0]*3/1000;
    f[1]=f[1]/1000;
    f[2]=f[2]/10000;
    f[3]=f[3]/20000;
    while(1){
        nt=0;
        tt=0;

```

```

        l=0;
        k=0;
        ntt=0;
        ttt=0;
        ol=0;
        output=0;
        ol=0;
        O=0;
        yuk_hesapla();
        tt=l+k;
    }
}
void yuk_hesapla(void){
    TMR1ON=1;
    TMR1H=0;
    TMR1L=0;
while(l<4){
    while(k<4){
        o1l=w1[l][k]*f[k];
        ol=ol+o1l;    //ol=0
        qq[l]=o1l;
        k++;
    }
    nt=(ol+6.4)*10;
    ntt=(int) nt;

    owl[l]=ss[ntt];
    k=0;
    ol=0;
    l++;
}
owl[l]=-1;
l=0;
while(l<5){
    OO=W[l]*owl[l];
    O=O+OO;
    l++;
}
tt=(O+6.4)*10;
ttt=(int)tt;
output=ss[ttt];
out=(output/0.15);
zaman=(TMR1H*256)+TMR1L;
TMR1ON=0;
return;
}

```


ÖZGEÇMİŞ

Ad Soyad: Reyhan Türk
Doğum Yeri ve Tarihi: Kırcaali 16.05.1981
Adres: Denizköşkler Mah. Kent Çıkması S. No:6/6
Avcılar/İstanbul
Lisans Üniversite: Orta Doğu Teknik Üniversitesi