

**SENSÖR LİNEERLEŞTİRME AMAÇLI
MİKRODENETLEYİCİLİ TASARIM**

**YÜKSEK LİSANS TEZİ
Müh. Ömer DOĞAN
(504031219)**

**Tezin Enstitüye Verildiği Tarih : 8 Mayıs 2006
Tezin Savunulduğu Tarih : 15 Haziran 2006**

Tez Danışmanı : Prof.Dr. Sadri ÖZCAN (İTÜ)
Diğer Jüri Üyeleri: Yrd.Doç.Dr. Neslihan Serap ŞENGÖR (İTÜ)
Doç.Dr. Oğuzhan ÇİÇEKOĞLU (BÜ)

HAZİRAN 2006

ÖNSÖZ

Yüksek lisans tez çalışmam kapsamında bana her türlü desteği sağlayan Sn. Prof. Dr. Sadri ÖZCAN' a; tez kapsamındaki uygulamaların yapılabilmesi için tüm laboratuvar ve malzeme imkanlarını kullanma olanağını sunan Arçelik Çamaşır Makinesi İşletmesine; yapay sinir ağlarının uygulanması konusunda bilgi birikimini paylaşmaktan çekinmeyen Sn. Yük. Müh. Mahmut MERAL' e; maddi manevi her türlü desteği esirgemeyen aileme teşekkürü bir borç bilirim.

Mayıs, 2006

Müh. Ömer DOĞAN

İÇİNDEKİLER

KISALTMALAR	vi
TABLO LİSTESİ	vii
ŞEKİL LİSTESİ	ix
SEMBOL LİSTESİ	xii
ÖZET	xiii
SUMMARY	xv
1. GİRİŞ	1
2. SENSÖR LİNEERLEŞTİRME YÖNTEMLERİ	4
2.1 Non-linear Analog İşaretin Analog Tasarımla Düzeltilmesi Yöntemi	4
2.1.1 Sensör karakteristiği tersini alma	4
2.1.2 Kısmi parçalı lineerleştirme yaklaşımı	6
2.2 Non-linear Sayısal İşaret İşleme Yöntemi	8
2.2.1 Tablo tutma yöntemi (look-up table)	8
2.2.2 Yapay sinir ağlarının kullanılması	10
2.3 Lineer Olmayan ADC Kullanılması Yöntemi	12
3. YAPAY SİNİR AĞLARI	14
3.1 Yapay Sinir Ağlarının Temel Özellikleri	15
3.1.1 Doğrusal olmama	15
3.1.2 Öğrenme	15
3.1.3 Genelleme	16
3.1.4 Uyarlanabilirlik	16
3.1.5 Hataya karşı tolerans	16
3.2 Nöron Yapısı	17
3.3 Ağ Mimarileri	19
3.3.1 Tek katmanlı cebrik ağlar	19
3.3.2 Çok katmanlı cebrik ağlar	20
3.3.2.1 Çok katmanlı ağlara ilişkin denklemsel ifadeler	21
3.3.3 Radyal tabanlı ağlar	22
3.4 Geriye Yayılım Algoritması	23
3.4.1 Geriye yayılım algoritmasının temel kavramları	24
3.4.1.1 Hata kavramı	24
3.4.1.2 Nörona ait yapılan işlemler	24
3.4.1.3 Hesaplama yönleri	26
3.4.1.4 Etkinleştirme fonksiyonu	26
3.4.1.5 Öğrenme oranı	27
3.4.1.6 Eğitimin ardışıl ve grup modları	29
3.4.1.7 Algoritmayı durdurma kriteri	30
3.4.2 Geriye yayılım algoritmasının yazılımsal ifadesi	31
4. UYGULAMALAR İÇİN OLUŞTURULAN SİSTEMİN YAPISI	33
4.1 Tasarlanan Devrede NTC Sıcaklık Sensörü Okuma Yöntemi	34
4.1.1 A/D çevrim için mikrodenetleyicide yapılması gereken işlemler	37

4.1.2 Elde edilen sayısal verilerin YSA ve tabloya aktarılması	38
4.2 Tasarlanan Devrede Frekans Çıkışlı Basınç Sensörü Okuma Yöntemi	41
4.2.1 Elde edilen verilerin YSA girişine aktarılması	44
5. SENSÖR VERİLERİNİN MİKRODENETLEYİCİDE İŞLENMESİ	45
5.1 Çok Katmanlı Ağ Uygulamaları	48
5.1.1 Çok katmanlı ağ yapıları için oluşturulan C kodunun açıklanması	49
5.1.1.1 Giriş verilerinin normalizasyonu	50
5.1.1.2 Verilerin adreslenmesi	50
5.1.1.3 YSA ağırlıklarına rastgele değerler atanması	50
5.1.1.4 YSA eğitim aşaması	50
5.1.1.5 YSA test aşaması	51
5.1.1.6 Verilerin bellekten boşaltılması	51
5.1.2 Çok katmanlı ağ yapıları için mikrodnetleyiciye girilen YSA yazılımı	52
5.2 Radyal Taban Fonksiyonları Kullanılarak Yapılan Uygulamalar	56
5.2.1 Radyal taban fonksiyonlarının eğitimi için oluşturulan C kodu	57
5.2.2 Radyal taban fonksiyonları uygulamaları için mikrodnetleyicide oluşturulan YSA yazılımı	58
5.3 Tablo Tutma Yöntemi İle Yapılan Uygulamalar	60
5.3.1 Tablo tutma yöntemi için mikrodnetleyicide yapılan yazılım	61
5.3.1.1 Sensör dönüşüm tablosu	61
5.3.1.2 Tabloda verinin taranması	61
5.3.1.3 Tabloda bulunan değerin bilgisayara aktarılması	61
5.4 Radyal Taban Fonksiyonlarına İlişkin Yapılan Uygulamaların Sonuçları	62
5.5 Çok Katmanlı Ağ Yapılarına İlişkin Uygulamaların Sonuçları	68
5.6 Tablo Tutma Yöntemi ile Yapılan Uygulamaların Sonuçları	74
6. SONUÇLAR VE TARTIŞMA	77
KAYNAKLAR	83
EK A	85
ÖZGEÇMİŞ	86

KISALTMALAR

AC	: Alternatif Akım (Alternating Current)
ADC	: Analog Sayısal Çevirici (Analog to Digital Converter)
CCP	: Capture Compare PWM Modülü
ÇKA	: Çok Katmanlı Ağ Yapıları
ÇM	: Çamaşır Makinesi
DC	: Doğru Akım (Direct Current)
GKDS	: Gizli Katman Düğüm Sayısı
MLP	: Çok Katmanlı Ağ Yapıları (Multi-Layer Perceptron)
NN	: Yapay Sinir Ağları (Neural Network)
NTC	: Negatif Isıl Katsayılı Direnç (Negative Thermal Coefficient)
OKH	: Ortalama Karesel Hata
RBF	: Radyal Taban Fonksiyonları (Radial Base Functions)
UART	: Üniversal Asenkron Alma Vere Protokolü
YSA	: Yapay Sinir Ağları

TABLO LİSTESİ

	<u>Sayfa No</u>
Tablo 5.1 : Çok katmanlı ağırlarla oluşturulan YSA yapısı için giriş katmanı ile gizli katman arasındaki ağırlıkları içeren tablo	53
Tablo 5.2 : Çok katmanlı ağırlarla oluşturulan YSA yapısı için gizli katman ile çıkış katmanı arasındaki ağırlıkları içeren tablo	53
Tablo 5.3 : Radyal taban fonksiyonları ile oluşturulan YSA için gizli katmanında 8 nöron içeren yapıda nöronların merkezlerini içeren tablo	59
Tablo 5.4 : Radyal taban fonksiyonları ile oluşturulan YSA için gizli katmandaki nöronlarla çıkış katmanı arasındaki ağırlıkları içeren tablo	59
Tablo 5.5 : Çamaşır makinesi sıcaklık sensörü için radyal taban fonksiyonları ile oluşturulan YSA yapılarına ilişkin ortalama karesel hata değerleri	63
Tablo 5.6 : Fırın sıcaklık sensörü için radyal taban fonksiyonları ile oluşturulan YSA yapılarına ilişkin ortalama karesel hata değerleri	64
Tablo 5.7 : Çamaşır makinesi basınç sensörü için radyal taban fonksiyonları ile oluşturulan YSA yapılarına ilişkin ortalama karesel hata değerleri	66
Tablo 5.8 : Radyal taban fonksiyonları ile oluşturulan YSA ile yapılan uygulamalar sonucu gizli katmandaki her bir nöron sayısı ile yapılan uygulamaların bellekte kapladığı alan	67
Tablo 5.9 : Çamaşır makinesi NTC için çok katmanlı ağ yapıları ile oluşturulan YSA yapılarına ilişkin ortalama karesel hata değerleri	69
Tablo 5.10 : Fırın NTC için çok katmanlı ağ yapıları ile oluşturulan YSA yapılarına ilişkin ortalama karesel hata değerleri	70
Tablo 5.11 : Çamaşır makinesi basınç sensörü için çok katmanlı ağ yapıları ile oluşturulan YSA yapılarına ilişkin ortalama karesel hata değerleri	72

Tablo 5.12 : Çok katmanlı Yapay Sinir Ağları ile yapılan uygulamalar sonucu gizli katmandaki her bir nöron sayısı ile yapılan uygulamaların bellekte kapladığı alan	74
Tablo 5.13 : Tablo tutma yöntemi ile yapılan uygulamalar sonucunda elde edilen ortalama karesel hata değerleri	75
Tablo 5.14 : Tablo tutma yöntemi kullanılarak yapılan uygulamalarda mikrodnetleyicide tutulan bellek miktarı	76
Tablo 6.1 : Çamaşır makinesi NTC ile her iki YSA yönteminin uygulanması sonucu ortaya çıkan ortalama karesel hata tablosu	78
Tablo 6.2 : Fırın NTC ile her iki YSA yönteminin uygulanması sonucu ortaya çıkan ortalama karesel hata tablosu	79
Tablo 6.3 : Çamaşır makinesi basınç sensörü ile yapılan YSA tabanlı algoritmanın her iki yöntem için ortalama karesel hata değerleri	79
Tablo 6.4 : Uygulamalarda RBF ve Sigmoid fonksiyonları ile oluşturulan YSAların mikrodnetleyici belleğinde kapladıkları alanlar	80

ŞEKİL LİSTESİ

	<u>Sayfa No</u>
Şekil 1.1 : Herhangi bir NTC için non-lineer çıkış karakteristiği	2
Şekil 1.2 : Sensör lineerleştirme amaçlı oluşturulan yapının blok diyagramı	2
Şekil 2.1 : Analitik tersini alma yönteminin blok diyagramı	5
Şekil 2.2 : Analitik tersini alma yönteminin grafiklerle ifadesi	5
Şekil 2.3 : Kısmi Parçalı Lineer Yaklaşım Yöntemi ile Modellenen Sensöre Ait Grafik	7
Şekil 2.4 : Tablo tutma yönteminin blok diyagramla gösterilimi	8
Şekil 2.5 : Yapay Sinir Ağının Genel Yapısı	10
Şekil 2.6 : Eğitilmiş yapay sinir ağında nöronun genel yapısı	11
Şekil 2.7 : Non-lineer karakteristikli A/D Çevirici ile yapılan sensör lineerleştirme yapısı blok diyagramı	12
Şekil 2.8 : Non-lineer A/D Çevirici Yapısı[7]	12
Şekil 3.1 : Temel nöron modeli	17
Şekil 3.2 : Bazın kullanıldığı nöron modeli	18
Şekil 3.3 : Çok katmanlı ağ yapısı[19]	20
Şekil 3.4 : Radyal taban fonksiyonunun genel yapısı[19]	22
Şekil 3.5 : j numaralı çıkış nöronuna ait işaret akış diyagramı	24
Şekil 4.1 : Sistem için kullanılan devre blok diyagramı	33
Şekil 4.2 : NTC Analog gerilim çıkışı okuma algoritması	34
Şekil 4.3 : NTC değerini okumak için tasarlanan analog devre yapısı	35
Şekil 4.4 : Mikrodenetleyici içerisinde tümleşik olarak bulunan A/D Çeviricinin' nin giriş modeli	36
Şekil 4.5 : İleri yol YSA yapısı	38
Şekil 4.6 : Frekans çıkışlı basınç sensörü okuma algoritması	41
Şekil 4.7 : Basınç sensörünü frekansa çevirmek için kullanılan devre yapısı	42
Şekil 4.8 : CCP modülü blok diyagramı	42
Şekil 5.1 : Veri depolama sisteminin genel yapısı	46

Şekil 5.2	: Uygulamalarda kullanılan ÇM NTC sıcaklık sensörünün, çalışılan bölge için genel karakteristiği	47
Şekil 5.3	: Uygulamalarda kullanılan fırın NTC sıcaklık sensörünün, çalışılan bölge için genel karakteristiği	47
Şekil 5.4	: Uygulamalarda kullanılan frekans çıkışlı basınç sensörünün, çalışılan bölge için genel karakteristiği	48
Şekil 5.5	: Çok Katmanlı YSA için Kullanılan Sistemin Yapısı	48
Şekil 5.6	: Tablo tutma yöntemi için kullanılan sistemin blok diyagramı	60
Şekil 5.7	: Çamaşır makinesi NTC uygulamasında gizli katmanında iki nöron içeren ve radyal taban fonksiyonları ile oluşturulmuş YSA ile her bir sensör ADC değerine karşılık referans sıcaklık ve YSA sıcaklık çıkış değerleri grafiği	63
Şekil 5.8	: Çamaşır makinesi NTC uygulamasında gizli katmanında iki nöron içeren ve radyal taban fonksiyonları ile oluşturulan YSA çıkışında her bir referans sıcaklık değerine karşılık elde edilen sıcaklık fark değer grafiği	64
Şekil 5.9	: Fırında kullanılan NTC uygulamasında gizli katmanında iki nöron içeren radyal taban fonksiyonları ile oluşturulan YSA için 10 bit ADC verilerine karşılık olarak referans sıcaklık ve YSA çıkışındaki sıcaklık değerleri	65
Şekil 5.10	: Fırında kullanılan NTC ile yapılan uygulamalarda gizli katmanında iki nöron içeren radyal taban fonksiyonları ile oluşturulan YSA için her bir referans değere karşılık referans sıcaklık değerleri ile ağ çıkışları arasındaki farkları içeren grafik	65
Şekil 5.11	: Çamaşır makinesi NTC uygulamasında gizli katmanında iki nöron içeren YSA yapısı sonucu her bir sensör ADC verisine karşılık referans değer ve YSA çıkışı sıcaklık değerleri	69
Şekil 5.12	: Çamaşır makinesi NTC uygulamasında, gizli katmanında iki nöron içeren çok katmanlı ağ yapıları ile oluşturulan YSA uygulamaları sonucu her bir referans sıcaklık değerine karşılık elde edilen fark grafiği	70
Şekil 5.13	: Fırın NTC uygulamalarında gizli katmanında iki nöron bulunan çok katmanlı YSA için sıcaklık ADC grafiği	71

- Şekil 5.14** : Fırın NTC uygulamalarında gizli katmanında iki nöron bulunan sistem için referans sıcaklıklar ile her bir referans sıcaklık değerindeki farklar arasındaki grafik 72
- Şekil 5.15** : Tablo tutma yöntemi ile yapılan fırın sıcaklık sensörü uygulaması sonucu referans değerler ile tabloda tutulan değerler arasındaki farkların dağılım grafiği 75

SEMBOL LİSTESİ

$\varphi()$	: Etkinleştirme Fonksiyonu
$\omega_{ij}()$	: Bir katmandaki i numaralı nöron ile bir sonraki katmandaki j numaralı nöronlar arası ağırlık
$m_{ij}()$	: Ağların merkezleri
y	: Ağ çıkışı
x	: Ağ girişi
e_{jn}	: İlgilenilen yerdeki ağın çıkışı ile olması gereken değer arasındaki farkı ifade eden hata değeri
u_k	: Nöronlar arası ağırlıklar ile giriş değerlerinin çarpımların toplamı
W	: Çıkış katmanına ilişkin bağlantı matrisi
μ_j	: j numaralı nöron için baz fonksiyonunun merkezi
σ	: Standart Sapma
$d_j()$	: j numaralı nörona ait arzulanan çıkış
$\xi_n()$	: Hatalara ilişkin enerjilerin ani değerlerinin toplamı
$\xi_{av}(n)$	: Ortalama hata enerjisi
$v_j(n)$	: Gizli katmana gelen girişlerin nöronlar arası ağırlıklarla çarpımlarının toplamı
$\Delta w_{ij}(n)$	: Ağırlık düzeltme faktörü
η	: Öğrenme hızı parametresi
$\delta_j(n)$	: Yerel gradiyent
α	: Momentum sabiti
mU	: YSA eğitimi için kullanılan sensör ADC verilerinin en küçüğü
MU	: YSA eğitimi için kullanılan sensör ADC verilerinin en büyüğü
MY	: YSA eğitimi için kullanılan referans sensör verilerinin en küçüğü
MY	: YSA eğitimi için kullanılan referans sensör verilerinin en büyüğü

SENSÖR LİNEERLEŞTİRME AMAÇLI MİKRODENETLEYİCİLİ TASARIM

ÖZET

Mikrodenetleyicilerin günlük hayatımızdaki kullanım alanları gün geçtikçe artmaktadır. Özellikle kontrol uygulamalarında mikrodenetleyiciler sık sık kullanılmaktadır. Kontrol uygulamalarında, sensör verisini okuma ve gelen verilerle işlem yapma sık rastlanılan bir durumdur. Kontrol sisteminden istenilen başarımın elde edilebilmesi için okunan sensör verisinin ilgilenilen çalışma aralığında yüksek doğrulukta elde edilebilmesi gerekmektedir. Yüksek doğruluklu değer elde edilebilmesindeki en büyük engel, sensörlerin genel olarak karakteristiğinin non-lineer olmasıdır. Bu tez çalışmasında, belirtilen engeli ortadan kaldırma amaçlı sensör karakteristiği lineerleştirme konusu ele alınmıştır.

Konu ile ilgili olarak, donanımsal ve yazılımsal lineerleştirme yöntemlerine değinilmiş, yapay sinir ağlarının ve tablo tutma yönteminin kullanıldığı mikrodenetleyicili tasarımlar üzerinde uygulamalı çalışmalar yapılmıştır. Uygulamalar için çamaşır makinesi NTC sıcaklık sensörü (EPCOS), çamaşır makinesi basınç sensörü (DAEHAN) ve fırın NTC sıcaklık sensörü (EPCOS) kullanılmıştır.

Yapay sinir ağları ile yapılan uygulamalar, özellikle modeli tam olarak belirlenemeyen sensörler için yüksek doğrulukla modelleme yapabilme ve ardından elde edilen karakteristiğe ilişkin sensör karakteristiğinin lineerleştirmesini sağlamaktadır. Yapay sinir ağlarının sensör lineerleştirme yapılarındaki performansını incelemek için çok katmanlı ağ yapıları ve radyal taban fonksiyonlarının kullanıldığı ağ yapıları kullanılmıştır. Çok katmanlı ağ yapılarında aktivasyon fonksiyonu olarak sigmoid fonksiyonları kullanılmıştır. Yapay sinir ağları parametreleri oluşturulurken ağın eğitim aşaması bilgisayar ortamında yapılmış; mikrodenetleyici içerisine, eğitimi yapılmış ve parametreleri oluşturulmuş ağın ileri yol aşaması girilmiştir. Mikrodenetleyiciye yapay sinir ağı parametrelerinin girilmesi ve ağın koşturulması aşamasında mikrodenetleyicinin mevcut olan yapıları kullanıldığından bu durumda ek devrelerin kullanılmasına ya da tasarlanmasına gerek kalmamıştır.

Uygulamalarda kullanımına en sık rastlanılan sensör lineerleştirme yöntemlerinden birisi, sensör referans değerlerin bir tabloda tutulduğu tablo tutma yöntemidir. Tablo tutma uygulamalarında, referans sensör verileri ile sensör değerlerini içeren dönüşüm tabloları kullanılmıştır. Dönüşüm tablosunun girişine gelen sensör değerlerine karşılık, tablonun çıkışında, lineer karakteristiğin elde edileceği değerlere ulaşılmıştır.

Yapay sinir ağlarının ve tablo tutma yöntemlerinin mikrodenetleyici programlarına aktarılması ve programların mikrodenetleyicide koşturulması ile yöntemler arasında bir takım farklılıklara rastlanmıştır.

Tasarımcının tablo tutma yöntemi ile YSA yöntemlerinden birisini seçme kararında mikrodenetleyici içerisinde kaplanan alan, ortalama karesel hata kriteri ve YSA yapılarına has olan genelleştirme kriterleri göz önünde bulundurulmalıdır. Tasarımcının, çalıştığı sensörle geniş bir çalışma aralığında düşük ortalama karesel hata ile çalışmak ve bu kriterler için aynı zamanda mikrodenetleyici bellek alanından tasarruf etmek istemesi halinde, tablo tutma yöntemi yerine YSA yöntemlerine yönelmelidir. Bunun yanında, YSA yapılarının temel karakteristiği olan genelleştirme özelliği, YSA yöntemini tablo tutma yöntemine göre daha ön plana çıkarmaktadır.

Sensör lineerleştirme amacıyla kullanılacak sistemde YSA yöntemlerinden tasarımcıya radyal taban fonksiyonları ya da çok katmanlı ağ tabanlı YSA yöntemleri önerilebilir. Belirtilen iki yöntem arasında daha düşük ortalama karesel hata ile çalışabilmek istendiği durumda tasarımcıya radyal taban fonksiyonları ile oluşturulan YSA önerilmektedir.

Veri alanının daha az kullanımı ve neticede tasarım maliyetinin indirilmesi, yapay sinir ağları ile kullanılan yapıyı ön plana çıkarmaktadır. Ardından, iki farklı aktivasyon fonksiyonuna ilişkin yapay sinir ağı yapısı başarımlı açısından karşılaştırılmıştır. Radyal taban fonksiyonlarının kullanımı ile elde edilen yapıda daha az nöron kullanımı ile sistemden beklenen başarımlı elde edilebilmektedir. Neticede, sensör karakteristiği lineerleştirmeye yönelik olarak radyal taban fonksiyonlarının aktivasyon fonksiyonu olarak kullanıldığı yapay sinir ağlarında daha az matematiksel işlem yapılarak ortalama karesel hataya dayalı benzer başarımlı elde edilebilmektedir. Çalışmalar sonucunda, yapay sinir ağlarının radyal taban fonksiyonlarında kullanılması ile oluşturulan sensör lineerleştirme yapısı tasarımcıya önerilmektedir.

MICROCONTROLLER BASED SENSOR LINEARIZATION SYSTEM DESIGN

SUMMARY

Nowadays, microcontroller applications are frequently used in control applications. One of the control applications with microcontroller is reading and modelling sensor output. After reading the sensor output, some critical works can be made. To achieve the good control characteristics, high linearity of the sensor output is needed. Unfortunately, most of the sensor output characteristics are naturally highly non-linear. In this thesis work, sensor linearization technics are explained to overcome the mentioned problem.

In the thesis, mostly used linearization techniques in the applications by software and hardware designs are taken as main topic. After that, applications by neural networks and look-up table on microcontroller are explained. Two different types of NTC thermistors (EPCOS) are used in washing machine and cooker. Pressure sensor (DAEHAN) for washing machine is also used for all of the applications for the thesis.

Neural Networks can be also used for the sensors that can not be exactly modelled. Neural networks provide highly linear outputs due to nonlinear activation functions used in its network. Radial base functions neural networks and multi-layer perceptron neural networks are used to investigate the effect of neural networks for sensor linearization techniques in the applications. Sigmoid function is used as the activation function for MLP NN and gaussian function is used for the RBF NN. While forming the network parameters, back propagation algorithm is used for the training the NN. Training of the network is done on PC for both of NN technics. After training the NN on PC, only feedforward path of the NN is written in microcontroller with network parameters.

Another frequently used technic in sensor linearization is look-up table approach. In this technic, reference values and sensor values are kepted in a sensor transformation table. Mentioned table generates output according to the input from the sensor at that time. Reference values are arranged so that at the output of the table, sensor characteristic should be linear as desired.

While running the applications with NN and look-up table approach, different results are obtained for different type of applications. Evaluation of the technics are done according to mean square error criteria and memory allocation amounts for the technics. Also, generalization feature of the NN should be mentioned. If low amount of mean square error and small memory allocation are desired for wide range of sensor output, NN technics can be suggested instead of look-up table approach. What in more, generalization feature is another powerful point of NN technics compared to look-up table technics.

When NN technics are preferred to look-up table approach for sensor linearization, two method frequently used in the applications are adviced to use. These technics are expressed as MLP NN by sigmoid activation function and RBF NN by gaussian function. When lower mean square error values are desired, RBF NNs are suggested instead of MLP NN. For the memory allocation point of view, there is no big difference between MLP NN and RBF NN. RBF NN needs %0.3 more memory allocation than MLP NN for the same results obtained from the network.

1. GİRİŞ

Sensörler, günlük hayatımızın birçok alanında kullanıma girmiş, bunun yanında kullanımı giderek daha da yaygınlaşmakta olan temel algılama elemanlarından. Uygulamalarını daha çok kontrol sistemleri, ölçüm sistemleri, bilimsel alanlar ve endüstriyel alanlarda görmekteyiz. Genel olarak sensörlerin kullanıcılara verdiği veriler analog gerilim boyutunda olup, analog işarettir. Çıkıştan alınan analog işaret, çeşitli işlemlere tabi tutulduktan sonra amaca uygun olarak tasarımlarda kullanılır[5,11, 18].

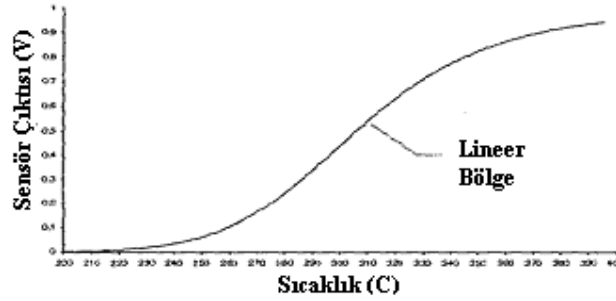
Endüstriyel uygulamalarda sensörler için düşük maliyetli olmaları, hassas olmaları, yüksek doğruluklu olmaları ve lineer karakteristik göstermeleri istenmektedir. Sensörler kontrol altına alınabilir karakteristiğe sahip olmalarına rağmen, ölçülen değişken değerine göre yüksek non-lineer karakteristik göstermektedirler. Bu durumda mevcut sensör çıktısını herhangi bir işleme tabi tutmadan yapılan işlemlerde non-lineerlikten kaynaklanan olumsuzluklar ortaya çıkabilmektedir. Bu durumda sensörlerin modellenmesi ile ilgili zorluklar ortaya çıkmaktadır[6,7,18].

Sensör lineerleştirme olgusunun temelinde yatan, sensörden elde edilen verilerle çizilen karakteristikte görülen eğriselliklerin azaltılması ile sensörün lineer (doğrusal) çalışma aralığını genişletmektir. Böylece sensör kullanarak, daha geniş bölgede lineer kontrol sağlanabilecektir.

İçerikte yer alan Şekil 1.1 ile görüldüğü bir sensörün lineer çalışma bölgesini genişletmek üzere istenilen lineer çıkış ile gerçek sensör çıkışı arasına Şekil 1.2 ile görüldüğü gibi bir lineerleştirme bloku yerleştirilir. Bu yapı, analog sistemler için gelen sensör verisini koşullandırma amaçlı devre olabileceği gibi sayısal sistemlerle oluşturulan ve programlama ile çalışan yapılar da olabilmektedir[8,9].

Uygulamalarda, sensör karakteristiği modelleme için Şekil 1.2 ile görülen blok diyagramdaki ikinci bloğun içerisine genel olarak transfer fonksiyonu ikinci dereceden olan modeller kullanılmaktadır. Ancak, birçok uygulamada, ikinci

derecede modelleme yeterli olmamakta, olması gereken ise daha yüksek derecede modellemelerin kullanılmasıdır[11].



Şekil 1.1: Herhangi bir NTC için non-linear çıkış karakteristiği



Şekil 1.2: Sensör karakteristiğinin lineerleştirilmesi amacıyla yönelik oluşturulan yapının blok diyagramı

Kullanılacak olan sensörden, hedeflenen başarımda yararlanabilmek için istenilen doğrudan çıkışı alabilmek amacıyla sensörün doğal karakteristiğinde görülen non-linear biçim, sistem için hedeflenen derecede lineerleştirilebilmektedir. İstenilen nitelikte lineerleştirmenin yapılması ile sensörün fiziksel çıkışı ile istenilen doğruluktaki çıkış arasında doğrudan ve kesin uyumluluk yakalanacaktır[6,7].

Yapılan tez çalışması altı ayrı bölümde ele alınmıştır. Bölüm 2 içeriğinde genel olarak sensör lineerleştirme yöntemlerinden bahsedilmiştir. Bu bölümde, kullanılan kaynaklardan görülen bilgilere göre en sık kullanılan tasarım türleri ele alınmıştır. Anlatılan her bir tasarım yönteminin önyararına ve önsakıncalı yönlerine değinilmiştir.

Bölüm 3 içeriğinde, uygulamalarda kullanılan YSA yapılarının okuyucu tarafından anlaşılır hale gelmesi için YSA genel olarak ele alınmıştır. Bu kapsamda en temel yapı olan nörondan YSA ağ yapısının tamamına kadar yapılar ele alınmıştır. Ayrıca, kullanılan YSA yapılarının birbirleri ile olan ilişkileri konusu da işlenmiştir. YSA temel özellikleri de bu bölümde ele alınmıştır.

Bölüm 4 içeriğinde, uygulamalar kapsamında oluşturulan sistemlerin yapılarına değinilmiştir. Bu kapsamda öncelikle veri depolama sisteminin okuyucu tarafından kolaylıkla kavranabilmesi için yapı, şekillerle birlikte anlatılmıştır. Bunun yanında

sensörleri okumak için nasıl bir donanımın tasarlandığı, devre şekilleri ile desteklenerek anlatılmıştır. Ek olarak, mikrodenetleyici içerisinde sensör okuma için yapılan algoritmalar, yapılan ayarlamalar açıklanmıştır.

Bölüm 5 içeriğinde tez çalışması kapsamında yapılan uygulamalar belirtilmiştir. Bölümün içeriğinde, hangi sensörlerin çalışıldığı, hangi sensöre hangi uygulamaların yapıldığı, uygulamaların ayrıntılı olarak açıklanması yapılmıştır. Uygulamaların her birinin sonuçları belirtilmiş ve sonuçlar grafik ve tablolarla desteklenmiştir.

Bölüm 6 içeriğinde ise uygulamalar ile elde edilen sonuçlar belirtilmiştir.

2. SENSÖR LİNEERLEŞTİRME YÖNTEMLERİ

Uygulamalarda, birçok sensör karakteristiği lineerleştirme yöntemi kullanılmaktadır. Bu yöntemler genel olarak 3 grup altında sınıflandırılabilir[5-9,11].

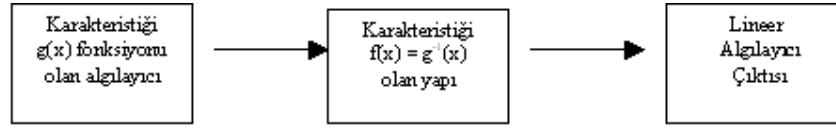
- a. Non-linear analog işareti analog tasarımla düzeltme yöntemi (Non-linear analog signal conditioning)
- b. Non-linear sayısal işaret işleme yöntemi
- c. Non-linear analog / sayısal çevrici yöntemi

2.1 Non-linear Analog İşaretin Analog Tasarımla Düzeltilmesi Yöntem

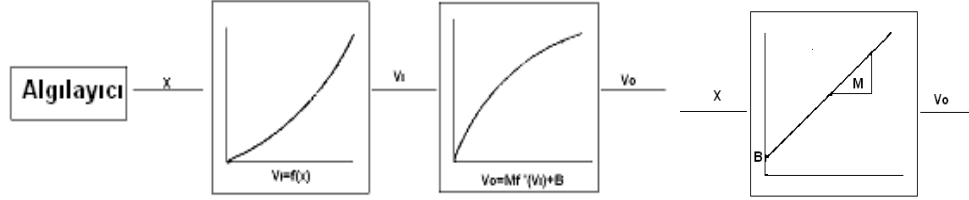
Sensör karakteristiğinin lineerleştirilmesi konusunda analog devre yapılarından da yararlanılmaktadır. Bu tür devrelerin tasarımında temel amaç, lineerleştirilmesi için seçilen hedef sensör karakteristiğini, kullanılacak olan yönteme göre uygun sistem parametrelerini analog devre yapıları ile sağlamaktır. Lineerleştirmeyi sağlayacak parametrelerin oluşturulması için birçok yöntem kullanılmaktadır. Bunlardan bazıları aşağıdaki maddelerde belirtilmiştir.

2.1.1 Sensör karakteristiği tersini alma

Analitik tersini alma yöntemi, uygulamada en fazla karşılaşılan sensör lineerleştirme tekniklerinden birisidir. Bu yöntemde, sensörün çalışma karakteristiği, sensörün kullanılacağı sisteme uygun kabul edilebilir bir hata seviyesinin altında belirlenir. Ardından sensör lineerleştirme amacına yönelik olarak sensörün çıkış karakteristiğinin tam tersi olan analog işaret işleme devresi sentezlenir. Örneğin, çıkış karakteristiği logaritmik ifadelerle belirlenebilen bir sensör için lineerleştirme amacıyla sentezlenecek devre yapısı, parametreleri üstel olan analog işaret işleme devresi olacaktır. Konu ile ilgili blok diyagram, Şekil 2.1 içeriğinde görülmektedir[11].



Şekil 2.1: Analitik tersini alma yönteminin blok diyagramı



Şekil 2.2: Analitik tersini alma yönteminin grafiklerle ifadesi

Belirtilen şekilde model parametrelerine göre oluşturulan analog devrenin girişine verilen, genellikle gerilim boyutunda olan non-lineer karakteristiğe sahip analog sensör, devrenin çıkışında lineer karakteristiğe sahip sensöre dönüştürülür. Lineer karakteristikli sensörün elde edilmesinin ardından basit bir ölçeklendirme yolu ile işlenmiş olan işaret, istenilen birimlere çevrilebilir[11].

Oluşturulabilecek yapının grafiklerle ifadesi, Şekil 2.2 ile gösterilmiştir. Şekil 2.2 grafiğinde görülen x bağımsız değişkeni, ilgilenilen sensörün çıkışını ifade etmektedir. x bağımsız değişkeninin referans sıcaklık değerlerine karşılık aldığı değerlerle grafikte görülen V_1 fonksiyonu oluşturulmuştur. V_1 fonksiyonunun istenilen amaca uygun lineer hali, V_0 fonksiyonu ile ifade edilmiştir. Analitik tersini alma amaçlı yapılarda V_0 ile V_1 arasında aşağıda belirtilen ilişki olmalıdır[11].

$$V_o = Mf^{-1}(V_i) + B \quad (2.1)$$

Denklemdaki M sabiti, lineerleştirme sonunda oluşacak lineer V_0 fonksiyonunun eğimini ifade etmektedir. Sabit, genel olarak 1 değerini almaktadır. Ancak, bazı özel uygulamalar için farklı değerler alabilmektedir. Denklemden görülen B sabiti, referans değer en küçük olduğu durumda sensör değerinin alması gereken değeri ifade etmektedir. Belirtilen aşamaların sonunda tasarımı hedeflenen sensör karakteristiğinin grafiği Şekil 2.2 ile $V_o=f(x)$ grafiği ile gösterilmiştir[11].

Yöntem, anlaşılması ve uygulanması oldukça açık ve basit bir yöntem olarak görünmesine rağmen bir takım sorunları da bünyesinde barındırmaktadır. Analitik ters alma fonksiyonunun devre uygulaması, sıcaklık ve gürültüye karşı yüksek duyarlılık gösterebilmektedir[6,8,11].

Analitik ters fonksiyon alma amacıyla oluşturulmuş devre, hedef sensöre özel bir tasarımıdır ve esnek değildir. Buna göre farklı karakteristiğe sahip her sensör için ayrı devre tasarımı gerekecektir. Bu durum, tasarımcı açısından zaman alıcı ve maliyetli olabilmektedir[1,11].

Sensör karakteristiği düzeltme amaçlı tasarımlarda karşılaşılan bir diğer sorun, çözünürlükle ilgilidir. Bu tür tasarımların transfer fonksiyonu, sensör karakteristiğinin tamamını içeren fonksiyonun tersi ya da belirlenen kısmi bir çalışma bölgesinin tersi olabilmektedir[11].

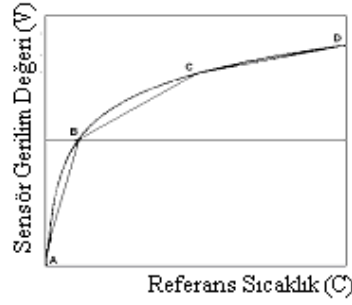
Sensörün belirli bir çalışma bölgesi için yapılan tasarımlarda, tasarımın kriter olarak alındığı bölgelerde istenilen çözünürlükte sonuç alınabilmekte, fakat tasarımın hedef alındığı bölgenin dışında kalan kısımlarda ve özellikle sensör çalışma bölgesinin uç kısımlarında, tasarlanan analog devrede doymaya girmeler görülebilmektedir. Tüm çalışma bölgesinin hedef alındığı devre tasarımlarında devrenin başarımının bölgesel olarak incelendiğinde yeterli derecede çözünürlük elde edilemeyen kısımlar görülebilmektedir. Buna göre hassas olarak okuma yapmak zorlaşmaktadır[11].

Bu yöntemin en büyük önvararı, analitik olarak tersine alma yönteminin sayısal yöntemlerle gerçekleştirilen uygulamalara göre daha hızlı olmasıdır[11].

2.1.2 Kısmi parçalı lineerleştirme yaklaşımı

Uygulamada görülen yöntemlerden birisi, kısmi parçalı lineerleştirme yaklaşımı yöntemidir. Analitik tersine alma yöntemi ile paralel olarak kısmi parçalı lineer yapı ile sensörün mevcut transfer fonksiyonunun tersinin alınması hedeflenmiştir[6,7,10,11].

Bunu sağlamak amacıyla sensörün çıkış eğrisi karakteristiğinin belli noktaları ve bu belli noktalar arasında kalan her bir parçasının kazancı lineerleştirmeye uygun seçilir[10,11].



Şekil 2.3: Kısmi Parçalı Lineer Yaklaşım Yöntemi ile Modellenen Sensöre Ait Grafik

Yöntemin tasarımlara uygulanmasında sensör çıkışı Şekil 2.3 ile gösterildiği gibi A, B, C, D gibi belirlenen noktalar ile farklı bölgelere ayrılmaktadır. Eğri üzerinde her belirlenen nokta ile ayrılan bölge lineer kabul edilmektedir. Bunun yanında, belirlenen noktalar arasındaki her bir bölgenin kazancı birbirinden farklı olmaktadır[6,7,11].

Bu yöntem, sensör karakteristiğinin analitik olarak belirlenemediği zamanlarda kullanışlı olmaktadır. Bu yöntem kullanılarak oluşturulan devrenin karmaşıklığı, kullanılacak sensörün çalışma aralığı ve devreden istenilen çözünürlüğe göre değişmektedir. Bu yöntem kullanılarak geniş bir çalışma aralığında yüksek çözünürlük isteniyorsa, yöntemi gerçeklemek için fazla sayıda komponent kullanmak gerekmektedir[6,7,10,11].

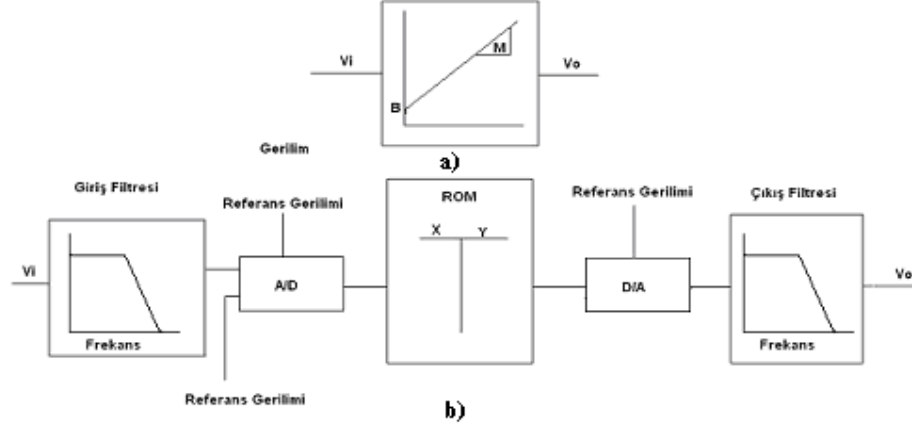
Analitik tersine alma yönteminde belirtilen birçok sakınca, bu yöntemle ortadan kalkmıştır. Yöntem tasarımcıya hedef karakteristiğe uygun olarak belirleyebileceği noktaları ve belirlenen noktalar arasındaki her bir lineer parçanın kazancını kendisinin belirleyebilmesi şansı tanınmaktadır[10,11,17].

Kesikli lineer yaklaşım yönteminin temel sorunu, farklılaşan karakteristiklere göre lineer kabul edilen bölgelerin yeniden kalibrasyonunda görülür. Eğri üzerinde belirlenen bu noktalarının kalibrasyonunun yapılması esnasında, bir noktanın kalibrasyonu, diğer noktanın kalibrasyonunu etkilemektedir. Bu nedenle, sistem için farklı karakteristiklerdeki sensörlerin kalibrasyonu zaman alıcı bir süreç halini almaktadır[10,11].

2.2 Non-lineer Sayısal İşaret İşleme Yöntemi

2.2.1 Tablo tutma yöntemi (look-up table)

Lineerleştirme, sayısal devrelerle yapılan tasarımlarla da gerçekleştirilebilmektedir. Bunu gerçekleştirebilmenin yollarından birisi, sensörün sayısal veriye dönüştürülmüş her bir çıkış değerine karşılık tablonun tutulduğu yöntemdir[11,13,17].



Şekil 2.4: Tablo tutma yönteminin blok diyagramla gösterilmesi

Tablo tutma yönteminin blok diyagramı Şekil 2.4 ile gösterilmiştir. Ulaşılmak istenen hedef sensör karakteristiği, Şekil 2.4 a) ile gösterilen grafik. Blok diyagramda görülen V_i girişi, sensörden alınan analog giriş işaretini göstermektedir. V_i nin fonksiyonu Şekil 2.2 a) içeriğinde verilmiştir. Blok diyagram, V_i değerini gerilim boyutunda olduğu kabul edilerek çizilmiştir. Yapıya göre öncelikle gelen analog sensör değeri alçak geçiren filtreden geçirilmektedir. Filtreden geçirmenin temel nedeni, sensörden alınan işarete görülen istenmeyen ve çok ani görülen değişimleri önlemektir. Filtrelenmiş olan analog sensör gerilimi, A/D çevrici ile dijital işarete çevrilmektedir. Kullanılan A/D çevricinin karakteristiği konusunda uygulamalarda farklılıklar görülmektedir. Genellikle lineer karakteristikte A/D çevricilerin kullanılmasının yanında non-lineer karakteristikli A/D çeviricinin kullanıldığı uygulamalara da rastlanmaktadır[11].

Tablo tutma yönteminin uygulandığı tasarımlarda tablonun oluşturulması için son tasarımın oluşturulmasından önce istatistiksel olarak yeterliliğin sağlanabileceği sayıda, içeriğinde sensörün sayısal verisi ve buna karşılık gelen referans sensör verisi değerleri olan veri toplanır. Toplanan verilere örnek olarak sıcaklık sensörü

uygulamasý için 8 bitlik bir sistemde 0 ile 256 arasında deęişen sayýsal sensör verileri ile bunlara karřılık gelen $^{\circ}\text{C}$ cinsinden sıcaklık deęerleri verilebilir. Veri toplamanın ardından istatistiksel yöntemlerle her bir sayýsal sensör verisine karřılık gelen referans sensör deęerleri ile tablo oluřturulur. Bu tablo, ROM tipi belleęe kaydedilir[11,13].

ROM belleęe verilerin girilmesinin ve sensörden gelen her yeni veriye karřılık tablo deęerine göre hedef verilerin elde edilmesinin ardından lineer sensör karakteristięine ait veriler elde edilir. Lineer veri eęer analog deęerde kullanılacak ise řekil 2.4 ile gösterildięi gibi belleęin ardından lineer D/A çevirici eklenir. D/A çeviricinin çıkışında lineerleştirilmiş olan analog işaret kullanıcıya bildirilir. Mikrodenetleyici tabanlı ve benzeri uygulamalarda bellek çıkışı herhangi bir işleme tabi tutulmadan kullanılır[11].

řekil 2.4 ile gösterilen blok diyagram, aktif ve pasif devre elemanları ile sağlanabileceęi gibi tek bir mikrodenetleyicinin kullanılması ile de sağlanabilir. řekilde belirtilen A/D çevirici, D/A çevirici, ROM bellek gibi yapılar, mikrodenetleyicilerin içerisinde tümleşik olarak bulunmaktadır[11,13].

Yöntemin temel sakıncası, yüksek çözünürlükte lineerlik istenmesi durumunda görölmektedir. Sistem başarımından istenilen çözünürlük deęeri arttıkça, katlanılacak ROM bellek maliyeti aynı oranda artacaktır. Örneęin, analog gerilim deęerinin 8 bitlik sistemde 256 farklı seviyeye kuantalandıęı tasarımda katlanılacak olan tablo tutma maliyeti için ařağıdaki maddeler oluřturulmuřtur[11,13,17].

- a. Sensörden elde edilen her bir deęerin sayısala çevrilmesi sonucunda 1 byte büyüklüğündeki verilerden oluřan 256 byte sensör verisi
- b. Her biri, gerçek sayı (double) cinsinden 16 bit (2 byte) veri uzunluęundaki 2 * 256 byte uzunluęunda referans sensör verileri
- c. Toplamda ise 768 byte veri alanı

Sensörün analog gerilim çıkış deęerini sayısala çevirmek için 8 bitten daha yüksek deęerli n bit çözünürlükle işlem yapılmak istendięi durumda, referans tablosunda tutulabilecek deęer sayısı, 2^n deęerine kadar çıkabilecektir. Tutulan sensör çıkış deęerleri, 1 byte ile ifade edilen en büyük deęer olan 256 deęerinden büyük olacaęı için sensör verileri en azından gerçek deęer (double) türünde deęişkenlerde tutulacaktır. Her bir gerçek deęerin bellekte kapladığı alan 2 byte (16 bit) olmaktadır.

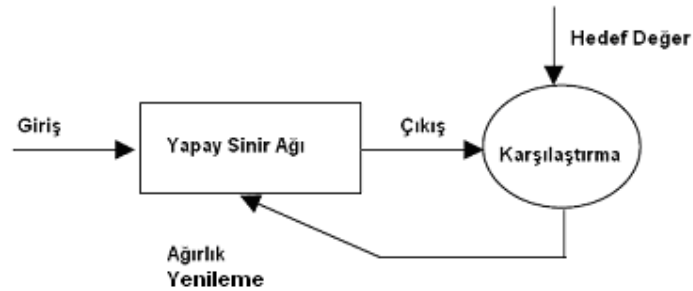
Bu durumda sensör çıkış değerlerinin 2^n farklı seviyeye kuantalandığı yapıda sensör verileri için $(2 \text{ byte} * 2^n)$ büyüklüğünde bellek alanı kullanılacaktır. Bunun yanında her bir sensör değerine karşılık tabloya konulacak gerçek verinin bellekte tutacağı yer $(2 \text{ byte} * 2^n)$ olur. Tablo tutma yöntemi ile toplam olarak bellekte kullanılan toplam alan ise $(4 \text{ byte} * 2^n)$ olur.

Analizlerden görülebileceği gibi sistemden istenilen n bit çözünürlük değeri arttıkça kullanılacak bellek alanı artmaktadır. Mikrodenetleyicilerin yapısında kullanılan bellek miktarı oldukça kısıtlıdır. Genel olarak mikrodenetleyicilerdeki bellek büyüklükleri 1 Kbyte ile 64 Kbyte arasında olmaktadır. Tasarımcının kullanacağı bellek miktarı arttıkça katlanılacak mikrodenetleyici maliyeti artmaktadır. Üretim adedi milyonları bulan ürünler için bahsedilen maliyet, önemli parasal miktarlara ulaşmaktadır[11,13,17].

2.2.2 Yapay sinir ağlarının kullanılması

Karmaşık yapıdaki sorunların çözümünde aktif olarak kullanılmasının ardından yapay sinir ağları, hayatımızda kullandığımız birçok uygulamada donanımsal ve yazılımsal tasarımlarda kullanılmaya başlanmıştır[4].

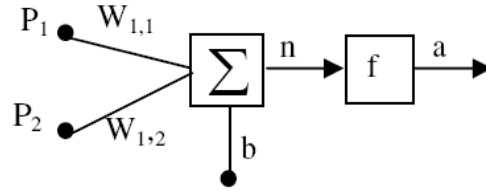
Yapay sinir ağları, genellikle sensörden elde edilen verinin karakteristiğinin modellenemediği uygulamalarda kullanılmaktadır. Yapay sinir ağlarında var olan mevcut yapı, kullanılacak sisteme göre kendisini adapte ederek lineerleştirme amacına yönelik tasarıma ulaştırmaktadır. En genel anlamda yapay sinir ağı ile oluşturulan yapı, seçilen etkinleştirme fonksiyonları nedeniyle non-lineer yapıdadır. Ağın non-lineer yapısı ile ağ girişine gelen non-lineer sensör istenilen hata düzeyi hassasiyetinde lineerleştirebilmektedir[5].



Şekil 2.5: Yapay Sinir Ağının Genel Yapısı

Şekil 2.5 içeriğinde yapay sinir ağı sisteminin genel olarak yapısı verilmiştir. Yapay sinir ağının temelinde birbirine belirli kurallarla bağlı nöronlar vardır. Nöronlar

arasındaki her bir bağlantının kendi aralarında ağırlıkları vardır. Ağırlıklar, yapay sinir ağının eğitim aşamasında, yapay sinir ağının çıkışı ile hedef değer arasındaki fark kabul edilebilir seviyenin altına gelinceye kadar yenilenir. Ağırlık yenileme işlemi sonunda yapay sinir ağı eğitilmiş olur. Eğitilmiş olan yapay sinir ağı, girişine uygulanan değerlerle, eğitim aşamasında oluşturulan nöronlar arası bağlantıların ağırlıkları ve sistem için kullanılan etkinleştirme fonksiyonunun türüne göre çıkış verir[1-3].



Şekil 2.6: Eğitilmiş yapay sinir ağında nöronun genel yapısı

Eğitilmiş yapay sinir ağında nöronun genel yapısı, Şekil 2.6 ile verilmiştir. Bu yapıda nöron çıkışının ifadesi Denklem 2.2 içerisinde verilen gibi olmaktadır.

$$a = f(P_1 W_{1,1} + P_2 W_{1,2} + b) \quad (2.2)$$

Denklem 2.2 içerisindeki ifadede belirtilen nöronların birleşmesi ile ağ oluşmaktadır. Nöronların birleşmesi ile Şekil 2.5 ile gösterilen çok katmanlı ağ yapıları oluşmaktadır[1-3].

Yapay sinir ağları ile oluşturulan yapılar, bilgisayarlı veri toplama sistemleri türünde sistemlerle uygulanabileceği gibi mikrodenetleyici tabanlı gömülü sistemlerde de uygulanabilir. Yapay sinir ağı parametreleri, mikrodenetleyici yazılımlarına aktarılır. Yazılımların çalışmasına yardımcı olması amacıyla yapay sinir ağlarının kullanıldığı alanlarda sisteme özel donanım gerekmemektedir. Mikrodenetleyicilerde bulunan mevcut yapılar, ağın çalışmasında yeterli olmaktadır. Mikrodenetleyici ile oluşturulan yapılarda ağ parametreleri, kullanılan yazılım ve derleyici kullanılarak mikrodenetleyiciye aktarılır[5].

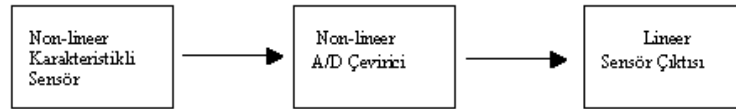
Yapay sinir ağları tabanlı analog olarak tasarlanan devrelerde istenilen sonucun elde edileceği yapay sinir ağı parametreleri, devre parametrelerine dönüştürülür. Sensör lineerleştirme amaçlı yapay sinir ağları parametrelerinin analog devrelerle gerçekleştirildiği devrelerin mikrodenetleyici tabanlı devrelere göre hız önyararı mevcuttur. Ancak, analog devrelerle gerçekleştirilen devrelerin tasarımının karmaşık

olması, devrelerin çevresel şartlardan olumsuz etkilenmeleri, esnek olmamaları yapay sinir ağlarının analog devre tasarımları ile uygulanmasının olumsuz yönlerini oluşturmaktadır[8,9].

Yapay sinir ağları ile oluşturulan yapılara ileride daha ayrıntılı olarak değinilecektir.

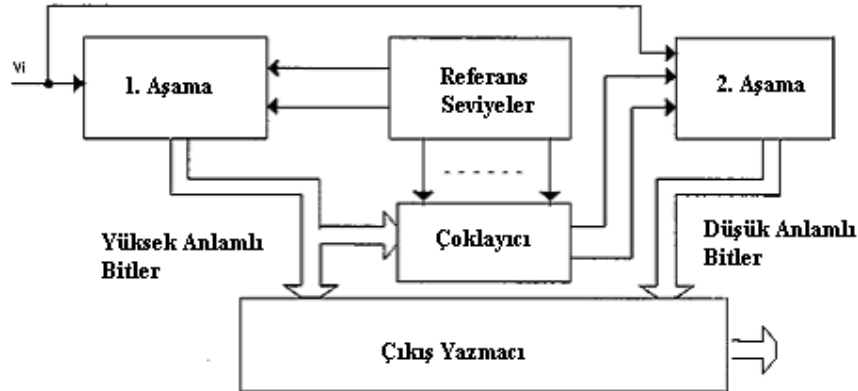
2.3 Lineer Olmayan ADC Kullanılması Yöntemi

Entegre devrelerle yapılan tasarımlar için maliyeti düşük tutmak en önemli parametrelerden birisidir. Buna paralel olarak tasarımın kapladığı alanın küçük tutulması, belirtilen isteğin bir uzantısı olmaktadır. Sayısal devrelerle gerçekleştirilen lineer karakteristikli sensörün sağlanması için non-lineer ADC kullanımı, tasarım için kullanılan alanın azaltılması amacına yönelik çalışmalardandır[6,7]. Lineer olmayan ADC yapıları ile oluşturulan lineerleştirme sistemlerinin blok diyagramı, Şekil 2.7 içeriğinde görülebilir.



Şekil 2.7: Lineer olmayan A/D Çevirici ile yapılan sensör lineerleştirme yapısı blok diyagramı

A/D çeviricilerle ilgili genel yapı Şekil 2.8 ile gösterilmiştir. Lineer olmayan A/D çeviricilerin lineer A/D çeviricilerden temel farkı, yapıda kullanılan gerilim referanslarının oransal olmasıdır[7]. Örnek olarak üstel karakteristiğe sahip bir sensör için Şekil 2.8 içerisindeki yapıda görülen çoklayıcının referans seviyeleri logaritmik olarak ayarlanmalıdır.



Şekil 2.8: Lineer olmayan A/D Çevirici Yapısı[7].

Sensör lineerleştirme amaçlı olan bir tasarımda lineer olmayan A/D çevirici kullanılması ile çıkış yazıcısının çıkışında lineer sensör karakteristiği elde edilir. Böylelikle lineerleştirmeye özel fazladan devre bloğu kullanmadan istenilen amaca uygun lineer sensör karakteristiği elde edilir[7,18].

Lineer olmayan A/D çeviricinin referans seviyeleri farklı yöntemlerle elde edilebilir. Uygulamada yukarıda belirtilen analitik tersine alma yöntemine uygun seviye belirlemenin yanında kesikli lineer yaklaşıma uygun tasarım ölçütlerine göre de belirlenebilmektedir. Zuza ve Carlosena'nın makalelerinde belirtildiği üzere kesikli lineer yaklaşıma göre referans seviyelerin belirlendiği tasarımlarda yapı daha basit olarak sağlanabilmektedir. Bununla beraber, yukarıda belirtilen kısmi parçalı lineer yaklaşımın analog devrelerde uygulanmasına ilişkin tasarımlarda görülen sakıncaları, lineer olmayan A/D çevirici tasarımında da barındırmaktadır[18].

3. YAPAY SİNİR AĞLARI

Sistemlerin doğrusal olmayan karakteristiğe sahip olmaları özelliğini göz önünde bulundurarak, sistemleri en az hata ile modelleme yollarından birisi yapay sinir ağları kullanarak modelleme yapmaktır[8]. Bu özelliğinden dolayı yapay sinir ağları, eğri uydurma alanında da yoğunlukla kullanılabilir. Yapay sinir ağları tabanlı yöntemlerin kullanılması ile sisteme ait giriş çıkış verilerinden yararlanılarak, doğrudan olmasa da dolaylı olarak sisteme ait giriş çıkış ilişkisini ortaya koyan fonksiyon elde edilebilir[4].

En genel anlamıyla yapay sinir ağları, beynin bir işlevi yerine getirme yöntemini modellemek için tasarlanan bir sistem olarak tanımlanabilir. Bir başka ifade ile söz konusu sistemin mevcuttaki öğrenme süreçleri ve beyindeki sinirsel faaliyetlerle ilgili yürütülen varsayımlar sonucunda oluşturulan, var olan verilerden öğrenme adı verilen süreç ile elde edilen bilgilerden yeni veriler hakkında fikir yürütebilen analitik tekniklerdir[8,9].

Yapay sinir ağının işlevi beyin ile özdeşleştirilirse, ağın temel işlem birimleri nöron olarak adlandırılmaktadır. Nöronların birbiri ile bağlanması ile katmanlı bir özellik kazandırılır. Nöronlar arasındaki her bir bağlantının ağırlığı vardır ve her bir bağlantı, bu ağırlıklar ile nicelendirilir. Yapay sinir ağları, pratikte donanımsal olarak elektronik devre parametreleri ile ve yazılımsal olarak gerçekleştirilebilmektedir[8,9].

Yapay sinir ağları, farklı alanlarda birçok soruna istenilen nitelikte sonuçlar sağlamıştır[1-3]. Bu alanlardan başlıcaları aşağıda verilmiştir:

- a. Modelleme ve Kontrol (Modeling and Control)
- b. Robotik (Robotics)
- c. Mali Uygulamalar (Financial Applications)
- d. Konuşma Tanıma (Speech recognition)
- e. İleriye Yönelik Görüşler (Vision)
- f. En Uygun Değeri Bulma (Optimization)

- g. Uzman Sistemler (Expert Systems)
- h. Zaman Serileri Tahmini (Time Series Prediction)
- i. İşaret İşleme (Signal Processing)
- j. Sınıflandırma Sorunu (Classification problem)

3.1 Yapay Sinir Ağlarının Temel Özellikleri

Yapay sinir ağlarının uygulandığı uygulamalardaki başarısının temelleri, yapısının paralel dağılmış olmasından, söz konusu sistemi öğrenebilme ve çeşitli tekniklerle sistem hakkında genelleme yapabilmesinden kaynaklanmaktadır. Buradaki genellemeden kastedilen, ağa eğitim ya da test aşamasında verilmeyen girişler için de ağın uygun çıkış üretmesidir. Belirtilen üstün özelliklerle yapay sinir ağları karmaşık sorunların çözümünde kullanışlı hale gelmektedir. Günümüzde var olan birçok uygulamada yapay sinir ağları aşağıda belirten özellikler nedeniyle etkin olmuş ve birçok alanda uygulanır hale gelmiştir[1].

3.1.1 Doğrusal olmama

Yapay sinir ağları, doğrusal olmayan işlem birimleri olan hücrelerden meydana gelmektedir. Böylelikle hücrelerin birleşimi ile oluşan yapı da doğrusal değildir. Bu yapısı ile yapay sinir ağları doğrusal olmayan karmaşık sorunların çözümünde etkin olmuştur[1-3].

3.1.2 Öğrenme

Yapay sinir ağlarının en temel özelliklerinden birisi, ağın çevresinden öğrenme kabiliyetine sahip olması, öğrenme sonucunda da mevcut başarımını arttırabilmesidir. En temel olarak öğrenme, ağın yapısında bulunan serbest parametrelerini, hedef sisteme ait benzetim (simülasyon) parametrelerine göre uydurduğu süreçtir. Yapay sinir ağlarının hedeflenen şekilde davranış gösterebilmesi için, ağda yapılacak ayarlamaların istenilen amaca uygun olması gerekmektedir. Bu hedefe ulaşabilmek, hücreler arasında oluşturulacak bağlantıların doğru yapılmasına ve bağlantıların olması gereken güce sahip olmasına bağlıdır. Yapay sinir ağlarının yapısının karmaşık olması nedeniyle belirtilen hücreler arası bağlantılar ve bu ağırlıkların her birinin taşıdığı güç önceden planlı olarak verilemez. Bundan dolayı, yapay sinir ağları, hedeflenen karakteristiği gösterecek biçimde ilgili sorundan alınan

eğitim verileri yolu ile öğrenme işlemini yapar ve buradan elde edilen bilgilere göre parametrelerini güncelleyebilmektedir[1-3].

3.1.3 Genelleme

Yapay sinir ağları, istenilen sorunla ilgili öğrenme işlevini yerine getirdikten sonra eğitim sırasında kendisine verilmeyen girişler için de istenilen tepkiyi üretebilir. Örneğin, karakter tanıma amaçlı bir yazılım için bozuk karakterin girilmesi sonucunda eğitimi yapılmış ağ, bozuk karakterin olması gereken düzeltilmiş halini oluşturabilir. Burada olduğu gibi eğitimi yapılmış yapay sinir ağı, eğitim süreci içerisinde verilmeyen giriş işaretleri için de sistemden beklenen davranışı gösterebilir[1-3].

3.1.4 Uyarlanabilirlik

Yapay sinir ağı, ilgili sorunda sonradan görülecek değişikliklerin niteliğine göre ağının yapısında bulunan ağırlıklarını tekrar ayarlayabilir. Belirli bir sorunun çözümü için oluşturulmuş olan yapay sinir ağı sorundaki değişimlere göre tekrar eğitilebilir ya da değişimler sürekli gerçekleşiyor ise gerçek zamanda da yapılan eğitime devam edilebilir. Belirtilen özelliklerin neticesinde yapay sinir ağı, uyarlamalı örnek tanıma, işaret işleme, sistem tanıma ve denetim gibi konularda etkin olarak kullanılabilir[1-3].

3.1.5 Hataya karşı tolerans

Yapay sinir ağları, nöron adı verilen birimlerin çeşitli kombinasyonlarda birbirine bağlı olmasından oluştuğu için paralel dağılmış bir yapıya sahiptir. Ağın elde ettiği bilgi, ağın her tarafına dağılmıştır. Buna göre eğitilmiş bir ağ içerisinde bazı bağlantıların ya da hücrelerin etkisiz hale gelmesi, ağın olması gereken bilgiyi üretmesini önemli ölçüde etkilemez. Bundan dolayı var olan geleneksel yöntemlere göre yapay sinir ağlarının hatayı en aza indirme derecesi oldukça yüksektir[1-3].

Yapay sinir ağları ile modelleme yapmak çeşitli şekillerde olabilmektedir. Modellemeler,

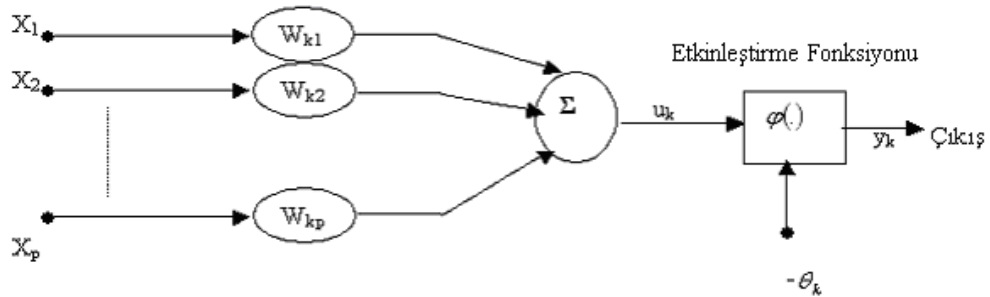
- a. Ağın yapısı,
 - i. İleri beslemeli(feedforward) ağlar
 - ii. Geri beslemeli(feedback) ağlar

- iii. Simetrik ya da asimetrik ağırlık matrisli ağlar
 - iv. Sabit veya değişken ağırlık matrisli ağlar
- b. Ağda yer alan düğümlerin özellikleri,
 - c. Kullanılan eşik fonksiyonunun deterministik veya stokastik oluşu,
 - d. Düğüme sadece ikili ya da sürekli değerlerin uygulanabilmesi,
 - e. Eğitim veya öğrenme kuralları,
- olmak üzere sınıflandırılabilirler.

3.2 Nöron Yapısı

Yapay sinir ağlarının temel işlem birimi nöronlardır. Nöronların bağlantılar arası ağırlıkları, toplayıcı ve etkinleştirme fonksiyonu olmak üzere üç temel bileşeni vardır. Bu maddelerle ilgili model Şekil 3.1 ile belirtilmiştir. Bu şekle göre k numaralı nörona ait j indisli girişe gelen x_j giriş işareti, w_{kj} ağırlığı ile çarpılır. Ağırlık değerleri, pozitif ya da negatif olabilmektedir[1].

Nöronun yapısında görülen toplayıcı, giriş işaretlerinin her birinin bağlantı ağırlıkları ile çarpılmasının ardından yapılan toplama işlemini ifade eder. Bu işlem ayrıca lineer birleşme olarak da adlandırılmaktadır[1].



Şekil 3.1: Temel nöron modeli

Nöronun yapısında görülen etkinleştirme fonksiyonunun görevi, nöron çıkışının şiddetini sınırlamaktır. Literatürde, sıkıştırma fonksiyonu olarak da bilinmektedir. Genel olarak normleştirilmiş biçimdeki çıkış değerleri $[0, 1]$ ya da $[-1, 1]$ aralığındadır. Etkinleştirme fonksiyonu lineer seçilirse, bu durumda yapı Adaptif Lineer Birleştirici adını alır[2].

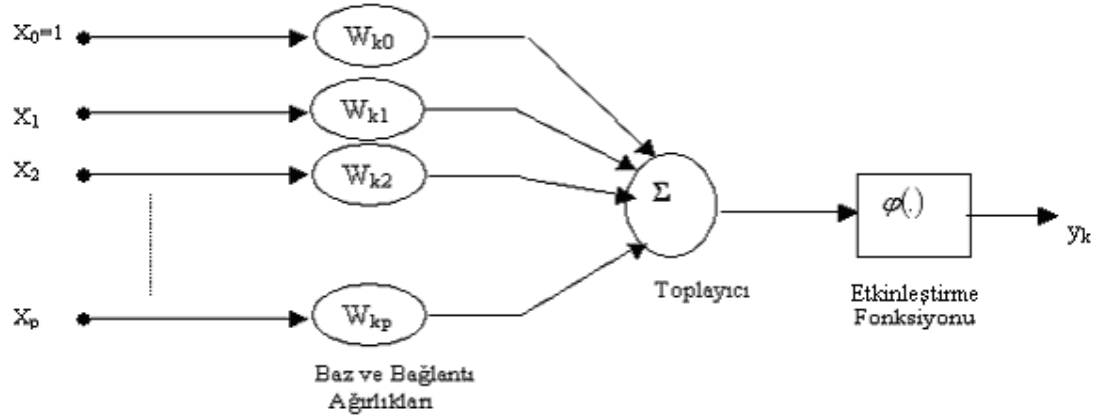
Modele Şekil 3.2 ile gösterildiği gibi eşik değeri uygulandığı durumda uygulanan eşik değeri, etkinleştirme fonksiyonuna gelen net girişi azaltır. Bu değeri negatif seçilerek net giriş artırılırsa baz değeri adını alır[1-3].

Nöron tarafından temsil edilen fonksiyon $x_1, x_2, \dots, x_p$ giriş işareti, k indisli nöronun bağlantısal ağırlıkları $w_{k1}, w_{k2}, \dots, w_{kp}$ olmak üzere nöronda görülen matematiksel ifadeler, aşağıdaki denklemler şeklinde ifade edilebilir.

$$u_k = \sum_{j=1}^p \omega_{kj} x_j \quad (3.1)$$

$$y_k = \varphi(u_k - \theta_k) \quad (3.2)$$

Genel olarak ifadede geçen baz değeri, sabit değerli $x_0=1$ girişine karşılık düşen $w_{k0} = b_k$ bağlantı ağırlığı, eşik değeri ise sabit değerli $x_0 = (-)1$ değerine karşılık düşen $w_{k0} = \theta_k$ bağlantı ağırlığı olarak değerlendirilir. Uygulamalarda genelde baz gösterilimi kullanılmaktadır. Baz değerinin kullanıldığı nöron yapısı Şekil 3.2 ile verilmiştir[1-3].



Şekil 3.2: Bazın kullanıldığı nöron modeli

Etkinleştirme fonksiyonunun seçimi konusunda farklı seçenekler bulunmaktadır. Belirtilen seçenekler, ilgilenilen uygulama alanına göre değişebilmektedir. Buna göre uygulamalarda etkinleştirme fonksiyonunun seçiminde sıklıkla tercih edilen fonksiyonlar; sigmoid fonksiyonu, gausiyen fonksiyonlar, signum fonksiyonları, tanh fonksiyonudur[1-3].

Uygulamamızla ilgili kullanılacak fonksiyonlara ileride değinilecektir.

3.3 Ağ Mimarileri

Yapay sinir ağlarında ağ değişkenleri, bağlantılar arasındaki ağırlıklardır. Yapay sinir ağı mimarisi ise söz konusu ağın sahip olduğu düğüm sayısı ve bu düğümlerin birbiriyle bağlantı biçimi tarafından belirlenir. Tüm ağın söz konusu olduğu durumda ağın giriş – çıkış karakteristiği, düğümlerdeki fonksiyonlar ve bağlantı ağırlıklarınca belirlenir[1-3].

3.3.1 Tek katmanlı cebrik ağlar

Yapay sinir ağlarının kullanıldığı uygulamalarda, ağın girişi ile çıkışı arasında doğrusal olmayan ve statik bir ilişki vardır. Belirtilen ilişkinin yapısının oluşturulması için nöronlar çeşitli seviyeler halinde gruplar oluşturmakta ve katmanlı yapılar ortaya çıkmaktadır. Belirtilen yapılarda, aynı katmanda yer alan nöronlar arasında bağlantı bulunmamakta, fakat her bir nöronun bir önceki ve bir sonraki katmanda yer alan nöronlarla bağlantısı bulunmaktadır[1-3].

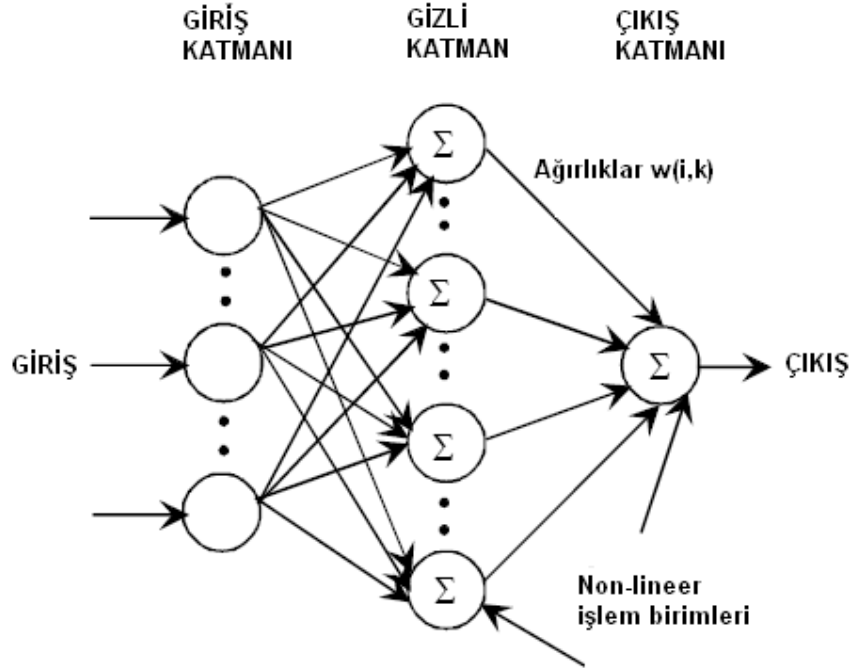
Katmanlı yapıların en temel hali, bir giriş ve bir çıkış katmanından oluşan tek katmanlı cebrik ağlardır. Ağın cebrik olması, ağda yer alan tüm bağlantıların ileriye doğru yapıyor olmasından kaynaklanmaktadır. Tek katmanlı denmesinin nedeni ise giriş katmanının gerçekte katman olarak algılanmamasından kaynaklanmaktadır. Çünkü giriş katmanında herhangi bir hesaplama yapılmamaktadır. Katman olma niteliği, ağın o bölümünde hesaplama yapılıyor olunması ile kazanılmaktadır[1-3].

Tek katmanlı ağ yapısı içerisinde de ağ parametreleri, yapılan eğitim sonucunda elde edilir. Bu tür ağların başarılı olabilmesi, eğitim kümesinde kullanılan verilerin lineer ayrıştırılabilmesi durumunda başarılı olacaktır. Diğer durumlarda ağın istenilen başarıyı yakalaması beklenmemelidir. Bu durumda eğitim algoritması, istenilen ortalama karesel hata değeri kadar yakınsamayacaktır[1].

Tek katmanlı ağların yeterli kalmadığı durumlar için çok katmanlı ağların kullanımı önerilmektedir[1-3].

3.3.2 Çok katmanlı cebrik ağlar

Uygulamalarda en fazla karşılaşılan yapay sinir ağı mimarisi, çok katmanlı cebrik ağlardır. Bu ağ tipi, en temel yapıda bir giriş katmanı, bir çıkış katmanı ve en az bir gizli katmandan oluşan statik ağıdır. Gizli katmanların rolü, giriş katmanı ile çıkış katmanı arasında yol oluşturmaktır. Ağa bir veya daha fazla gizli katmanın eklenmesi ile ağın daha yüksek seviyede istatistiksel seviyeyi sağlaması sağlamış olmaktadır. Gizli nöronların yüksek seviyedeki istatistiksel bilgiyi sağlamaları ile giriş katmanının büyük olması doğru orantılıdır. Hedef sistem için ulaşılması istenen ağ karakteristiğine göre her bir katmanda birden fazla nöron bulunabilir. Böylelikle yüksek dereceli istatistiksel karakteristikler elde edilebilir. Çok katmanlı ağ yapısı ile ilgili örnek, Şekil 3.3 ile verilmiştir[1-3].



Şekil 3.3: Çok katmanlı ağ yapısı[19]

Çok katmanlı yapay sinir ağları ile değinilmesi gereken bir başka nokta, çok katmanlı ağın tam bağlantılı ya da kısmi bağlantılı ağ olmasıdır. Eğer Şekil 3.3 ile belirtildiği gibi bir katmanda bulunan tüm nöronlar, bitişikte bulunan katmandaki nöronlarla bağlantılı ise söz konusu ağ tam bağlantılı, eğer bilinçli olarak bazı bağlantılar yapılmamış ise bu durumda söz konusu ağ kısmi bağlantılı ağ adını almaktadır. Ağın kısmi bağlantılı ya da tam bağlantılı yapılarla oluşturulması, ağın yapacağı işlem yükünü değiştirmektedir. Eğer hedef sistem hakkında bir takım bilgiler mevcut ise bu

durumda bilinçli olarak kullanılacak ağın kısmi bağlantılı ağ olması daha uygundur; çünkü bu durumda ağın yapacağı eğitim zamanı kısaltılmış olacaktır[1].

3.3.2.1 Çok katmanlı ağlara ilişkin denklemsel ifadeler

Çok katmanlı ağlara verilecek isimler farklı şekillerde olabilmektedir. Bunlardan birisi, m nöron girişli, birinci gizli katmanında n, ikinci gizli katmanında p ve çıkış katmanında q nöronu olan ağ yapısı m–n–p–q ağı olarak bilinmektedir[1].

Bir adet gizli katman içeren bir ağ yapısı için giriş ile çıkış arasındaki ifade aşağıdaki denklemlerdeki gibi gösterilebilir[1].

$$y = W\phi(Vx + b)$$

$$y_i = \sum_{r=1}^{n_h} \omega_{ir} \phi \left(\sum_{j=1}^m V_{rj} X_j + b_r \right) \quad (3.3)$$

$$i = 1, 2, \dots, l$$

Denklem 3.3 te $X \in R$ girişi, $y \in R$ çıkışı, $\phi(.)$ amaca göre seçilecek lineer olmayan bir fonksiyonu, $W \in R^{l \times m}$ çıkış katmanına ilişkin bağlantı matrisini, $V \in R^{n_h \times m}$ giriş katmanına ilişkin bağlantı matrisini, $b \in R^{n_h}$ hücrelerin baz değer vektörünü, n_h gizli katmandaki nöron sayısını göstermektedir[1].

İki adet gizli katman içeren bir nöron için aşağıdaki denklemler oluşur.

$$y = W\phi(V_2\phi(V_1x + b_1) + b_2) \quad (3.4)$$

$$y_i = \sum_{r=1}^{n_{h2}} \omega_{ir} \phi \left(\sum_{s=1}^{n_{h1}} v_{rs}^2 \phi \left(\sum_{j=1}^m v_{sj}^1 x_j + b_s \right) + b_r \right) \quad i=1,2,\dots,l \quad (3.5)$$

Denklemdeki $W \in R^{l \times n_{h2}}$ matrisi, çıkış katına ait bağlantı matrisi, $V_2 \in R^{n_{h2} \times n_{h1}}$ ikinci gizli katmana ilişkin bağlantı matrisi, $V_1 \in R^{n_{h1} \times m}$ birinci gizli katmana ilişkin bağlantı matrisi, $b_2 \in R^{n_{h2}}$ ikinci gizli katmana ait eşik değer vektörü, $b_1 \in R^{n_{h1}}$ birinci gizli katmana ait eşik değer vektörüdür. L katmanlı bir ağın giriş katmanı katman tanımına girmediği için belirtilen ağ L-1 adet gizli katmana sahiptir. L katmanlı ağı göstermek için aşağıdaki gösterimler kullanılmaktadır:

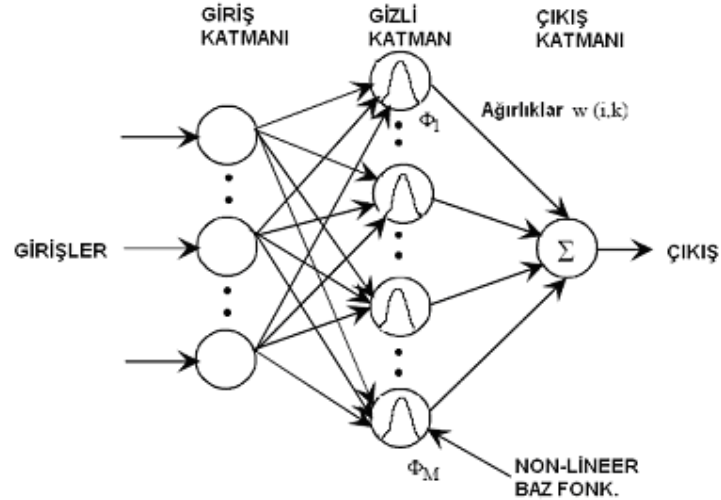
$$x_i^l = \phi(\xi_i^l) \quad (3.6)$$

$$\xi_i^l = \sum_{j=1}^{N_l} \omega_{ij}^l x_j^{l-1} \quad (3.7)$$

Denklemlerdeki l , katman indeksini, N_l bulunulan katmandaki nöron sayısını, x_j^{l-1} bulunulan l . katmandaki nöronların çıkışını göstermektedir. Sisteme ait baz değerleri, 1 değerli girişe karşılık düşürülen ağırlık olarak ağırlık matrisine yerleştirilir[1-3].

3.3.3 Radyal tabanlı ağlar

Radyal tabanlı ağları, çok girişli çok çıkışlı ağlar olarak da tanımlamak mümkündür. Radyal taban fonksiyonları ile oluşturulan yapılar genellikle 3 tane birbirinden farklı katmandan oluşmaktadır. Bu katmanlardan ilki giriş katmanıdır. İkinci katman, radyal taban fonksiyonlarından oluşan gizli katmandır. Son katman ise girişte görülen modelin istenilen forma dönüştürülmüş halidir. Giriş katmanından gizli katmanda yer alan her bir nörona yapılan dönüşüm non-lineerdir. Gizli katmandan çıkış katmanına yapılan dönüşüm lineerdir. Radyal tabanlı ağlarda tek gizli katman ve bu gizli katmanda n_h gizli düğüm içermektedir[1-3]. Radyal tabanlı ağların genel yapısı, Şekil 3.4 ile belirtildiği gibi gösterilebilir.



Şekil 3.4: Radyal taban fonksiyonunun genel yapısı[19]

Radyal tabanlı fonksiyonun çok katmanlı ağ yapılarından temel farkı, radyal tabanlı ağların bir adet gizli katman bulundurmalarına rağmen çok katmanlı ağ yapılarında kullanılan gizli katman sayısı birden fazla olabilmektedir[1].

Gizli katmanda bulunan her bir gizli nörondaki etkinleştirme fonksiyonu, giriş vektörü ile gizli katmandaki nöronun merkez noktası arasındaki öklidyen normu

hesaplar. Radyal taban fonksiyonlarının çıkış ile girişi arasındaki ifade aşağıdaki denklemlerle ifade edilir.

$$y_k(x) = \sum_{j=1}^M w_{kj} \varphi_j(x) + w_{k0} \quad (3.8)$$

Denklem 3.8 ile gösterilen M, gizli katmandaki nöron sayısını, k çıkış düğümünün indeks parametresini, φ_j ise baz fonksiyonunu ifade eder. Gausiyen baz fonksiyonlarının kullanıldığı durum, Denklem 3.9 ile ifade edilir.

$$\varphi_j(x) = \exp\left(-\frac{\|x - \mu_j\|^2}{2\sigma_j^2}\right) \quad (3.9)$$

Denklem 3.9 daki x değişkeni, x_i elemanlarından oluşan giriş vektörünü, μ_j baz fonksiyonun merkezini göstermektedir. Gausiyen baz fonksiyonları normalize edilmemiştir[1-3].

Radyal tabanlı fonksiyonları yapay sinir ağlarında kullanmanın en büyük önvararı, cebrik ağların yeniden eğitilmesi aşamasında görülür. Bu yapı ile gizli katmanlardaki hücreler, giriş uzayının sadece belirli bölgelerine duyarlı olmaktadır. Bu durumda eğitim aşamasının sisteme verdiği zahmet azalır.

3.4 Geriye Yayılım Algoritması

Geriye yayılım algoritmasına konu olan yapılarda iki türlü veri girişi vardır. Bunlar:

- Algoritmanın uygulanacağı sistemden alınan çıkış,
- Algoritmanın uygulanacağı sistemin uyması istenen hedef çıkış

Geriye yayılım algoritmasının hedef sisteme uygulanması ile amaçlanan, non-lineer karaktere sahip sistemden alınan çıkışları, sistemin uyması istenen hedef çıkış değerlerine istenilen seviyede hata değeri ile yaklaştırmaktır. Bu amaca ulaşabilmek için geriye yayılım algoritması, ileride açıklanacak yöntemlerle sistem parametrelerini ayarlamaktadır. Bir başka ifade ile sisteme verilen girişleri ve hedefte ulaşılması istenen girişleri kullanarak yapıyı öğrenmekte ve bu yapıya uygun sistem parametrelerini oluşturmaktadır[1-3].

3.4.1 Geriye yayılım algoritmasının temel kavramları

3.4.1.1 Hata kavramı

Algoritmada n numaralı iterasyondaki j numaralı nörona ait hata sinyali aşağıdaki denklemde olduğu gibi tanımlanmaktadır.

$$e_j(n) = d_j(n) - y_j(n) \quad (3.10)$$

Bu tanımdaki j numaralı nöron, çıkış düğümünü belirtmektedir. Buna göre elde edilen hataya ait anlık enerjilerin toplamı aşağıdaki gibi hesaplanmaktadır.

$$\xi(n) = \frac{1}{2} \sum_{j \in C} e_j^2(n) \quad (3.11)$$

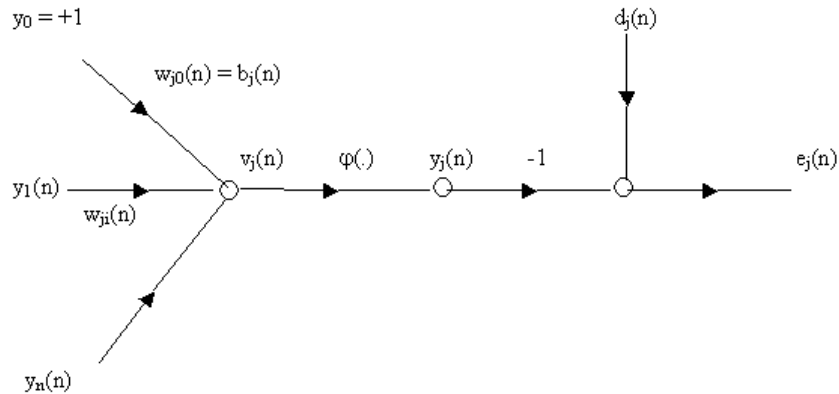
Denklemdeki C seti, ağıın çıkış katındaki tüm nöronları kapsamaktadır. N sayısı, eğitim setindeki örnek sayısını ifade etsin. Bu durumda ortalama hata enerjisi Denklem 3.12 ile gösterilen şekildeki formu almaktadır.

$$\xi_{av}(n) = \frac{1}{N} \sum_{n=1}^N \xi(n) \quad (3.12)$$

Ortalama hata enerjisi terimi, özellikle katsayıların dinamik olarak istenilen değere getirilmesi uygulamalarında kullanılmaktadır. Bu tür uygulamalar için ortalama hata enerjisi teriminin, belirlenen minimum değer aralığında kalması kontrol edilmektedir[1-3].

3.4.1.2 Nörona ait yapılan işlemler

Çıkış nöronuna ait akış diyagramı Şekil 3.5 ile gösterilmiştir.



Şekil 3.5: j numaralı çıkış nöronuna ait işaret akış diyagramı

Şekil 3.5 diyagramında görüleceği gibi j numaralı nöron, solunda bulunan nöronlar tarafından beslenmektedir. Nöron j yapısında bulunan ve etkinleştirme fonksiyonunun girişinde bulunan $v_j(n)$ ifadesi, Denklem 3.13 teki gibi elde edilir.

$$v_j(n) = \sum_{i=0}^m w_{ji}(n) y_i(n) \quad (3.13)$$

Burada m değeri nörona ait toplam giriş sayısını, $w_{ji}(n)$ büyüklüğü nöron i den nöron j ye sinaptik ağırlığı, n numaralı iterasyonda nöron çıkışındaki fonksiyon işareti $y_j(n)$ ifadesi Denklem 3.14 ile gösterildiği gibidir.

$$y_j(n) = \varphi_j(v_j(n)) \quad (3.14)$$

Şekil 3.5 te $d_j(n)$ büyüklüğü, arzu edilen lineer çıkışı, $y_j(n)$ büyüklüğü nöron çıkışını ve $e_j(n)$ ise nöron çıkışındaki hatayı ifade etmektedir. İşaret akış diyagramından da görülebileceği gibi hata ifadesi arzu edilen çıkış ile nöron çıkışı arasındaki farka eşittir. Geriye yayılım algoritmasında yer alan ağırlık düzeltme faktörü $\Delta w_{ij}(n)$ ifadesi Denklem 3.15 içeriğindeki gibi verilmektedir[1-3].

$$\Delta w_{ij}(n) = \eta \delta_j(n) y_i(n) \quad (3.15)$$

Denklem 3.15 içeriğinde yer alan η çarpanı öğrenme hızı olarak belirtilmektedir. Bu parametrenin kullanımına daha ayrıntılı olarak ileride değinilecektir.

$\delta_j(n)$ terimi yerel gradient olarak adlandırılmakta ve ilgilenilen nöronun konumuna göre değişik ifade edilmektedir. Buna göre, ilgilenilen j numaralı nöronun çıkış düğümü olduğu durumda yerel gradient ifadesi Denklem 3.16 içeriğinde belirtilmiştir.

$$\delta_j(n) = e_j(n) \varphi'(v_j(n)) \quad (3.16)$$

Bu ifadede $e_j(n)$ büyüklüğü daha öncede belirtildiği gibi arzulanan çıkış ile nöron çıkışı arasındaki farka eşittir. $\varphi'(v_j(n))$ büyüklüğü de uygulanan etkinleştirme fonksiyonunun türevidir[1-3].

Gizli katman düğümündeki j numaralı nöronun yerel gradient ifadesi, Denklem 3.17 ile verilmiştir[1-3].

$$\delta_j(n) = \varphi'(v_j(n)) \sum_k \delta_k(n) w_{kj}(n) \quad (3.17)$$

3.4.1.3 Hesaplama yönleri

Geriye yayılım algoritmasının uygulanmasında geri ve ileri olmak üzere iki farklı hesaplama yönü bulunmaktadır[1-3].

İleri yönde hesaplama aşamasında, sinaptik ağırlıklar ağ boyunca değişmemektedir. Ağdaki fonksiyonlar nörondan nörona hesaplanmaktadır. Hesaplamanın ileri yönü, giriş vektörünün uygulanması ile ilk gizli katmandan başlar. Çıkış katında bu katmandaki her bir nöron için hataların hesaplanması ile sona erer[1-3].

Geri yön ise çıkış katından ağ boyunca hataların bir sola iletilmesi ile başlar ve yinelemeli olarak her bir nöron için yerel gradientin hesaplanması ile devam eder. Denklem 3.15 ile verilen delta kuralına göre sinaptik ağırlıkların değişmesi sağlanır. Daha önce belirtildiği gibi yerel gradient, çıkış katındaki hata terimi ile etkinleştirme fonksiyonunun türevinin (non-lineerlik ifadesi) çarpımına eşittir. Diğer katlardaki yerel gradient ifadesi için de 3.17 numaralı denklem kullanılır. Rekürsif hesaplamalar katmandan katmana ağı her tarafına yayılarak, gerekli olan sinaptik ağırlıklardaki değişiklikler yapılır[1-3].

3.4.1.4 Etkinleştirme fonksiyonu

Önceki ifadelerden de görülebileceği gibi yerel gradientin hesaplanabilmesi için nörona ait olan etkinleştirme fonksiyonu $\phi(.)$ nin bilinmesi gerekmektedir. Etkinleştirme fonksiyonu için seçilen ifadenin her noktada türevlenebilir, sürekli olması gerekmektedir. Uygulamalarda en sık kullanılan sürekli non-lineer etkinleştirme fonksiyonu, sigmoid fonksiyonudur. Bu fonksiyonun iki türü vardır[1-3].

Bu fonksiyonlardan Lojistik fonksiyonunun genel ifadesi aşağıda verilmiştir.

$$\phi_j(v_j(n)) = \frac{1}{1 + \exp(-av_j(n))} \quad a > 0 \text{ ve } -\infty < v_j(n) < \infty \quad (3.18)$$

Bu fonksiyonun uygulanması ile çıkış işaretinin genliği $0 \leq y_j \leq 1$ olmaktadır. Elde edilen ifade ile lojistik fonksiyonunun türevi, Denklem 3.19 ile verilmiştir.

$$\phi_j'(v_j(n)) = ay_j(n)[1 - y_j(n)] \quad (3.19)$$

Türev teriminin maksimum değeri, $y_j(n)$ ifadesinin 0,5 değeri için elde edilir. $y_j(n)$ büyüklüğünün minimum değeri ise $y_j(n) = 0$ ve 1 ile elde edilir. Buradan da görülebileceği gibi sigmoid etkinleştirme fonksiyonunda sinaptik ağırlıkların en büyük değişimi, fonksiyon işaretlerinin orta değerlerinde olduklarında gerçekleşmektedir. Rumelhart' a göre bu özellik dolayısıyla öğrenme algoritması stabiliteyi sağlayabilmektedir[1-3].

Sigmoidal non-lineerlik için kullanılan bir diğer fonksiyon, hiperbolik tanjant fonksiyonudur. Hiperbolik tanjant fonksiyonunun genel olarak ifadesi, Denklem 3.20 ile gösterilmiştir. Denklemde, a ve b terimleri sabittir. Fonksiyonun türevi Denklem 3.21 ile gösterilmiştir[1-3].

$$\varphi_j(v_j(n)) = a \tanh(bv_j(n)) \quad (a,b,j>0) \quad (3.20)$$

$$\varphi_j'(v_j(n)) = \frac{b}{a} [a - y_j(n)][a + y_j(n)] \quad (j>0) \quad (3.21)$$

3.4.1.5 Öğrenme oranı

Ağ ağırlıklarında istenilen değerlere yaklaşma için kullanılan parametrelerden birisi de öğrenme oranı parametresidir. Öğrenme oranı parametresi η ne kadar küçük ise bir iterasyondan diğerine ağırlıklarda gerçekleşen değişim o oranda az olacaktır. Bu durumda değişimin büyük olduğu yapılarda hedef değer olması gereken zamandan daha geç sürede yakalanacaktır. Buna karşın, bir iterasyondan diğerine gerçekleşen değişimler küçük ise bu durumda öğrenme hızının küçük seçilmesi tercih sebebi olacaktır[1-3].

Öğrenme parametresi büyük seçildiğinde ise sistemde bir iterasyondan diğerine gerçekleşen öğrenme hızı eskisine göre daha hızlı artmaktadır. Fakat, bu parametrenin gerektiğinden büyük seçilmesi, istenmeyen durumlara yol açabilmektedir. Parametrenin gerektiğinden büyük seçilmesi durumunda uygulamada en fazla karşılaşılan sorun, sistemin kararlı olmaktan çıkmasıdır. Bu durumda sistem değerlerinde istenmeyen osilasyonlar gözlenebilmektedir[1-3].

Öğrenme hızını artırmak için önerilecek yöntemlerden birisi, delta kuralına momentum terimi eklemektir. Bu durumda Denklem 3.15, Denklem 3.22 halini alır.

$$\Delta w_{ji}(n) = \alpha \Delta w_{ji}(n-1) + \eta \delta_j(n) y_i(n) \quad (3.22)$$

Denklemdaki α , momentum sabitidir. Genellikle pozitif değer almaktadır[1-3].

Momentum sabitinin sinaptik ağırlıkları uydurması üzerindeki etkisinin görülebilmesi için Denklem 3.22, zaman domeininde yazılmalıdır. Bu durumda Denklem 3.23 elde edilmektedir.

$$\Delta w_{ji}(n) = \eta \sum_{t=0}^n \alpha^{n-t} \delta_j(t) y_i(t) \quad (3.23)$$

$$\Delta w_{ji}(n) = -\eta \sum_{t=0}^n \alpha^{n-t} \frac{\partial \xi(t)}{\partial w_{ji}(t)} \quad (3.24)$$

Denklem 3.23 ve Denklem 3.24 ten yararlanılarak aşağıdaki izlenimler elde edilebilir:

- a. Son durumdaki $\Delta w_{ji}(n)$ değeri üstel olarak ağırlıklandırılmış zaman serilerini temsil eder. Zaman serilerinin yakınsaklığının sağlanması için momentum sabitinin değer aralığı $0 \leq \alpha \leq 1$ olmalıdır. Momentum sabiti 0 olduğu durumda geriye yayılım algoritması momentum olmadan çalıştırılır. Bunun yanında, momentum sabiti pozitif ya da negatif olabilir. Ancak pratikte negatif değerli momentum katsayısı ile karşılaşmamaktadır.

- b. Denklem 3.24 ile verilen kısmi diferansiyel değeri $\frac{\partial \xi(t)}{\partial w_{ji}(t)}$ ardı ardına iterasyonlarda aynı işareti taşıyorsa bu durumda üstel ağırlıklandırılmış toplam $\Delta w_{ji}(n)$ değeri genlikte büyür. Bu nedenle $w_{ji}(n)$ ağırlığı büyük değerle ayarlanır. Momentum sabitinin ifadeye eklenmesi ile işarettaki inişi hızlandırır ve stabilizeye kavuşmayı sağlar.

- c. Denklem 3.24 ifadesindeki kısmi diferansiyel ifade, yinelemeli iterasyonlarda farklı işaretleri taşıyorsa $\Delta w_{ji}(n)$ değeri genlikte küçülür. Bu durumda $w_{ji}(n)$ ağırlığı küçük değerde ayarlanır. Momentum sabitinin algoritmaya eklenmesi ağırlıkların değişiminde osilasyon etkisini önleyerek ağırlıklarda kararlılık etkisi gösterir.

Momentum sabitinin algoritmaya eklenmesi, ağırlıkların tekrar eldesinde küçük etki gösterebilir. Fakat, algoritmanın öğrenme davranışı üzerinde büyük etkiye sahiptir[1-3].

Geriye yayılım algoritmasının sağlanmasında öğrenme oranı parametresi η sabit olarak alınmıştır. Gerçekte bu sabit η_{ji} olarak tanımlanabilmektedir. Bazı durumlarda öğrenme oranı parametresinin bağlantı odaklı olması önerilebilmektedir. Bu parametrenin değişken yapılması ile bir çok ilginç sonuç elde edilebilmektedir[1-3].

Bazı durumlarda da algoritmadaki sinaptik ağırlıkların değerlerinin, ağırlıkların uydurulması aşamasında belirlenmesi ve ardından sabit olması sağlanabilir. Bunun için öğrenme oranı parametresinin 0 alınması gerekecektir[1].

3.4.1.6 Eğitimin ardışıl ve grup modları

Geriye yayılım algoritmalarında öğrenme, birçok eğitim setlerinin uygulanması sonucu oluşmaktadır. Öğrenme süreci, sinaptik ağırlıkların ve kutuplama seviyelerinin ağı stabilize edinceye ve ortalama hata karelerini toplamının belirli bir minimumun altına yakınsamasına kadar devam eder[1-3].

Verilen bir eğitim takımı için geriye yayılım algoritması iki şekilde gerçekleştirilebilir. Bunlar, ardışıl mod ve takım tarzlarıdır.

Geriye yayılım öğrenmesinde, ardışıl tarza aynı zamanda on-line, stokastik tarz adı da verilmektedir. Bu tip uygulamalarda, sinaptik ağırlıkların güncellenmesi, her bir eğitim örneğinin sunulması sonunda gerçekleştirilmektedir. Örneğin, N adet eğitim örneği bulunan yapıyı ele alalım $((X_1, d_1), (X_2, d_2), \dots (X_N, d_N))$. Bu durumda ağda önce (X_1, d_1) örneği ele alınır ve ağırlıklar buna göre belirlenerek, (X_2, d_2) ve en son (X_N, d_N) örneği ele alınır. Her birinde oluşturulan ağırlıklar farklı olmaktadır[1-3].

Takım tarzında ise ağırlık güncelleme, tüm eğitim örneklerinin ağda uygulanmasından sonra elde edilmektedir[1-3].

Ağ parametresine ait sinaptik ağırlıkların, sistemin çalışması sırasında da sürekli olarak yenilediği uygulamalarda takım tipinde olan uygulamalara tercih edilmektedir. Bunun en önemli nedeni, bu tür uygulamaların sistemde her bir sinaptik bağlantı için daha az veri saklama amaçlı yer ihtiyacı gerektirmesidir. Bunun yanında ağa örnekler rasgele bir şekilde sunulabilir. Bu durumda örnekten örneğe göre ağırlıklarda yapılan adaptasyonlar sistemin doğasını stokastik hale getirmektedir. Böylece sistemin lokal minimumu elde etmesi kolay olmayabilmektedir[1-3].

Bunun yanında ardışıl mod sisteminin stokastik yapısı, algoritmadaki yakınsaklık ifadelerinin elde edilmesinde zorluklar oluşturmaktadır. Buna karşın, takım tarzının kullanılması, gradient vektörünün daha keskin tahmin edilmesini sağlamaktadır. Yerel minimuma yakınsama basit yöntemlerle garanti edilmektedir[1-3].

Eğitim verilerinin gereğinden fazla olması ve bu verilerin birbirinin kopyasının olması durumunda takım modu yerine ardışıl mod önerilmektedir[1-3].

Özet olarak; ardışıl tarz, öğrenme algoritması üzerindeki sakıncalarına rağmen oldukça yaygın olarak kullanılmaktadır[1]. Bunun en önemli nedenleri:

- a. Algoritmanın uygulama için basit olması
- b. Büyük ve zor sorunlara etkili çözümler sağlamasıdır.

3.4.1.7 Algoritmayı durdurma kriteri

Geriye yayılım algoritması her zaman etkin bir yakınsama göstermeyebilir. Bu özellikten dolayı genel olarak ortaya konabilecek iyi tanımlanmış bir durdurma kriteri bulunmamaktadır. Fakat, ağırlık yenilemelerini bitirebilecek, uygulamalara göre özel kriterler geliştirilebilmektedir.

Ağırlık yenilenmelerini durdurma amaçlı kullanılabilecek yöntemlerden birisi, hata yüzeyinin lokal ya da global minimuma düşmüş olmasıdır. Bu yöntemle göre gradient vektörünün öklidyen normu, değeri küçük olan gradient eşiğine yaklaştığında geriye yayılım algoritması yakınsamış olur[1].

Belirtilen kritere göre ağırlık vektörü w^* minimum olsun. w^* minimum olması için ağırlık vektörü w için katsayılarla ait hatalar yüzeyinin gradient vektörü $w = w^*$ için $g(w) = 0$ olmalıdır.

Oluşturulabilecek bir başka durdurma kriteri, hata ölçüsü Denklem 3.12 ile belirtilen $\xi_{av}(w)$ değeridir. Durdurma kriteri için uygun durum $w = w^*$ noktası için $\xi_{av}(w)$ fonksiyonunun durağan olmasıdır. Bu kriter için her bir ağırlık yenileme safhasında ortalama hata kareleri değerine bakılır. Hesaplanan ortalama hata karelerindeki değişim, daha önceden belirlenen bir değerden küçük olduğu zaman geriye yayılım algoritmasının yakınsadığından bahsedilmektedir[1-3].

Karesel ortalama hata değerindeki değişimin değeri %0.1 ile 1 arasında olduğu zaman küçük olarak kabul edilebilmektedir. Bazı durumlarda da küçük olarak kabul

edilen hata değışiminin değeri %0.01 olabilmektedir. Bu kriterin seçilmesi ile katlanılması kabul edilen hata değerin altında hata ile ağ katsayıları elde edilebilmektedir.

3.4.2 Geriye yayılım algoritmasının yazılımsal ifadesi

Önceki bölümde belirtilenlere paralel olarak geriye yayılım algoritması için yapılması gerekenler aşağıdaki maddelerde verildiğı gibi özetlenebilir[2].

- Yapay sinir ağına girişler uygulanır
- Ağın ağırlıkları rasgele seçilmiştir ve bu ağırlıklara göre ağın çıkışları alınır.
- Ağın çıkışı, olması gereken çıkışla karşılaştırılır.
- Ortalama karesel hata değeri hesaplanır.
- Hata değeri ağ boyunca geriye doğru hesaplanır.
- Ağırlık değışimleri, hataları azaltacak şekilde hesaplanır.
- Yukarıdaki tüm aşamalar, istenilen hata değeri elde edilene kadar sürdürülür.

Algoritmanın kaba hatları ile yazılımsal olarak ifadesi[2]:

- Rasgele seçilmiş ağırlıklarla başla
- While** (MSE istenilen değerden büyükse ve hesaplanabilir sınırlar aşılmamışsa)

do{

for { her bir x_p girişı için $1 \leq p \leq P$

Gizli düğüm girişlerini hesapla($net_{p,j}^{(1)}$);

Gizli düğüm çıkışlarını hesapla($x_{p,j}^{(1)}$);

Çıkış düğümündeki girişleri hesapla($net_{p,k}^{(2)}$);

Ağ çıkışlarını hesapla($o_{p,k}$);

$O_{p,k}$ çıkışı ile istenilen çıkış $d_{p,k}$ arasındaki hatayı hesapla;

Gizli düğümler ve çıkış düğümleri arasındaki ağırlıkları yeniden düzenle

$$\Delta w_{k,j}^{(2,1)} = \eta (d_{p,k} - o_{p,k}) \delta' (net_{p,k}^{(2)}) x_{p,j}^{(1)}$$

Giriş düğümleri ile gizli düğümler arasındaki ağırlıkları yeniden düzenle

$$\Delta w_{j,i}^{(1,0)} = \eta \sum_k \left((d_{p,k} - o_{p,k}) \delta'(net_{p,k}^{(2)}) w_{k,j}^{(2,1)} \right) \delta'(net_{p,j}^{(1)}) x_{p,i}$$

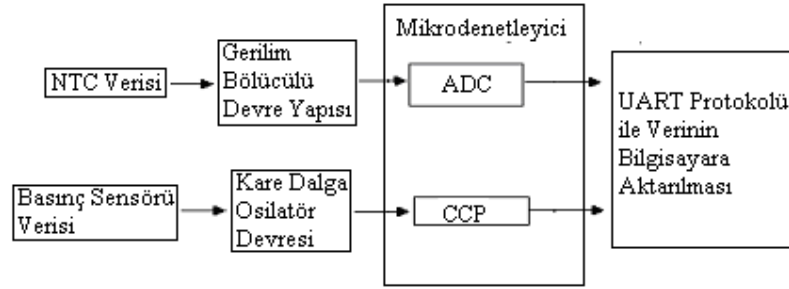
end for }

end while }

4. UYGULAMALAR İÇİN OLUŞTURULAN SİSTEMİN YAPISI

Mikrodenetleyici ile yapılan uygulamalar kapsamında oluşturulan sistem için elektronik devre tasarlanmıştır. Tasarlanan elektronik devrenin en genel blok diyagramı Şekil 4.1 ile verilmiştir. Buna göre sistem, iki tür sensör verisini elde etmeye ve değerlendirmeye göre kurulmuştur. Bu sensör türleri:

- Ortam sıcaklığına göre yapısındaki direnci değişen NTC sıcaklık sensörü,
- Girişindeki basınç değerine göre yapısındaki endüktans değeri değişen frekans çıkışlı basınç sensörü,



Şekil 4.1: Sistem için kullanılan devre blok diyagramı

Elektronik kartın girişine sensör verileri girilmektedir. Kart üzerinde, sensörleri okuyabilmek amacıyla ilgili mikrodenetleyici girişi ile kullanılan sensörün sisteme bağlandığı yer arasında analog devre yapıları kullanılmıştır. Analog devre yapılarının ayrıntıları, ilerleyen aşamalarda verilmiştir. Kullanılan devre yapıları:

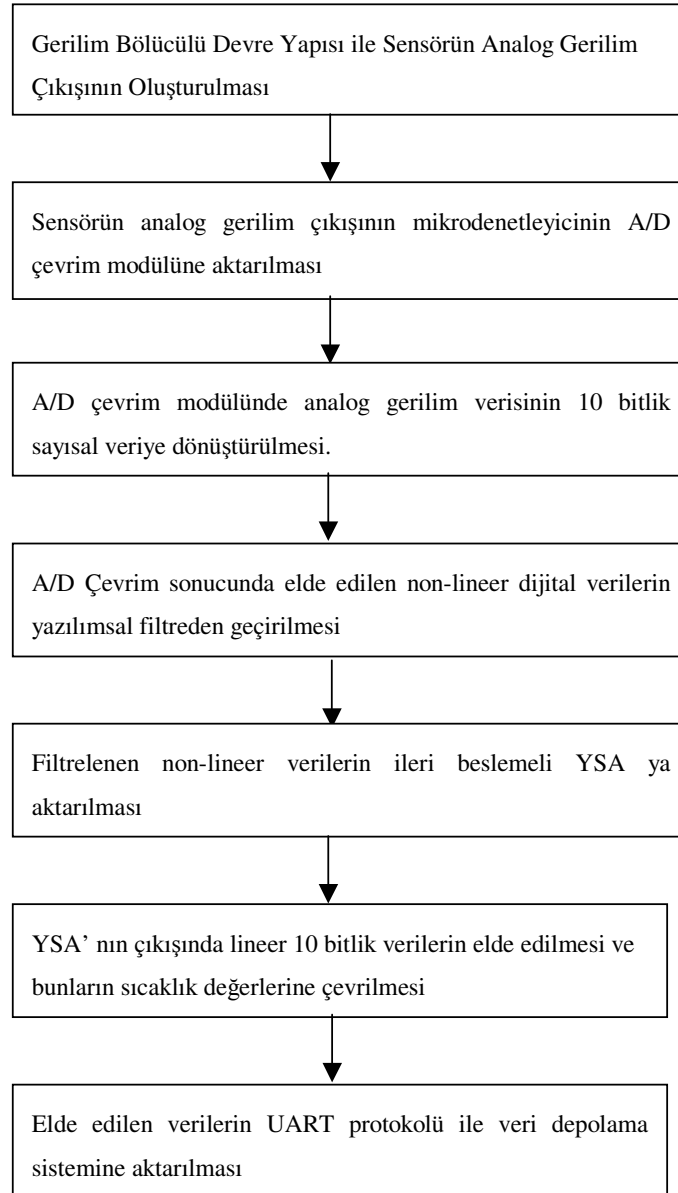
- NTC sıcaklık sensörünün direnç değerini gerilime çevirme amaçlı gerilim bölücülü devre yapısı
- Basınç sensörünün endüktans değerini baz alan kare dalga osilatör devresi

Oluşturulan sistemde yukarıda belirtilen analog devre yapılarının çıkışlarında, sensör verileri mikrodenetleyicinin okuyabileceği hale dönüştürülmüş olmaktadır. Şekil 4.1 de belirtildiği üzere mikrodenetleyicide sensör verilerini değerlendirmek için 10 bitlik ADC modülü ve 16 bitlik CCP modülü kullanılmıştır.

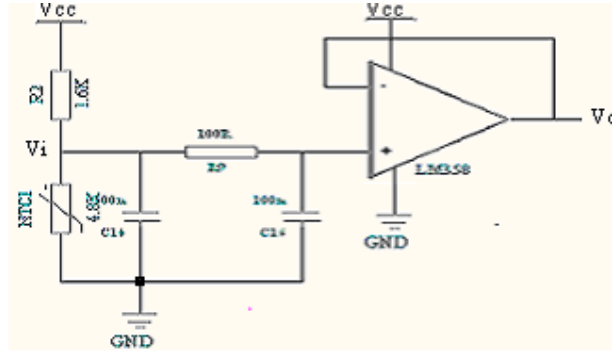
Sensörlerden elde edilen veriler, UART seri haberleşme protokolü ile veri depolama sistemine aktarılmakta ve kullanıcının bilgisine sunulmaktadır.

4.1 Tasarlanan Devrede NTC Sıcaklık Sensörü Okuma Yöntemi

Bu amaçla oluşturulan genel algoritma, Şekil 4.2 ile gösterilmiştir. Algoritmada belirtilen gerilim bölücülü devre yapısı, Şekil 4.3 ile verilmiştir.



Şekil 4.2: NTC Analog gerilim çıkışı okuma algoritması.



Şekil 4.3: NTC değerini okumak için tasarlanan analog devre yapısı.

Şekil 4.3 ile gösterilen devrede görülebileceği gibi NTC sıcaklık sensörü, NTC1 olarak belirtilen şekil ile temsil edilmiştir. NTC sıcaklık sensörü ile R_2 direnci bir gerilim bölücü devresi oluşturmaktadır. Tasarlanan yapı ile NTC nin devreye bağlı olduğu yerin toprağa göre gerilimi, NTC nin direnç değeri değiştiğinde farklı değer alması sağlanmaktadır. NTC sıcaklık sensörünün direnç değerine göre oluşan gerilim değerinin ifadesi Denklem 4.1 ile gösterilebilir.

$$V_i = R_s * \frac{V_{cc}}{R_s + R_2} \quad (4.1)$$

Denklemdeki V_{cc} , devrenin kutuplama gerilimidir. R_s ise anlık olarak sensörün aldığı direnç değeridir.

Devrede kullanılan C_{14} ve C_{16} kapasitörleri, gerek kaynaktan, gerekse dışarıdan gelebilecek gürültüleri süzme amaçlı olarak kullanılmıştır.

Tasarlanan devrede, gerilim bölücülü yapının sensör çıkış gerilimi değeri, mikrodenetleyicinin yapısında tümleşik olarak bulunan A/D çeviriciye aktarılmaktadır. Gerilim aktarılması aşamasında mikrodenetleyici ile gerilim bölücünün çıkışının arasına birim kazançlı işlemsel kuvvetlendirici kullanılmıştır. Birim kazançlı işlemsel kuvvetlendiricili yapının kullanılmasının amacı, işlemsel kuvvetlendiricinin çıkış empedansının küçük olmasından yararlanarak, mikrodenetleyicinin girişinden dışarı bakıldığında görülen empedansı küçültmektir. Mikrodenetleyicinin A/D çevirici girişinde kullanılan örnekleme ve tutma devresinin modeli Şekil 4.4 ile verilmiştir.

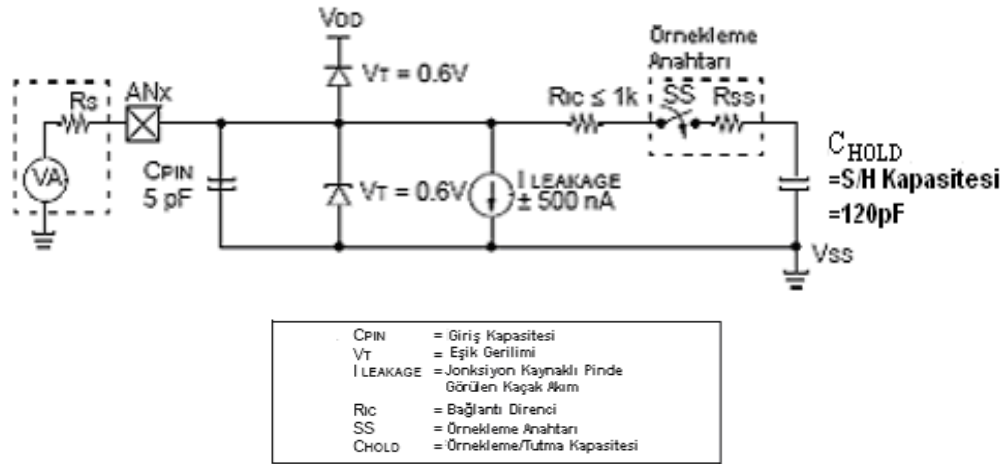
Mikrodenetleyicinin analog giriş modeline göre öncelikle uygulanan giriş gerilimi, analog giriş modelinde belirtilen blok diyagramda gösterilen C_{HOLD} tutma kapasitörünü doldurur. Yapıda gösterilen örnekleme ve tutma devresinin çıkışı, A/D

çeviricinin girişı olmaktadır. Bu şekilde alınan giriş gerilimi, mikrodnetleyiciye verilmektedir.

A/D çevrimin istenilen doğrulukta yapılabilmesi için C_{HOLD} kapasitörü, girişine gelen gerilimle tam olarak dolması gerekmektedir. Ancak, bu kapasitörün tam olarak dolması belirli koşullara bağlıdır. Bu koşullar:

- A/D çevirici girişine uygulanan kaynak empedansı
- İç yapıdaki örnekleme anahtarı empedansıdır.

Belirtilen nedenler ile C_{HOLD} kapasitörünün dolma zamanı etkilenmektedir.



Şekil 4.4: Mikrodnetleyici içerisinde tümleşik olarak bulunan A/D çeviricinin giriş modeli

A/D çeviriciye giriş için önerilen en büyük empedans değeri, $10k\Omega$ dur. Bu empedans değeri azaldıkça, C_{HOLD} kapasitörü daha kısa sürede dolacak, böylelikle sayısal verinin elde zamanı azalacaktır. Bu açıdan Şekil 4.4 ile gösterilen R_s giriş empedansının küçük tutulması, A/D çevrim işleminin hızı ile direk olarak ilişkili olmaktadır. Uygulamamızda, A/D çevrim süresini kısa tutmak, mikrodnetleyicinin A/D girişine uygulanan giriş empedansını küçültmek için işlemsel kuvvetlendiricinin çıkış empedansından yararlanılmıştır. Bu amaçlı olarak oluşturulan devre, Şekil 4.3 ile gösterilmiştir. Böylelikle, A/D çeviricinin girişinden görülen empedansın işlemsel kuvvetlendiricinin çıkış empedansı kadar olması sağlanarak, R_s empedansın değeri küçük tutulmuştur.

4.1.1 A/D çevrim için mikrodnetleyicide yapılması gereken işlemler

Mikrodnetleyici içerisinde temel olarak yapılması gereken işlemler, maddeler halinde aşağıda özetlenmiştir.

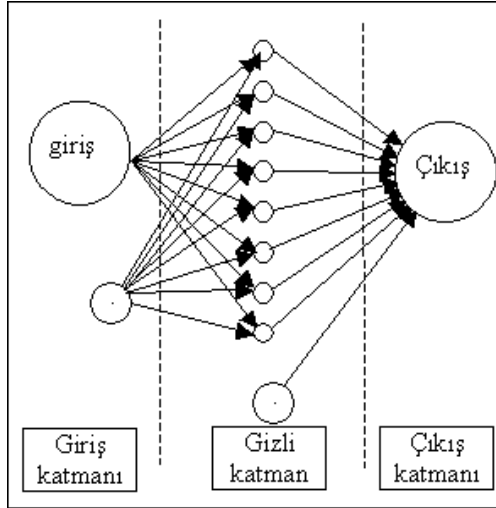
- a. A/D modül, isteklerin karşılanacağı biçimde ayarlanmalıdır:
 - i. ADCON1 ayarlanmalı
 - ii. A/D girişi olarak kullanılacak mikrodnetleyici girişi belirtilmeli
 - iii. A/D çevrim saati seçilmeli
 - iv. A/D modül açılmalı
 - v. Yukarıdaki maddelere göre ADCON0 ayarlanmalı
- b. Gerekli olan süre kadar beklenilmeli
- c. Çevrim başlatılmalıdır. Bunun için ADCON0 da yer alan GO/DONE bitine lojik 1 değeri verilmelidir.
- d. A/D çevrim bitene kadar beklenmelidir. Bunun için GO/DONE bitinin lojik 0 değerini alması kontrol edilebilir. Bu bitin lojik 0 değerini alması, A/D çevrim bittiği anlamına gelmektedir.
- e. A/D sonuç kaydedicileri (10 bitlik veri için ADRESH ve ADRESL) okunmalıdır.
- f. Bir bitin A/D çevriminin oluşturulması için geçen süre T_{AD} olarak belirtilmektedir. Bir sonraki çevrim başlamadan önce en azından $2T_{AD}$ süresi kadar beklenmesi gerekmektedir.
- g. Bekleme süresinin ardından çevrime tekrar geri dönülür.

Yukarıdaki ayarlamaların mikrodnetleyici yazılımına aktarılması ve mikrodnetleyicinin çalıştırılmasının ardından sensör için oluşturulan okuma devresinin çıkışındaki gerilim değeri her bir A/D çevrimde 10 bitlik sayısal veriye dönüştürülür. Elde edilen sayısal verilerde, giriş gerilimindeki değişimden, eleman toleranslarından, çevresel şartlardan vb. nedenlerden kaynaklanan anlık salınımlar görülebilmektedir. Bu salınımların verilere yansımaları engellemek için mikrodnetleyici yazılımının içerisinde ek işlem yapılmaktadır. Yapılan ek işlemde aşağıdaki aşamalar vardır:

- Her bir A/D çevrimde elde edilen veri 21 elemanlı dizi içerisinde atılır.
- Belirtilen boyuttaki dizi dolduktan sonra elde edilen veriler küçükten büyüğe sıralanır.
- Sıralanan dizideki verilerin ortancası bulunur.
- Verilerin medyan, değeri kullanılacak değer olarak alınır.

4.1.2 Elde edilen sayısal verilerin YSA ve tabloya aktarılması

Yukarıdaki yöntemle ek işleminden geçirilen sayısal veriler, artık yapay sinir ağının giriş verileridir. Bölüm 3 de belirtildiği gibi yapay sinir ağının parametreleri, bilgisayar üzerinde yapılan eğitim sonucu oluşturulan parametrelerdir. Bilgisayar üzerinde eğitim sonucu elde edilen yapay sinir ağı parametreleri, daha önceden yazılmış olan mikrodenetleyici yazılımına aktarılmaktadır. Mikrodenetleyici içerisinde yapay sinir ağı için yazılan programda, sadece ağı ileri yol aşaması girilmiştir. İleri yol aşamasını simgeleyen yapı Şekil 4.5 ile gösterilmiştir.



Şekil 4.5: İleri yol YSA yapısı

Belirtilen şekilde oluşturulan yapay sinir ağı yazılımına girilen kaynak kodlar ile non-lineer yapıda olan sayısal verilerin oluşturduğu sensör karakteristiği, lineer yapıdaki sayısal verilerin oluşturduğu sensör karakteristiğine dönüştürülür.

Lineer yapıda sayısal sensör verisinin oluşturulmasının ardından sayısal verilerin ilgili sensör birimine dönüştürülmesi işlemi yapılır. Bu amaçla lineer olduğu bilinen YSA çıkışına lineer bir formül girilerek her bir sayısal veriye karşılık ilgilenilen birimli sensör verisi elde edilir. Bu amaçla aşağıdaki yöntemler uygulanır:

- NTC analog çıkışına karşılık elde edilen sayısal 10 bitlik ADC verisi, eğitim aşamasında kullanılan sayısal verilere göre normalize edilir. Normalizasyon için kullanılan formül aşağıdaki gibidir.

$$Normalize_deg\ er = \frac{(sayisal_deg\ er - min_sayisal_deg\ er)}{(maks_sayisal_deg\ er)} \quad (4.2)$$

Normalize değer ifadesindeki “maks_sayısal_deger”, sayısala çevrilmiş sensör verilerinden, yapay sinir ağının eğitimi aşamasında eğitime giren en büyük sayısal değerdir. Benzer şekilde “min_sayısal_deger” ise yapay sinir ağının eğitimi aşamasında eğitime giren en küçük sayısal değerdir. Normalizasyon formülünde görülen “sayısal deger” ise o anda sensörden gelen ve 10 bit ADC değerine dönüştürülen sayısal veriyi ifade etmektedir.

- Bir önceki aşamada elde edilen normalize değeri sensör değerine çevirmek için aşağıdaki formül kullanılmaktadır.

$$Sensör\ değeri = y*(MY) + mY \quad (4.3)$$

Sensör değeri Denklem 4.3 formülünde belirtilen y, yapay sinir ağının 0 ile 1 arasında değer alan normalize çıkışı ifade eder. MY değeri, ağın eğitimi sırasında kullanılan en büyük referans sensör değerine karşılık gelmektedir. Formüldeki mY değeri ağın eğitimi sırasında kullanılan en küçük referans sensör verisine karşılık gelmektedir.

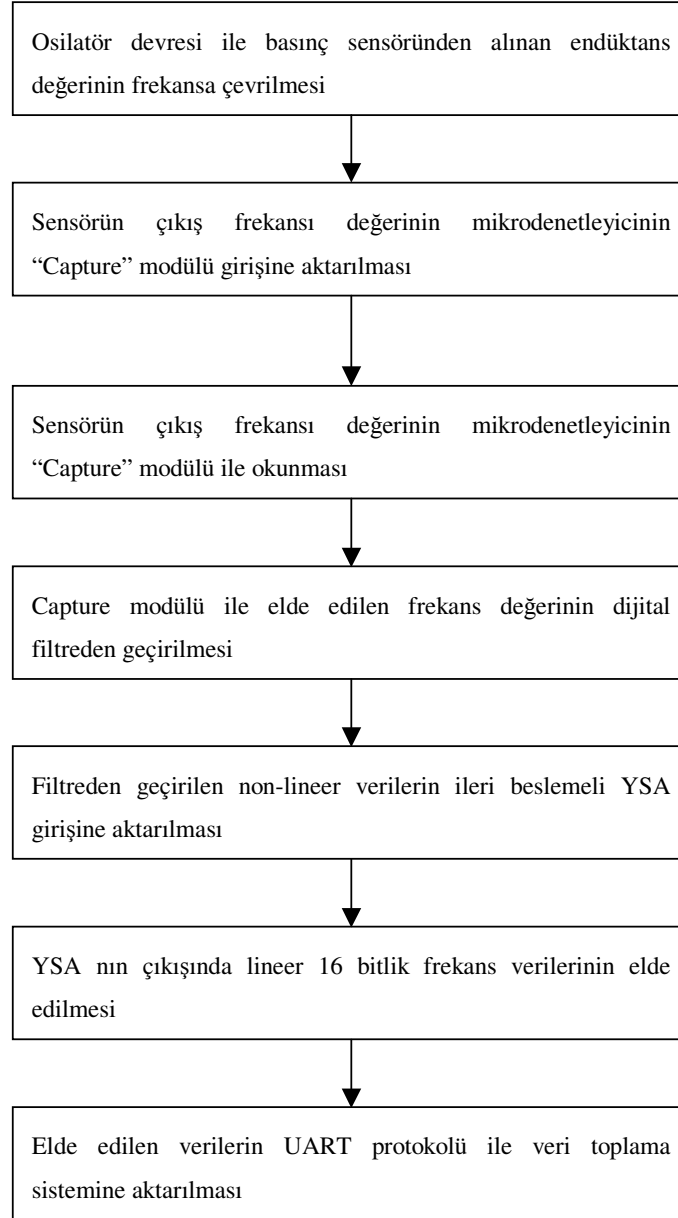
Tablo tutma yöntemi ile yapılan uygulamalar için oluşturulan sistemde, sayısal sensör verileri ile işlem yapılmaktadır. Sayısala dönüştürülmüş olan sensör verilerinden yararlanarak sensör dönüşüm tablosu oluşturulmuştur.

Sensör dönüşüm tablosunu oluşturmak için veri depolama sistemi ile toplanan referans veriler ve her bir referans veriye karşılık gelen sensör değerlerinden yararlanılır. Veri depolama sisteminde toplanan verilerle gelen her bir sensör değerine karşılık referans değerler oluşturulur. Ardından, içeriği sensör sayısal değerlerine karşılık referans veriler olan sensör dönüşüm tablosu kurulur. Oluşturulan yazılım ile sensörden gelen her bir veriye karşılık sensör dönüşüm tablosundan referans veri çekilmektedir. Çekilen veriler, hedef değer olarak kullanılmaktadır.

Uygulanan yöntemlerin başarımını görebilmek için UART seri haberleşme protokolü ile YSA çıkışı ve tablo tutma yöntemi çıkışı, referans verilerle birlikte veri depolama sistemine aktarılır. Veri depolama sisteminde toplanan veriler, ortalama karesel hata yönteminin uygulanması yolu ile değerlendirilmiştir. Değerlendirmelerin sonuçları, 5 ve 6. bölümlerinde verilmiştir.

4.2 Tasarlanan Devrede Frekans Çıkışlı Basınç Sensörü Okuma Yöntemi

Basınç sensörünü okumak için oluşturulan sistemin yapısı Şekil 4.6 içeriğinde verilmiştir. Belirtilen sensör çıkışını frekansa çevirmek için kullanılan devre yapısı Şekil 4.7 ile verilmiştir.

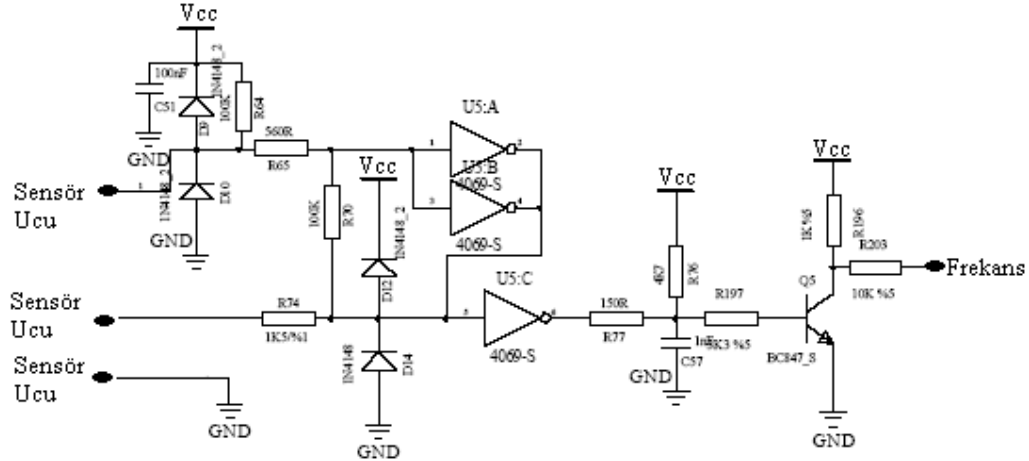


Şekil 4.6: Frekans çıkışlı basınç sensörü okuma algoritması

Basınç okumak için kullanılan sensör, çamaşır makinelerinde su seviyesini algılama amaçlı olarak kullanılmaktadır. Sensör, pasif devre elemanlarından değişken değerli

endüktans olarak kabul edilebilir. Temel çalışma prensibi olarak sensörün, girişinde algıladığı basınçla doğru orantılı olarak, sensörün endüktans değeri değişmektedir.

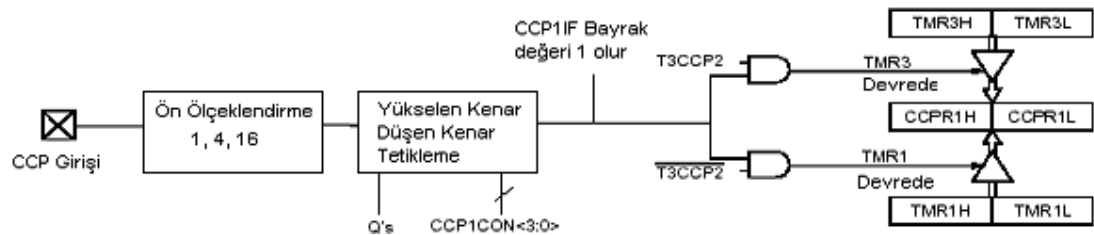
Frekans çıkışlı basınç sensörü, Şekil 4.7 ile gösterilen devrenin girişine bağlanmaktadır. Şekilde gösterilen devre, osilatör devresidir. Osilatör devresi çıkış düğümünde, frekansı sensör endüktans değeri ile orantılı kare dalga görülmektedir.



Şekil 4.7: Basınç sensörünü frekansa çevirmek için kullanılan devre yapısı

Sensörün endüktans değerine göre elde edilen frekans değeri, mikrodenetleyicinin yapısında tümleşik olarak bulunan CCP (Capture, Compare, PWM) modülü girişine iletilir. Devrede görülen V_{cc} , devrenin kutuplama gerilimidir. Bu amaçla devreye kutuplama gerilimi olarak 5V uygulanmıştır.

CCP modülü, devre çıkışında elde edilen frekansın okunması amacıyla kullanılır. Mikrodenetleyicide kullanılan CCP yazıcısı 16 bit hassasiyette frekans verisi okuyabilmektedir. Yapıdaki CCP modülü blok diyagramı, Şekil 4.8 ile belirtildiği gibidir.



Şekil 4.8: CCP modülü blok diyagramı

Mikrodenetleyicinin CCP modülüne gelen frekans verisini okuyabilmek için mikrodenetleyici üzerinde aşağıdaki ayarlamalar yapılmıştır.

- CCP modülü, isteklerin karşılanacağı biçimde ayarlanmalıdır:

- i. Frekansın okunması için CCP modülüne ilişkin kesme (interrupt) kaydedicileri istenilen doğrultuda ayarlanmalıdır.
 - ii. CCP1CON kaydedicisine, frekans okumaya uygun olarak değer atanmalıdır.
 - iii. CCP modülünün girişinin mikrodnetleyicinin hangi bacağından yapılacağı ile ilgili değer mikrodnetleyiciye yazılmalıdır.
 - iv. CCP modülüne giriş için seçilen bacak, giriş olarak yazılmalıdır.
 - v. CCP modülü devreye alma kaydedicisine, işlemin başlatılabilmesi için 1 değeri verilmelidir.
- b. Frekans saymak için mikrodnetleyici içerisinde hangi saatin kaynak olarak kullanılacağı belirlenmeli, bunun için gerekli ayarlama saat kaydedicilerinde yapılmalıdır.
- c. Frekansın sayılması aşamasında kenar tetiklemenin türü (yükselen kenar ya da düşen kenarda tetiklenme) ve kaç tetiklemelik periyotlarla sayımın yapılacağı(prescaler değeri) mikrodnetleyiciye yazılmalıdır.
- d. Çevrim başlatılmalıdır. Bu amaçla kesme yetki biti (interrupt enable bit) 1 değerini almalı ve kesme bayrak biti (interrupt flag bit) 0 değerine çekilmelidir. Bu işlemlerle aynı zamanda frekans sayma amaçlı kullanılacak olan saatin değerine sıfır atanmalıdır.
- e. Yükselen kenar tetiklemeli bir sistem için ilk yükselen kenarın algılanmasının ardından ikinci yükselen kenar algılanana kadar beklenir. İkinci yükselen kenar geldiğini algılamının yolu, CCP1IF (CCP kesme bayrak biti) bitinin kontrolü ile olur. Eğer bu bitin değeri 1 ise ikinci yükselen kenar algılanmış anlamına gelmektedir.
- f. CCP modülü blok diyagramından görülebileceği gibi CCP1IF bitinin mikrodnetleyici tarafından 1 değerine çekilmesi ile birlikte saat içerisinde bulunan 16 bitlik değer, CCPR1L ve CCPR1H kaydedicilerinde iki parça halinde yerleştirilir. CCPR1L kaydedicisinde düşük anlamlı 8 bit yer alırken, CCPR1H kaydedicisinde yüksek anlamlı 8 bit yer alır.

- g. Elde edilen 16 bitlik deęer, CCP mod l ne gelen kare dalga iřaretin periyodunun deęeridir. Mikrodenetleyiciye gelen frekans deęerinin  ğrenilebilmesi i in CCP kaydedicilerinde okunan deęerin  arpmaya g re tersi alınır. Bundan sonraki periyotlarda da frekans sayma iřleminin yapılabilmesi i in mikrodenetleyici tarafından 1 deęeri atanan CCP1IF biti, 0 deęerine  ekilir. İřlemlere d rd nc  adımdan devam edilir.

Sens r verisinin kullanıcı tarafından saęlıklı bir řekilde algılanabilmesi i in elde edilen sens r verilerinin periyodik olarak ortalamaları alınır. Bu ama la, her frekans okuma  evriminin ardından elde edilen sens r verileri, her biri 16 bitlik frekans verilerinden oluřan bir dizinin i erisine yerleřtirilir. Dizinin boyut sınırı kadar frekans verisinin alınmasının ardından dizideki t m veriler b y kten k   ęe doęru sıralanır. Bu ařamadan sonra dizinin ortasında kalan medyan deęeri bulunur. Ortanca deęeri, kullanılacak deęer olarak sens r yazılımının ilerleyen ařamalarına aktarılır.

4.2.1 Elde edilen verilerin YSA giriřine aktarılması

Bu ařamada, sens rden gelen frekans bilgisi okunmuř olmakta ve bundan sonra sens rden elde edilen frekans bilgisi lineerleřtirme y ntemine girdi olmaktadır.

Yapay sinir aęlarının anlatıldıęı B l m 3 i erięinde belirtildięi gibi aę parametreleri, bilgisayar  zerinde yapılan eęitim sonucu oluřturulan parametrelerdir. Eęitim i in anlık olarak iki t r veri toplanmıřtır. Bu veriler:

- a. Sens r verisi okumak i in oluřturulan osilat r devresinin  ıkıřında oluřan kare dalganın mikrodenetleyici tarafından  l  len frekansı
- b. Frekansın mikrodenetleyici tarafından  l  ld ę  anda referans basın   l  m cihazı olan manometrenin g sterdięi basın  deęeri.

Toplanan veriler ile bilgisayar  zerindeki eęitim ile yapay sinir aęı parametreleri oluřturulmuřtur. Eęitim sonucu elde edilen parametrelerle, yapay sinir aęının ileri yol ařaması oluřturulmuřtur. Aęın ileri yol ařaması mikrodenetleyici i in oluřturulan yapay sinir aęı kodunun i erisine girilmiřtir. Aęın ileri yol ařamasını simgeleyen yapı, řekil 4.5 ile g sterilmiřtir.

Mikrodenetleyici i erisine girilen aęın bařarımını g rebilmek i in UART protokol  ile mikrodenetleyici i erisinde yer alan aę  ıkıř bilgisi ve referans basın  deęerleri olan manometre deęerlerini i eren anlık sens r verileri toplanmıřtır.

5. SENSÖR VERİLERİNİ MİKRODENETLEYİCİDE İŞLENMESİ

Bölüm 4 içerisinde belirtilen yöntemlerle sıcaklık ve basınç sensörlerine ait veriler elde edilmiş, mikrodnetleyiciye aktarılmıştır. Aktarılan sensör verisi, çeşitli yöntemler kullanılarak işlenmiş ve istenilen büyüklüklere dönüştürülmüştür. Verinin işlenmesi için kullanılan yöntemler aşağıdaki gibi verilebilir.

- a. Tablo Tutma Yöntemi
- b. Çok Katmanlı Ağ Yapıları ile Oluşturulan YSA Yöntemi
- c. Radyal Taban Fonksiyonları ile Oluşturulan YSA Yöntemi

Uygulamaların yapılması esnasında oluşturulacak sensör karakteristiklerinin belirlenmesi amacıyla ilgilenilen sensöre ilişkin referans verileri ile sensörün anlık olarak elektriksel çıkış verileri toplanmıştır.

Basınç sensör karakteristiklerinin elde edilebilmesi için manometre kullanılmıştır. Manometreden elde edilen veriler, veri depolama sisteminden isteğe göre seçilen periyotlarda ani değer olarak kaydedilebilmektedir.

Sıcaklık sensörü karakteristiklerinin elde edilebilmesi için T tipi termokupl kullanılmıştır. T tipi termokupl, tam olarak sensör ile sıcaklık ölçümünün alındığı nokta üzerine yerleştirilmektedir. Termokupldan elde edilen veriler, isteğe göre seçilen periyotlarda veri depolama sistemine kaydedilebilmektedir.

Sensörlerin anlık elektriksel çıkış verilerinin toplanması için bilgisayar ile mikrodnetleyici arasında RS-232 protokolü ile haberleşen program kullanılmıştır. Belirtilen haberleşme programının kullanılması ile sensör verisinin her aşaması ile oluşturulan algoritmanın verileri gözlenebilmektedir. Algoritma çıkışında hesaplanan sensör verisi, istenilen periyotta bilgisayara aktarılabilir. Sensör verileri, belirtilen programın kullanılması ile birlikte her saniyede bir bilgisayara aktararak depolanmıştır.

Veri depolama için kullanılan sistemin genel yapısı Şekil 5.1 ile gösterilmiştir.



Şekil 5.1: Veri depolama sisteminin genel yapısı

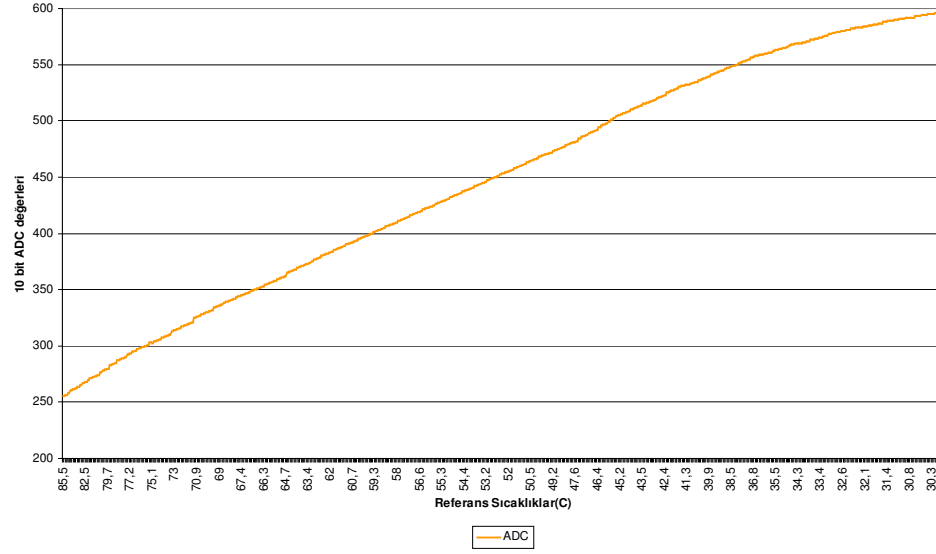
Yöntemlerin uygulanması için oluşturulan tasarımda, mikrodnetleyici olarak Microchip firması tarafından üretilen PIC 18F252 kullanılmıştır.

Seçilen mikrodnetleyicide, sıcaklık sensörünün gerilim çıkış değerini okuyabilmek için 10 bitlik ADC yapısı bulunmaktadır. ADC dinamik çalışma aralığı, 0 – 5V arasında seçilmiştir. Buna göre ADC kuantalama büyüklüğü, 4.88mV tur.

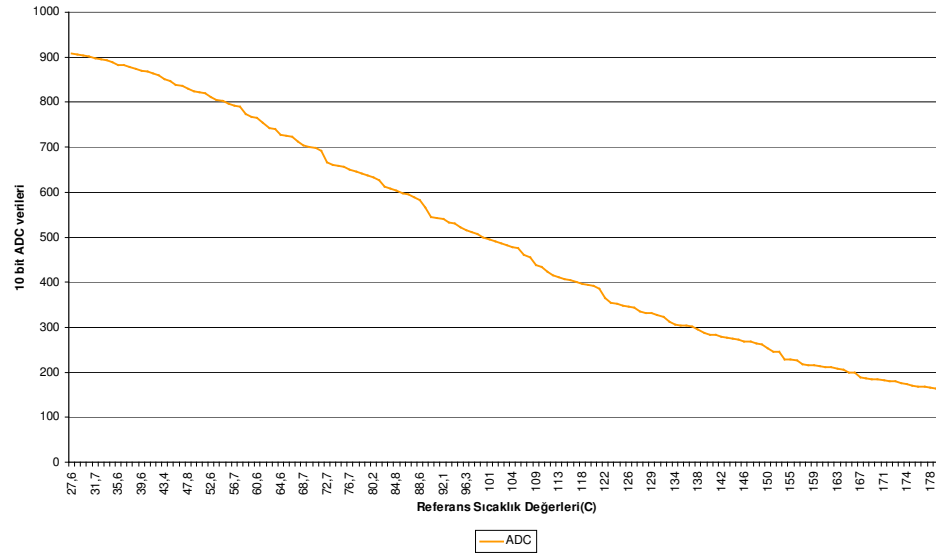
Basınç sensörünün çıkış değerine ilişkin frekans verisini okumak için mikrodnetleyici içerisinde gömülü olarak 16 bitlik CCP modülü tümleştirilmiştir. Bunların yanında, mikrodnetleyicinin uygulama için öne çıkan yönleri arasında; yeterli büyüklükte bellek alanı ihtiva etmesi, 40MHz değerine kadar çalışabilmesi ve yeterli sayıda I/O içermesi sayılabilir. PIC 18F252 mikrodnetleyicisi 32Kbyte program hafızasına sahip ve 25 adet I/O bacağı bulunmaktadır. Mikrodnetleyicinin özellikleri ile ilgili ayrıntılı bilgiler ekte bulunabilir.

Özellikle yapay sinir ağları tabanlı olanlar başta olmak üzere mikrodnetleyicide oluşturulan yazılımsal kodlarda, sensörden gelen verinin işlenmesi için matematiksel ifadeler yoğun olarak kullanılmaktadır. Matematiksel ifadelerin istenilen nitelikte derlenebilmesi amacıyla derleyici olarak “Hi-Tec PIC-18 C Compiler” kullanılmıştır. Derleyici, özellikle sigmoid fonksiyonu ile işlem yapıldığı zamanlarda kullanılan üstel ifadeler için 24 bit ya da istenildiği takdirde 32 bit hassasiyette işlem yapabilmektedir. Buna göre üstel ifadelerin kullanım aralığı -87.3365447506, +88.7228391117 arasında olabilmektedir.

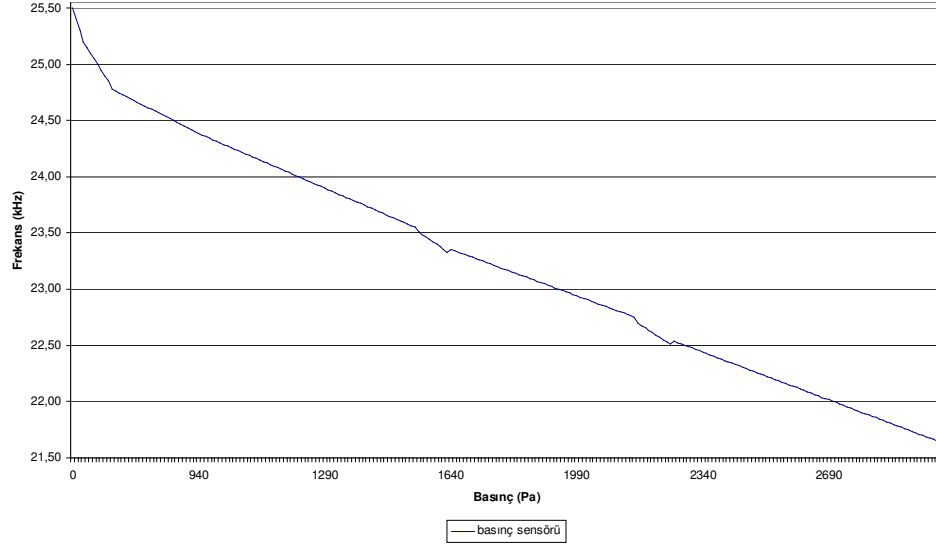
Yapısı kabaca belirtilen veri depolama sistemi kullanılarak elde edilen genel sensör karakteristikleri, Şekil 5.2, Şekil 5.3 ve Şekil 5.4 ile gösterilmektedir. Şekillerdeki karakteristikler, sensörlere lineerleştirme yapıları uygulamadan önceki karakteristikleridir.



Şekil 5.2: Uygulamalarda kullanılan ÇM NTC sıcaklık sensörünün, çalışılan bölge için genel karakteristiği



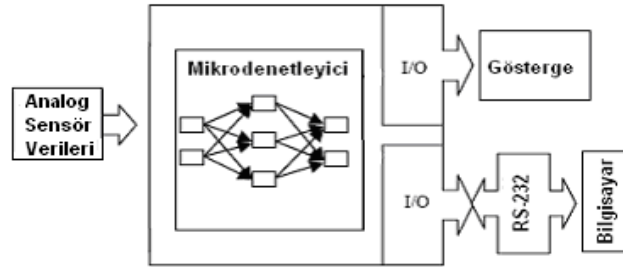
Şekil 5.3: Uygulamalarda kullanılan fırına ait NTC sıcaklık sensörünün, çalışılan bölge için genel karakteristiği



Şekil 5.4: Uygulamalarda kullanılan frekans çıkışlı basınç sensörünün, çalışılan bölge için genel karakteristiği

5.1 Çok Katmanlı Ağ Uygulamaları

Çok katmanlı ağlarla yapılan uygulamalar için kullanılan sistemin yapısı Şekil 5.5 ile verilmektedir.



Şekil 5.5: Çok Katmanlı YSA için Kullanılan Sistemin Yapısı

Çok katmanlı ağ yapısının oluşturulması aşamasında sensörün referans verileri ve referans verilere karşılık elde edilen gerçek sensör verileri eğitime tabi tutulmuştur. Ağ parametrelerinin oluşturulması için kullanılan eğitim algoritması, geriye yayılım algoritmasıdır. Yazılımda kullanılan algoritma Bölüm 3.4 de belirtilmiştir.

Yapay sinir ağları tabanlı sensör lineerleştirme yapısının oluşturulması iki aşamada yapılmaktadır. İlk aşama olan eğitim aşaması, bilgisayar üzerinde gerçekleştirilmektedir. Geriye yayılım algoritmasının bilgisayar üzerinde yapılabilmesi için C programlama dili ile program oluşturulmuştur. Ağ parametrelerinin oluşturulabilmesi için aşağıda aşamaları belirtilen C koduna, txt

dosyası ile bir takım bilgiler önceden ağa girilmektedir. Bu veriler; ağa girilen verinin boyutu, test verisi olarak kullanılacak olan veri adedi, YSA giriş boyutu büyüklüğü, ağda gizli katmanda kullanılacak olan nöron sayısı, mikrodenetleyicinin ADC modülü ya da CCP modülü çıkışında elde edilen sensör verisi, her bir sensör verisine karşılık gelen referans sensör verileridir.

Çok katmanlı yapay sinir ağının giriş ve çıkış katlarında birer adet nöron kullanılmıştır. Gizli katmanda ise 8, 7, 6, 5, 4, 3 ve 2 nöron kullanılmıştır. Buna göre kullanılan ağ yapıları aşağıdaki gibi olmuştur.

- a. $1*8*1$
- b. $1*7*1$
- c. $1*6*1$
- d. $1*5*1$
- e. $1*4*1$
- f. $1*3*1$
- g. $1*2*1$

5.1.1 Çok katmanlı ağ yapıları için oluşturulan C kodunun açıklanması

Eğitim aşaması için oluşturulan C kodunda aşağıda belirtilen konuların her birisi için ayrı alt fonksiyonlar oluşturulmuştur. Bu fonksiyonların başlıkları aşağıda belirtilmiştir.

- a. Dosyadaki giriş verilerinin normalizasyonu,
- b. Ağın dinamik belleğinin adreslenmesi,
- c. Ağ başlangıç değerleri olarak rastgele ağırlıkların verilmesi ve ağ başlangıcının oluşturulması
- d. Eğitim algoritması olarak geriye yayılım algoritmasının uygulanması
- e. Eğitim algoritmasının uygulanması sonucu elde edilen ağ parametreleri ile YSA ileri yol ağının kurulması ve ortalama karesel hata algoritmasının yardımıyla girişteki bir kısım verilerin test edilmesi,
- f. Bütün işlemlerin sonunda bilgisayar belleğinde tutulan tüm verilerin boşaltılması

5.1.1.1 Giriş verilerinin normalizasyonu

Bu aşamalardan ilki olan normalizasyon aşamasında, txt dosyası ile alınan sensör ADC ham verileri ve buna karşılık gelen referans verilerinin her birisinin kendi arasında minimumu ve maksimumu bulunur. Her bir veriyi 0 ve 1 arasında normalize etmek amacıyla aşağıdaki formül her birine ayrı olmak üzere referans sensör verileri ve sensör ham verilerine uygulanmıştır:

$$\text{Normalize Değer} = (\text{Veri-Dizi Minimum Değeri})/(\text{Dizi Maksimum Değeri}) \quad (5.1)$$

5.1.1.2 Verilerin adreslenmesi

0 ile 1 arasında normalize edilen tüm veriler, algoritmadaki bir sonraki aşamaya aktarılmaktadırlar. Bir sonraki aşamada bilgisayar üzerinde işlemlerin yapılabilmesi için dinamik olarak bellek ayırma işlemi yapılmaktadır. Böylelikle, C koduna girilen eleman sayısı ve ağ parametre miktarına göre bellekte yer ayrılmaktadır. Gereksiz yere fazla bellek kullanımı ve gereğinden az bellek kullanımı sonucu oluşabilecek hataların önüne geçilmektedir.

5.1.1.3 YSA ağırlıklarına rastgele değerler atanması

YSA yı oluşturan tüm ağırlıklara -1 ile 1 arasında değer alacak biçimde rastgele sayılar atanmıştır. Buna göre, YSA ile yapılan her bir yeni denemede farklı başlangıç değerleri ile işlem yapılmaktadır. Rastgele sayı üretiminde C standart fonksiyonlarından “rand()” kullanılmıştır.

5.1.1.4 YSA eğitim aşaması

Eğitim aşamasında geriye yayılım algoritması uygulanmıştır. Kullanılan geriye yayılım algoritmasının aşamaları Bölüm 3.4 ile verilmiştir. Eğitim algoritması, YSA çıkışı ile söz konusu çıkışa karşılık gelen referans değer arasında belirlenen ortalama karesel hatanın altına gelene kadar devam ettirilmektedir. Çok katmanlı ağ yapısı için uygulamalarda kullanılan hata değeri, %0,00001 dir. Çok katmanlı ağ yapısında kullanılan etkinleştirme fonksiyonu için aşağıdaki matematiksel ifade kullanılmıştır.

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}} \quad (5.2)$$

Yazılımsal algoritmalarda, ağda kullanılan etkinleştirme fonksiyonunun türevi aşağıdaki matematiksel ifadedir. Bu ifadeye, sigmoid fonksiyonunun türevinin alınması ile ulaşılmaktadır.

$$Etkinleştirme(x) = x * (1 - x) \quad (5.3)$$

YSA parametresi olarak bilgisayar üzerinde yapılan eğitimde öğrenme hızı sabiti olan η 0.5, momentum sabiti olan α ise 0.5 olarak alınmıştır.

5.1.1.5 YSA test aşaması

Eğitim aşaması sonucu parametreleri oluşan ağın test edilmesi için kullanıcı tarafından txt dosyasında belirtilen değer kadar veri test için ayrılmaktadır. Belirtilen sayı kadar veri, girilen tüm verilerin arasından rastgele olarak seçilmektedir.

Test amaçla oluşturulan yazılımda eğitim algoritması sonucu oluşan YSA ağırlıkları ile ileri yol ağı oluşturulmuştur. İleri yol ağında kullanılan etkinleştirme fonksiyonu, non-lineer yapıdaki sigmoid fonksiyonudur. Sigmoid fonksiyonunun gerçekleşmesi için matematiksel ifadelerden yararlanılmıştır. Kullanılan matematiksel ifadeler, mikrodenetleyicinin hesaplama kapasitesine uygun olacak formatta oluşturulmuştur. Buna göre elde edilen sonuçlar tatmin edici olmuştur.

Test aşamasında ağın ne kadar başarılı olduğunun görülebilmesi için ortalama karesel hata hesabından yararlanılmıştır. Ortalama karesel hata hesabında, test edilen veri için ağın çıkış değeri ile ham veriye karşılık gelen referans veri arasındaki farka bakılmaktadır. Bu değerler arasındaki toplam ortalama karesel hata, bilgisayar ekranına bastırılmaktadır.

$$OKH = \frac{\sum (Veri - referans_veri)^2}{N} \quad (5.4)$$

Ortalama karesel hata formülüne göre veri, sensörden alınan bilgilerden birisini, referans veri ise söz konusu sensör bilgisine karşılık referans ölçüm cihazından alınan bilgidir. N ise giriş veri dizisindeki toplam eleman sayısıdır.

5.1.1.6 Verilerin bellekten boşaltılması

Son aşamada ise dinamik olarak bellekte tutulan alanlar, daha sonraki uygulamaların olumsuz etkilenmemesi amacıyla boşaltılması gerekmektedir. Böylelikle, tekrarlı

olarak yapılacak tüm uygulamalar aynı rastgele olasılıkla başlamakta; testin yapıldığı bilgisayarın belleği ekonomik olarak kullanılmaktadır.

Bilgisayar üzerinde yapılan işlemlerin sonucunda oluşan YSA ağırlıkları, sistemden istenen ağ yapısına göre oluşmakta ve bilgisayar üzerinden görülebilmektedir. Bilgisayar üzerinde koşturulan program ile elde edilen YSA bilgileri, mikrodenetleyiciye aktarılmaktadır. Bilgisayar üzerindeki eğitim sonucunda istenilen ortalama karesel hata düzeyindeki bir başarımda, mikrodenetleyiciye sadece YSA ileri yol aşaması aktarılmaktadır.

Giriş bölümünde özellikleri belirtilen çok katmanlı ağ yapısı için uygulamalar kapsamında bir adet ara katman kullanılmıştır. Tek ara katman kullanılmasının temel nedenleri aşağıdaki maddelerde belirtilmiştir.

- a. Tatmin edici sonuçlar, tek ara katman kullanımı ile elde edilebilmektedir[8-9].
- b. Gereksiz yere fazladan ara katman kullanılması sonucu yapay sinir ağı istenen sonucu vermeyebilmektedir[1-3, 8-9]. Bunun temel nedeni, yapay sinir ağında gereğinden fazla katman ya da nöron kullanımı ile sistemde ıraksama problemlerinin oluşabilmesidir.
- c. Eklenen her ara katman, mikrodenetleyiciye matematiksel işlem yükü getirmekte, yapılacak işlem süresini arttırmakta, mikrodenetleyici içerisine yazılan yazılımsal kodun hafızada daha geniş yer tutmasına neden olmaktadır[8-9]. Yapılan uygulamalar sonucunda elde edilen tablolardan bu durum görülebilir.

5.1.2 Çok katmanlı ağ yapıları için mikrodenetleyiciye girilen YSA yazılımı

YSA çok katmanlı ağ yapısı için mikrodenetleyiciye girilen yazılımın içeriği, aşağıdaki maddelerde belirtilmiştir.

- a. Nöronlar arası ağırlıkları içeren tablo,
- b. YSA'nın eğitimi aşamasında sensör ADC verilerinin normalizasyonunda kullanılan maksimum ve minimum referans sıcaklık değerleri ve referans sıcaklıklara karşılık gelen ADC verileri,
- c. YSA için temel sabitler,

- d. Sensörden gelen ADC verilerinin YSA ya girilmeden önce normalize edilmesi
- e. YSA ileri yol algoritması
- f. Elde edilen normalize verinin ilgili sensör birimine çevrilmesi
- g. Son verinin bilgisayara aktarılması

Nöronlar arası bağlantılardaki ağırlıkları içeren tablo değerlerinin tamamı 16 bitlik verilerden oluşmaktadır. Tablonun boyutu, ilgilenilen ağdaki gizli katmanın düğüm sayısı ile orantılı olarak artmaktadır. $1 \times n \times 1$ türünde çok katmanlı YSA için giriş boyutu ile gizli katman arasında $n \times 2$ boyutundaki bir tabloda; gizli katman ile çıkış katmanı arasında ise $(n+1) \times 1$ boyutunda bir tabloda veriler tutulmaktadır. Örnek olarak mikrodenetleyiciye girilen ağırlıkların değerleri, Tablo 5.1 ve Tablo 5.2 ile verilmiştir. Ağırlık verileri, çamaşır makinesi NTC sıcaklık sensörü uygulamasında ve gizli katmanında sekiz nöron içeren yapı için oluşturulmuştur. Uygulamalarda kullanılan ağ parametreleri eklerde verilen dokümanlarla gösterilmiştir.

Tablo 5.1: Çok katmanlı ağlarla oluşturulan YSA yapısı için giriş katmanı ile gizli katman arasındaki ağırlıkları içeren tablo

Giriş katmanı			
Gizli Katman	Nöron No	1	2
	1	11,59409237	-8,43051
	2	0,498209208	-3,31709
	3	2,890365839	-5,02648
	4	5,416407585	-6,12306
	5	0,654543936	-3,36245
	6	0,845215797	-3,6007
	7	-4,330737591	-1,46246
	8	21,90751457	-13,034

Tablo 5.2: Çok katmanlı ağlarla oluşturulan YSA yapısı için gizli katman ile çıkış katmanı arasındaki ağırlıkları içeren tablo

Nöron No	Ağırlık
1	-4,54637
2	1,36364
3	-0,4515
4	-1,51252
5	1,171396
6	0,894873
7	2,912601
8	1,452222
Kutuplama	-0,02191

YSA eğitimi aşamasında sensör ADC verilerinin 0 ile 1 değerleri arasında normalize edildiğine önceki bölümlerde değilmişti. Mikrodenetleyiciye girilen ileri yol ağının girişine gelen ADC verilerini 0 ile 1 arasında normalize edebilmek için YSA eğitim aşamasındaki en büyük ve en küçük ADC verileri kullanılmaktadır. Sensöre ait ADC verisi, mikrodenetleyici içerisinde aşağıdaki formülle normalize edilir.

$$\text{Normalize Veri} = (\text{ADC Verisi} - \text{ADC_Min}) / \text{ADC_Max} \quad (5.5)$$

Formülde görülen ADC_Min ifadesi, YSA eğitim aşamasında kullanılan ADC verilerinin en küçüğü; ADC_Max ifadesi, YSA eğitim aşamasında kullanılan ADC verilerinin en büyüğüne karşılık gelmektedir.

YSA ileri yol algoritması çıkışında 0 ile 1 arasında normalize edilmiş veri oluşmaktadır. Belirtilen veriyi sıcaklık, basınç gibi ilgilenilen birimlere çevirebilmek için ADC_Min ve ADC_Max değerine karşılık gelen referans çıkışlardan yararlanılmaktadır. Bu çıkışlara sırasıyla Ref_Min ve Ref_Max olarak ad verilirse istenilen birimdeki veriye aşağıdaki formülle ulaşılır:

$$\text{Hedef Veri} = \text{YSA_Çıkış} * \text{Ref_Max} + \text{Ref_Min} \quad (5.6)$$

Mikrodenetleyici içerisine kodlanan YSA ileri yol algoritması, bilgisayar üzerinde koşturulan ileri yol için kullanılan algoritmanın tamamen aynısıdır. İleri yol algoritması genel yapısı Şekil 4.5 ile gösterilmiştir.

Mikrodenetleyicide normalize edilmiş olan sensör verisi, şekilde gösterilen giriş katmanından, gizli katmandaki her bir nörona, eğitim aşamasında belirlenmiş olan her bir ağırlıkla çarpılarak aktarılır. Gizli katmandaki her bir nöronda oluşan toplam, etkinleştirme fonksiyonu olarak kullanılan sigmoid fonksiyonundan geçirilir. Kullanılan sigmoid fonksiyonu, eğitim aşamasında kullanılan aşağıdaki fonksiyondur.

$$\text{Sigmoid}(x) = \frac{1}{1 + e^{-x}} \quad (5.7)$$

Giriş katmanı ile gizli katman arasında yapılan işlemler aşağıda belirtilmiştir:

$$v[i] = \sum w_x[i][j] * u[j] \quad (5.8)$$

Denklemdaki $u[j]$, sensörden gelen verilerin normalize edilmiş halidir. Bunun yanında $w_x[i][j]$ değişkeni j numaralı giriş katmanındaki nöron ile i numaralı gizli katmandaki nöron arasındaki ağırlıktır. Denklemden belirtilen $v[i]$ ise gizli katmanda

yer alan i numaralı nörondaki çarpımların toplamıdır. Gizli katmandaki nöronlar üzerinde aşağıdaki denklemde belirtildiği gibi çarpımların toplam değerleri etkinleştirme fonksiyonundan geçirilmektedir.

$$x[i] = \text{sigmoid}(v[i]); \quad (5.9)$$

Denklemde belirtilen $x[i]$, gizli katmandaki nöronlarda non-lineer etkinleştirme fonksiyonundan geçirilmiş olan çarpımların toplamı değeridir.

Belirtilen non-lineer etkinleştirme fonksiyonundan geçirilerek gizli katmana aktarılan veriler, gizli katmandaki nöronlarla çıkış katmanındaki nöron arasında, eğitim aşamasında oluşturulan ağırlıkla çarpılarak çıkış katmanına aktarılır. Yapılan çarpma işlemi aşağıda belirtilmiştir.

$$y = \sum w_y[j] * x[j] \quad (5.10)$$

Formüldeki y bağımlı değişkeni, lineer çarpma işlemi sonucunda çıkış katmanına aktarılan veriyi ifade etmektedir. $x[j]$ bağımsız değişkeni, non-lineer etkinleştirme fonksiyonunun uygulanması sonucu gizli katmandaki j numaralı nöronun değerini, $w_y[j]$ bağımsız değişkeni ise eğitim algoritmasının uygulanmasının sonucunda gizli katmandaki j numaralı nöron ile çıkış katmanı arasındaki ağırlığı ifade etmektedir. Formülden görülebileceği gibi gizli katman ile çıkış katmanı arasındaki işlem lineerdir. Gizli katmandan çıkış katmanına her bir nörondan belirlenen ağırlıkla çarpılarak gelen veriler çıkış katmanında toplanır ve sonuç elde edilir. Elde edilen değer, ilgili sensöre ait istenilen sonucun normalize edilmiş halidir.

Sonucun istenilen sensör birimine çevrilmesi için YSA çıkışında elde edilen veriye eğitim aşamasındaki normalizasyon işleminin tersi uygulanır. Bu amaçla elde edilen veri aşağıdaki fonksiyondan geçirilir.

$$\text{Sensör değeri} = y * (MY) + mY \quad (5.11)$$

Formüldeki y değişkeni, YSA çıkışındaki normalize değeri; MY değişkeni, eğitim aşamasında kullanılan en büyük referans sensör değerini; mY değişkeni, eğitim aşamasında kullanılan en küçük referans sensör değerini ifade etmektedir.

İşlemlerin sonucunda elde edilen sensör değeri 16 bitliktir. Mikrodenetleyici içerisindeki veri, bilgi kaybını engellemek için, sekizer bitlik iki parçaya bölünmüş ve iki farklı değişkene aktarılmıştır. Veri depolama amacıyla kullanılan programla

verinin bilgisayara aktarılması, sekiz bitlik veriler halinde olmaktadır. İki parçaya bölünen sensör verisi, bilgisayar üzerinde tekrar birleştirilir.

Referans sensör verisi ve gerçek sensör verisinin depolanması aşamasında sıcaklık ve basınç sensörleri referans verileri için ilgilenilen sensör biriminde 0.00001 doğrulukta veriler toplanmıştır.

5.2 Radyal Taban Fonksiyonları Kullanılarak Yapılan Uygulamalar

Radyal taban fonksiyonları ile yapılan uygulamalar için kurulan sistem, çok katmanlı YSA için önceki bölümde anlatılan sistemle aynıdır.

Radyal taban fonksiyonları ile oluşturulan YSA parametreleri bilgisayar üzerinde yapılan eğitimle oluşturulmaktadır. Kullanılan eğitim algoritması, geriye yayılım algoritmasıdır.

Çok katmanlı ağ yapılarında belirtildiği gibi eğitim algoritması için bilgisayar üzerinde C programlama dili ile program oluşturulmuştur. Kullanıcının tercihinine göre ağa ait bir takım bilgiler, programa dışarıdan oluşturulan bir dosya aracılığı ile girilmektedir. Kullanıcı tarafından girilen bilgiler; ağın eğitimi için toplanan ham verilerin toplamı, ağda test verisi olarak kullanılacak olan veri adedi, YSA giriş katmanının büyüklüğü, ağda gizli katmanda kullanılacak olan nöron sayısı, ilgili sensöre ilişkin mikrodenetleyicinin ADC modülü ya da CCP modülü çıkışında elde edilen ham veriler, her bir ham sensör verisine karşılık gelen referans sensör verileridir.

Yapılan uygulamalarda bir adet giriş katmanı, bir adet gizli katman ve bir adet çıkış katmanı içeren yapı kullanılmıştır. Gizli katmanda 8, 6, 5, 4, 3 ve 2 adet nöron kullanılmıştır. Buna göre radyal taban fonksiyonları kullanılarak yapılan uygulamalar kapsamında oluşturulan ağ yapıları aşağıda belirtilmiştir.

- a. 1*8*1
- b. 1*7*1
- c. 1*6*1
- d. 1*5*1
- e. 1*4*1

f. 1*3*1

g. 1*2*1

Uygulamalarda tek gizli katman denenmesinin nedeni, radyal taban fonksiyonları ile oluşturulan yapıların tek gizli katmanla kullanılmalarıdır[1-3].

5.2.1 Radyal taban fonksiyonlarının eğitimi için oluşturulan C kodu

Radyal taban fonksiyonları ile oluşturulan YSA'nın eğitim aşaması için oluşturulan C kodunda aşağıda belirtilen konuların her biri için alt fonksiyonlar oluşturulmuştur.

- Dosyadaki giriş verilerinin normalizasyonu,
- Ağın dinamik belleğinin adreslenmesi,
- Ağa başlangıç olarak rastsal olarak ağırlıkların verilmesi ve ağ başlangıcının oluşturulması
- Eğitim algoritması olarak geriye yayılım algoritmasının uygulanması
- Eğitim algoritmasının uygulanması sonucu elde edilen ağ parametreleri ile YSA ileri yol ağının kurulması ve ortalama karesel hata algoritmasının yardımıyla girişteki bir kısım verilerin test edilmesi,
- Bütün işlemlerin sonunda bilgisayar belleğinde tutulan tüm verilerin boşaltılması

Yukarıda belirtilen yazılım aşamalarının geneli, çok katmanlı ağ yapılarında belirtilen adımlarla aynıdır. Oluşturulan yazılımlar arasındaki farklar, eğitim ve test aşamasında kullanılan non-lineer fonksiyon ve onun türevidir. Radyal taban fonksiyonları için kullanılan non-lineer baz fonksiyonu aşağıda verilmiştir.

$$\varphi_j(x) = \exp\left(-\frac{(x - \mu_j)^2}{2\sigma_j^2}\right) \quad (5.12)$$

Radyal taban fonksiyonlarını bilgisayarda ve mikrodenetleyicide gerçeklemek için yazılımlarda aşağıdaki fonksiyon kullanılmıştır:

$$Etkinlestirme(x_j) = \exp(-(m_{ij} - x_j)^2) \quad (5.13)$$

Yukarıdaki ifadede m_{ij} , nöronların merkezlerini; x_j ise normalize girişleri ifade etmektedir. Mikrodenetleyici içerisinde bölme işlemi yapma ihtiyacını ortadan kaldırmak amaçlı olarak işlem kolaylığı açısından $2\sigma^2$ ifadesi 1 olarak alınmıştır.

Yapılan uygulamalar için eğitim aşamasında öğrenme hızı 0.5 ve momentum sabiti 0.5 olarak alınmıştır.

5.2.2 Radyal taban fonksiyonları uygulamaları için mikrodenetleyicide oluşturulan YSA yazılımı

YSA'nın Radyal Taban Fonksiyonlarını kullanarak oluşturulan YSA yapısı için mikrodenetleyici içerisine girilen yazılımın içeriği, aşağıdaki maddelerde belirtilmiştir:

- a. Nöronların merkezlerini içeren tablo,
- b. Nöronlar arası ağırlıkları içeren tablo,
- c. YS eğitim aşamasında sensör ADC verilerinin normalizasyonunda kullanılan maksimum ve minimum referans sıcaklık değerleri ve referans sıcaklıklara karşılık gelen ADC verileri,
- d. YSA için temel sabitler,
- e. Sensörden gelen ADC verilerinin YSA ya girilmeden önce normalize edilmesi
- f. YSA ileri yol algoritması
- g. Elde edilen normalize verinin ilgili sensör birimine çevrilmesi
- h. Son verinin bilgisayara aktarılması

Nöronların merkezlerini içeren tablolardaki değişkenler 16 bitlik verilerden oluşmaktadır. Tablonun boyutu, gizli katman düğüm sayısı ve giriş boyutu tarafından belirlenmektedir. Örneğin, gizli katmandaki düğüm sayısı n ise girilen merkez için oluşturulan matrisin boyutu $n*2$ olur. Matris boyutu ifadesindeki 2 değeri, girişteki düğüm ve kutuplama için kullanılan düğüm nedeniyledir. Matrisin her bir hücresindeki sayı, 16 bitlik gerçek sayılardan oluşmaktadır. Çamaşır makinesi NTC sıcaklık sensörü ile yapılan ve gizli katmanında sekiz nöron içeren uygulama için Tablo 5.3 içeriğinde görülen merkezler mikrodenetleyiciye yazılmıştır.

Nöronlar arası bağlantılardaki ağırlıkları içeren tablo değerlerinin tamamı 16 bitlik verilerden oluşmaktadır. Tablonun boyutu, ilgilenilen ağdaki gizli katmanın düğüm sayısı ile orantılı olarak artmaktadır. $1*n*1$ türünde radyal taban fonksiyonları ile oluşturulan bir ağda, gizli katman ile çıkış katmanı arasında $(n+1)*1$ boyutunda tabloda veriler tutulmaktadır. Verileri içeren örnek değerler Tablo 5.4 ile verilmiştir. Tablodaki değerler, çamaşır makinesi NTC sıcaklık sensörü uygulamasında kullanılan değerlerdir.

Tablo 5.3: Radyal taban fonksiyonları ile oluşturulan YSA için gizli katmanında 8 nöron içeren yapıda nöronların merkezlerini içeren tablo

Nöron No	x	y
1	0,572458446	-0,4999125
2	-0,944122493	0,476199716
3	0,572459698	-0,499918342
4	0,572459698	-0,499918342
5	-0,937970161	0,411920786
6	0,572459698	-0,499918342
7	0,572456896	-0,499921739
8	0,572459698	-0,499918342

Tablo 5.4: Radyal taban fonksiyonları ile oluşturulan YSA için gizli katmandaki nöronlarla çıkış katmanı arasındaki ağırlıkları içeren tablo

Nöron No	Ağırlık
1	-1,000058651
2	0,986070752
3	-1,000074625
4	-1,000074625
5	1,013823152
6	-1,000074625
7	-1,000084639
8	-1,000074625
Kutuplama	0,510014534

İleri yol aşamasında giriş katmanı ile gizli katman arasında yapılan işlemler aşağıda belirtilmiştir. Bu kapsamda yapılan işlemlerden ilki, gizli katmandaki nöronların merkezleri ile girişe gelen sensör verisine gaussiyen etkinleştirme fonksiyonunun uygulanmasıdır.

İleri yol algoritmasında kullanılan non-lineer etkinleştirme fonksiyonu aşağıda belirtilmiştir.

$$v[i] = \sum [(m[i][j] - u[j]) * (m[i][j] - u[j])] \quad (5.14)$$

$$x[i] = \exp((-1)*v[i]) \quad (5.15)$$

Denklemdaki $m[i][j]$, gizli katmandaki nöronların merkezlerini ifade etmektedir. Bunun yanında $u[j]$, sensörden gelen normalize giriş vektörünü gösterir. Denklemdaki $x[i]$ değişkeni gizli katmandaki nörona iletilen ve non-lineer fonksiyondan geçirilmiş olan veriyi ifade etmektedir. Mikrodenetleyiciye işlem yükü getirmemek için gauss fonksiyonundaki $2\sigma^2$ değeri 1 olarak alınmıştır.

Belirtilen non-lineer fonksiyonun kullanılması ile girişteki veri gizli katmana aktarılmış olur. Gizli katmanda elde edilen veriler, aşağıda belirtilen lineer çarpma fonksiyonu ile çıkışa aktarılır.

$$y = \sum wy[j] * x[j] \quad (5.16)$$

Yukarıdaki formülde belirtilen $wy[j]$, gizli katmandaki j numaralı nöron ile çıkış arasındaki ağırlığı; $x[j]$ bağımsız değişkeni, gizli katmandaki j numaralı nöronun etkinleştirme fonksiyonu çıkışını ifade eder.

Diğer aşamaların geneli, çok katmanlı ağ yapıları ile oluşturulan sistemde anlatılanlarla aynıdır.

5.3 Tablo Tutma Yöntemi İle Yapılan Uygulamalar

Tablo tutma yöntemi ile yapılan uygulamanın temelinde, daha önceden elde edilen sensör verilerinden yararlanarak sensörlerin elektriksel çıkışlarına karşılık referans sensör verilerinin bir tabloda tutulması yatmaktadır. Yöntemin ayrıntıları, Bölüm 2.2.1 içerisinde belirtilmiştir. Bölümde belirtildiği gibi sistem için ek donanım kullanılmamış, mikrodenetleyicinin mevcut yapısında bulunan donanımdan yararlanılmıştır. Bu amaca yönelik olarak mikrodenetleyici içerisinde yazılım oluşturulmuştur. Yöntem için oluşturulan sistemin blok diyagramı Şekil 5.6 ile gösterilmiştir.



Şekil 5.6: Tablo tutma yöntemi için kullanılan sistemin blok diyagramı

5.3.1 Tablo tutma yöntemi için mikrodnetleyicide yapılan yazılım

Tablo tutma yöntemi için mikrodnetleyicide oluşturulan yazılımın içeriği, aşağıdaki maddelerde belirtilmiştir.

- a. Sensör dönüşüm tablosu oluşturulması,
- b. Sensörün elektriksel çıkışına karşılık gelen verinin, dönüşüm tablosu içerisinde taranması ve bulunan verinin değişkene atanması,
- c. Tablo çıkışında bulunan ve değişkene atanan değerin bilgisayara iletilmesi,

5.3.1.1 Sensör dönüşüm tablosu

Sensör dönüşüm tablosu; sensöre ilişkin mikrodnetleyici içerisindeki ADC ya da CCP blokları çıkışındaki değerin gerçek değerlerle karşılığını içermektedir. Yönteme yönelik oluşturulan yazılımda verinin tablodaki konumu, sensörün gerçek değerini; tablo konumuna karşılık gelen sensör değeri ise sensörün mikrodnetleyici içerisindeki değerini içermektedir. Örneğin; `tablo[0]=22` ifadesinde 22, sensöre ilişkin mikrodnetleyici içerisindeki değeri ifade eder. 0 ise C cinsinden sıcaklığı ifade eder ve ADC bloğu çıkışındaki 22 değerine karşılık gelen gerçek değerdir.

Sensör dönüşüm tablosunun boyutu, uygulamadan uygulamaya farklılık göstermektedir. Örneğin, 0 ile 100C arasındaki değerleri 0.5C aralıklarla ifade edebilmek için çamaşır makinesi NTC uygulamasında 200 değer; fırın NTC uygulamasında 30 ile 230C arasındaki değerleri 0.5C hassasiyetle ifade edebilmek için 400 değer tabloda tutulmuştur.

5.3.1.2 Tabloda verinin taranması

Oluşturulan yazılıma göre sensöre ilişkin gerçek veriye ulaşmak için öncelikle, oluşturulan tablo içerisinde ADC ya da CCP bloğu çıkışında elde edilen veriye ilişkin arama yapılır. Tablo içerisinde, aranan değere eşit ya da aranan değerden büyük bir değer bulundu ise arama işlemi bitirilir. Bulunan değerin konumu bir değişkene aktarılır. Konum bilgisinin saklanması amacıyla 16 bitlik değer kullanılmıştır.

5.3.1.3 Tabloda bulunan değerin bilgisayara aktarılması

Mikrodnetleyici içerisinde sensör dönüşüm tablosundan elde edilen konum bilgisi, ulaşılması hedeflenen değerdir ve bu değerin değişkene aktarılmasının ardından aynı

veri, veri depolama sistemine gönderilir. Konum bilgisinin bilgisayara gönderilmesi aşamasında değişken, sekizer bitlik iki parçaya ayrılmakta ve iki parça halinde bilgisayara aktarılmaktadır. Bilgisayara aktarma aşamasında Şekil 5.6 ile gösterilen sistemin blok diyagramından da görülebileceği gibi RS-232 protokolünden yararlanılmaktadır.

5.4 Radyal Taban Fonksiyonlarına İlişkin Yapılan Uygulamaların Sonuçları

Radyal taban fonksiyonları ile oluşturulan YSA uygulamaları için ağırlıklı bir adet giriş, bir adet çıkış ve bir adet gizli katman içerdiğine daha önce değinilmişti. Radyal taban fonksiyonları ile yapılan uygulamalar arasındaki farklılıklar açısından gizli katmanda kullanılan nöron sayısı farklılık göstermektedir. Gizli katmandaki nöron adetleri 8, 7, 6, 5, 4, 3 ve 2 olmuştur. Buna göre radyal taban fonksiyonları ile yapılan YSA uygulamaları kapsamında 7 farklı yapı test edilmiştir.

Radyal taban fonksiyonları ile oluşturulan YSA yapısı üç farklı türde sensörde denenmiştir. Bu sensör türleri:

- a. Fırında kullanılan NTC
- b. Çamaşır makinesinde kullanılan NTC
- c. Çamaşır makinesinde kullanılan frekans çıkışlı basınç sensörü

YSA çıkışında elde edilen sensör verileri ve referans sensör verileri, önceki bölümlerde belirtilen yöntemlerle veri depolama sistemine aktarılmıştır.

İlk yapılan uygulamalar, çamaşır makinesinde kullanılan NTC ile yapılmıştır. Uygulamalar için beş adet NTC kullanılmıştır. Her bir sensör ile tüm YSA yapıları denenmiştir. Bu kapsamda, tüm sensörler her bir YSA yapısı için ayrı ayrı eğitime tabi tutulmuş, eğitim aşaması sonucu ilgili sensöre ilişkin oluşan ileri yol algoritması, ağırlık merkezleri ve nöronlar arası ağırlıklar mikrodenetleyici kodunun içerisine yazılmıştır.

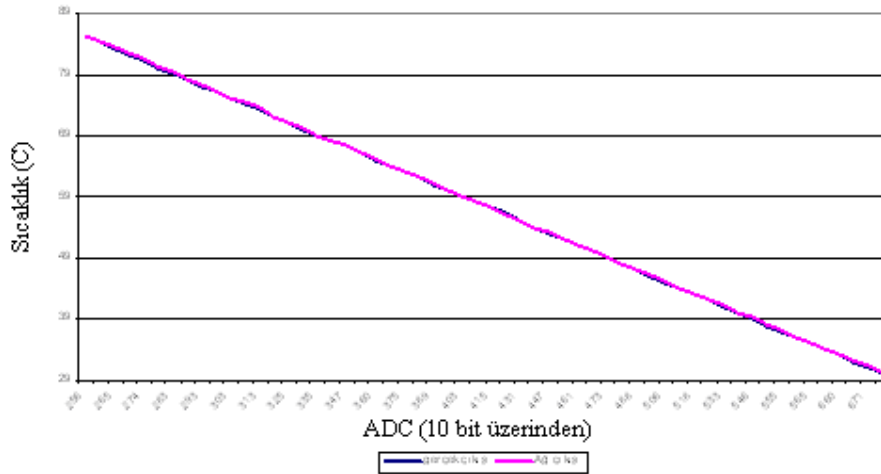
Radyal taban fonksiyonları ile oluşturulan YSA başarımını değerlendirmek için referans veri ile mikrodenetleyici içerisine girilen YSA çıkışında elde edilen veriler arasında oluşturulan ortalama karesel hata değerine bakılmaktadır. Beş adet çamaşır makinesi sıcaklık sensörü ile yapılan uygulamalar için her bir YSA yapısına karşılık olarak elde edilen ortalama karesel hatalar ile Tablo 5.5 oluşturulmuştur. Tablonun

her bir hücrendeki ortalama karesel hata değeri, beş adet sensöre ilişkin elde edilen ortalama karesel hata değerlerinin aritmetik ortalamasıdır.

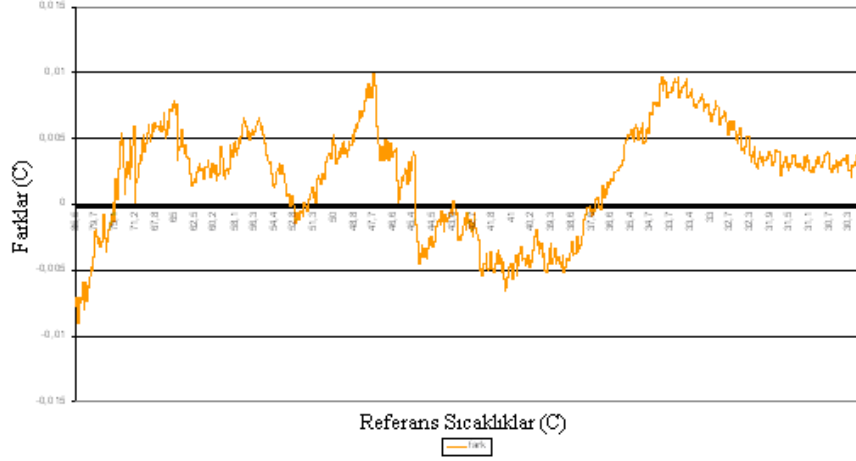
Tablo 5.5: Çamaşır makinesi sıcaklık sensörü için radyal taban fonksiyonları ile oluşturulan YSA yapılarına ilişkin ortalama karesel hata değerleri

Gizli Katmandaki Nöron Sayısı	O.K.H.
8	0,000770
7	0,000785
6	0,000982
5	0,001480
4	0,001850
3	0,002120
2	0,002350

Şekil 5.7 ve Şekil 5.8 ile gizli katmanında iki nöron içeren radyal taban fonksiyonları ile oluşturulmuş olan YSA sonuçları grafiksel olarak gösterilmiştir. Şekil 5.7 ile görülebileceği gibi sensör verileri ile elde edilen ADC değerlerine karşılık referans sıcaklık değerlerinin ve YSA çıkışında elde edilen sıcaklık değerlerinin üst üste bindiği görülebilmektedir. Şekil 5.8 ile gösterilen grafikte, sensörün çalışma aralığı boyunca her bir referans sıcaklık değerlerine karşılık olarak YSA çıkışında elde edilen sıcaklık değerleri ile referans sıcaklık değerleri arasındaki farklar görülebilmektedir. Grafikten görülebileceği gibi farkların genel olarak farklı bölgelere aynı rastgelelikte dağıldığı görülebilmektedir.



Şekil 5.7: Çamaşır makinesi NTC uygulamasında gizli katmanında iki nöron içeren ve radyal taban fonksiyonları ile oluşturulmuş YSA ile her bir sensör ADC değerine karşılık referans sıcaklık ve YSA sıcaklık çıkış değerleri grafiği



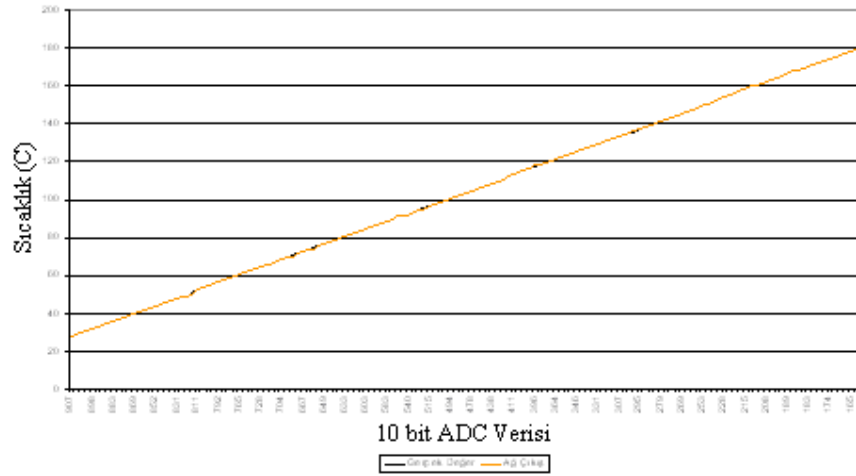
Şekil 5.8: Çamaşır makinesi NTC uygulamasında gizli katmanında iki nöron içeren ve radyal taban fonksiyonları ile oluşturulan YSA çıkışında her bir referans sıcaklık değerine karşılık elde edilen sıcaklık fark değer grafiği

Çamaşır makinesi sıcaklık sensörü ile yapılan çalışmanın benzeri, fırında kullanılan NTC sıcaklık sensörü için de yapılmıştır. Fırındaki NTC sıcaklık sensörü uygulaması için de beş adet aynı ürün kodlu NTC kullanılmıştır. Uygulama ile radyal taban fonksiyonları ile oluşturulan her bir YSA yapısı için Tablo 5.6 ile gösterilen ortalama karesel hata değerleri elde edilmiştir. Tablodaki her bir hücrede yer alan ortalama karesel hata değeri, uygulamada kullanılan beş adet sensöre ilişkin ortalama karesel hata değerlerinin aritmetik ortalamasını ifade etmektedir.

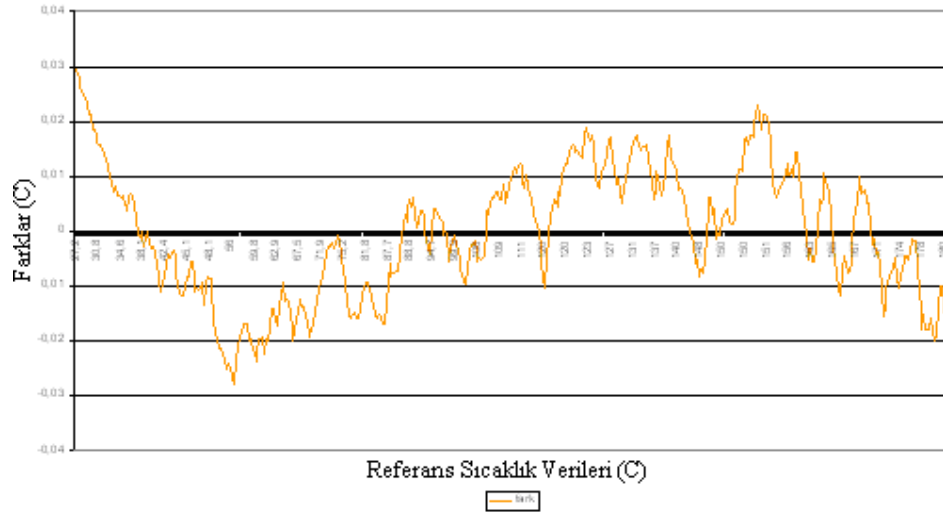
Uygulamalar ile ortaya çıkan sonuçlar ayrıca Şekil 5.9 ve Şekil 5.10 ile belirtilen grafiklerde gösterilmiştir. Şekil 5.9 ile gösterilen grafikte, sensör verisi ile elde edilen ADC değerlerine karşılık referans değerler ile YSA çıkışında elde edilen sıcaklık değerleri görülmektedir. Şekil 5.10 ile referans sensör değerlerine karşılık, referans sıcaklık değeri ile YSA çıkışında elde edilen sıcaklıklar arasındaki farkların dağılımı gösterilmektedir.

Tablo 5.6: Fırın sıcaklık sensörü için radyal taban fonksiyonları ile oluşturulan YSA yapılarına ilişkin ortalama karesel hata değerleri

Gizli Katmandaki Nöron Sayısı	O.K.H.
8	0,00083
7	0,00085
6	0,00097
5	0,00092
4	0,00106
3	0,0012
2	0,00183



Şekil 5.9: Fırında kullanılan NTC uygulamasında gizli katmanında iki nöron içeren radyal taban fonksiyonları ile oluşturulan YSA için 10 bit ADC verilerine karşılık olarak referans sıcaklık ve YSA çıkışındaki sıcaklık değerleri



Şekil 5.10: Fırında kullanılan NTC ile yapılan uygulamalarda gizli katmanında iki nöron içeren radyal taban fonksiyonları ile oluşturulan YSA için her bir referans değere karşılık referans sıcaklık değerleri ile ağ çıkışları arasındaki farkları içeren grafik

Sıcaklık sensörleri ile yapılan çalışmanın benzeri, çamaşır makinesinde kullanılan basınç sensörü için de yapılmıştır. Basınç sensörü uygulaması için de beş adet aynı ürün kodlu frekans çıkışlı basınç sensörü kullanılmıştır. Uygulama ile radyal taban fonksiyonları ile oluşturulan her bir YSA yapısı için Tablo 5.7 ile gösterilen ortalama karesel hata değerleri elde edilmiştir. Tablodaki her bir hücrede yer alan ortalama karesel hata değeri, uygulamada kullanılan beş adet sensöre ilişkin ortalama karesel hata değerinin aritmetik ortalamasını ifade etmektedir.

Tablo 5.7: Çamaşır makinesi basınç sensörü için radyal taban fonksiyonları ile oluşturulan YSA yapılarına ilişkin ortalama karesel hata değerleri

Gizli Katmandaki Nöron Sayısı	O.K.H.
8	0,00526
7	0,00292
6	0,0047
5	0,0056
4	0,0062
3	0,0064
2	0,0067

Tablo 5.5, Tablo 5.6 ve Tablo 5.7 verileri ve şekiller incelendiğinde aşağıdaki ortak noktalar göze çarpmaktadır:

- Çamaşır makinesi sıcaklık sensörü için ortaya çıkan ortalama karesel hata değerleri ile fırın uygulaması sonucu ortaya çıkan ortalama karesel hata değerleri birbirine paralel olmuştur. Çamaşır makinesi sıcaklık sensörü ile yapılan uygulamaların çalışma aralığı 20 ile 90 °C arasında, fırın uygulaması için yapılan uygulamalar ise 30 ile 180 °C arasındadır. Çamaşır makinesi sıcaklık sensörünün uygulama kapsamındaki çalışma aralığı fırın uygulamasındaki çalışma aralığına göre daha dardır. Ancak, sensörler farklı olmasına rağmen çamaşır makinesi sıcaklık sensörü ve fırın sıcaklık sensörü uygulamalarındaki ortalama karesel hata değerleri aynı seviyede çıkmıştır. Değerlerin mertebeleri, tablolarda görülebilmektedir.
- Sıcaklık sensörü uygulamalarında gizli katmanda 8 – 5 sayılarında nöron içeren uygulamalarda ortalama karesel hata değerleri birbirine yakın çıkmaktadır. Bunun temel nedeni, gizli katmanda 8 – 5 arasında nöron içeren YSA yapılarında ağların merkezlerinin bir kısmının birbirine çok yakın çıkmasıdır. Bilgi, Tablo 5.3 ve Tablo 5.4 verileri ile eklerde yer alan YSA ağırlık tablolarına bakılarak görülebilir.
- Çamaşır makinesi basınç sensörü uygulamasında, sıcaklık sensörü ile yapılan uygulamaların sonuçlarından farklı bir sonuç ortaya çıkmıştır. Basınç sensörü uygulamasındaki gizli katmanda sekiz nöronun kullanıldığı koşulda ortalama karesel hata değeri, gizli katmanda daha az sayıda nöron kullanılan YSA yapılarına göre daha büyük çıkmıştır. Ortaya çıkan sonucun temel

nedeni, gizli katmanda gereğinden fazla kullanılan nöronların olması sonucu ıraksamanın ortaya çıkmasıdır[1-3].

- d. Uygulamalarda ortaya çıkan hataların en büyüğü, tablolardan görülebileceği gibi gizli katmanında iki nöron içeren YSA yapılarında görülmüştür. En fazla hatalı olan ve gizli katmanında iki nöron içeren YSA yapıları ile oluşturulan sistemlerde grafiklerden görülebileceği gibi referans sıcaklık değerleri ile YSA çıkışında elde edilen sıcaklık değerlerinin üst üste bindiği görülmüştür. Hataların dağılım grafiği incelendiğinde referans sıcaklıklar ile YSA çıkışında elde edilen sıcaklıkların farklarının belirli bir bölgeye özel olarak artmadığı, farklardaki artış ve azalışların rastgele dağıldığı görülmektedir. Böylelikle kaynaklarda belirtilen ve YSA için sorun olarak görülen çalışma bölgelerinin uç bölgelerinde hataların artması sorunu uygulamalar sırasında görülmemiştir[1-3, 8,9].

Radyal taban fonksiyonları ile oluşturulan YSA yapıları ile yapılan uygulamalarda sistemin başarımını gösterebilecek bir başka ölçüt, tasarım için oluşturulan yazılımın mikrodenetleyici içerisinde kapladığı alandır. Çok katmanlı YSA algoritmasının ileri yol aşamasının mikrodenetleyici içerisine girildiğinde sadece algoritma için bellekte kullanılan yerin byte cinsinden değerlerini Tablo 5.8 içermektedir.

Tablo 5.8: Radyal taban fonksiyonları ile oluşturulan YSA ile yapılan uygulamalar sonucu gizli katmandaki her bir nöron sayısı ile yapılan uygulamaların bellekte kapladığı alan

Gizli Katman Nöron Adedi	Bellekte Kaplanan Alan(Byte)
8	4115
7	4103
6	4091
5	4079
4	4067
3	4055
2	4043

Tablo 5.8 verilerinden görülebileceği gibi mikrodenetleyici belleğinde kaplanan alan açısından gizli katmanında n adet nöron olan YSA yapısı ile n+1 nörona sahip YSA arasında 12 byte fark bulunmuştur. Gizli katmanda kullanılan her bir ek nöron, sistemin bellekte kapladığı alanı yaklaşık olarak %0,3 arttırmaktadır.

5.5 Çok Katmanlı Ağ Yapılarına İlişkin Uygulamaların Sonuçları

Çok katmanlı ağ yapıları ile oluşturulan YSA uygulamaları kapsamında ağ yapısında bir adet giriş, bir adet çıkış ve bir adet gizli katman kullanılmıştır. Yapılan uygulamalar arasındaki temel farklılıklar, kullanılan sensör türü ve gizli katmandaki nöron sayısıdır.

Gizli katmandaki nöron adetleri açısından 7 farklı YSA yapısı denenmiştir. YSA yapılarında gizli katmanda kullanılan nöron adetleri 8, 7, 6, 5, 4, 3 ve 2 dir. Uygulamalarda kullanılan sensör türleri açısından 3 farklı sensör kullanılmıştır. Kullanılan sensör türleri aşağıda verilmiştir.

- a. Fırında kullanılan NTC sıcaklık sensörü
- b. Çamaşır makinesinde kullanılan NTC sıcaklık sensörü
- c. Çamaşır makinesinde kullanılan frekans çıkışlı basınç sensörü

Her bir sensör ile yapılan uygulamalarda, YSA çıkışında elde edilen sensör verileri ve referans sensör verileri, önceki bölümlerde belirtilen yöntemlerle veri depolama sistemine aktarılmıştır.

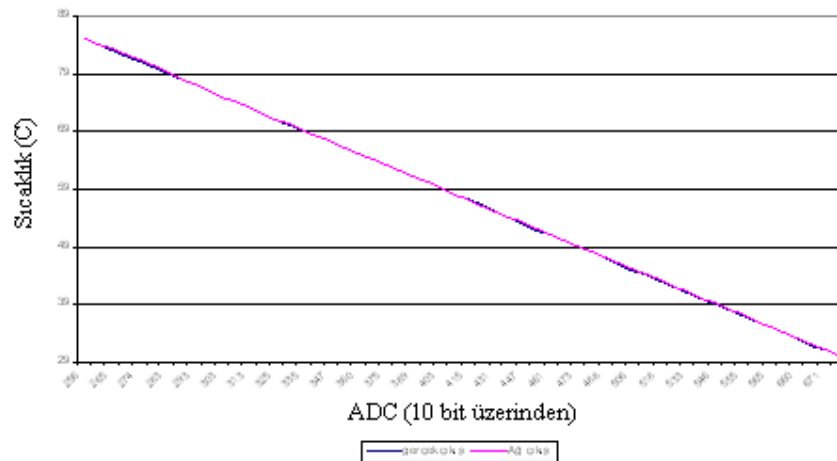
İlk uygulamalar, çamaşır makinesi NTC sıcaklık sensöründe yapılmıştır. Uygulamalar kapsamında aynı model NTC sıcaklık sensöründen beş adet kullanılmıştır. Aynı model her bir NTC sıcaklık sensörünün verileri, veri depolama sistemine aktarılmıştır. Aktarılan verilerin yardımı ile beş adet aynı model NTC sıcaklık sensörünün her biri ayrı olarak çok katmanlı ağ yapısı için eğitime tabi tutulmuş; eğitim aşaması sonucu ilgili sensöre ilişkin oluşan ileri yol algoritması, nöronlar arası ağırlıklar vb. parametreler mikrodenetleyici kodunun içerisine yazılmıştır.

Çok katmanlı ağ yapısı ile oluşturulan YSA başarımını değerlendirmek için referans veri ile YSA çıkışında elde edilen veriler arasında oluşturulan ortalama karesel hata değerine bakılmaktadır. Beş adet çamaşır makinesi NTC sıcaklık sensörü ile yapılan uygulamalar için her bir YSA yapısına karşılık olarak elde edilen karesel hatalar Tablo 5.9 ile belirtilmiştir. Her bir hücredeki ortalama karesel hata değeri, beş adet sensöre ilişkin elde edilen ortalama karesel hata değerlerinin aritmetik ortalamasıdır.

Tablo 5.9: Çamaşır makinesi NTC için çok katmanlı ağ yapıları ile oluşturulan YSA yapılarına ilişkin ortalama karesel hata değerleri

Gizli Katmandaki Nöron Sayısı	O.K.H.
8	0,00096
7	0,00145
6	0,00199
5	0,00259
4	0,00319
3	0,00448
2	0,00673

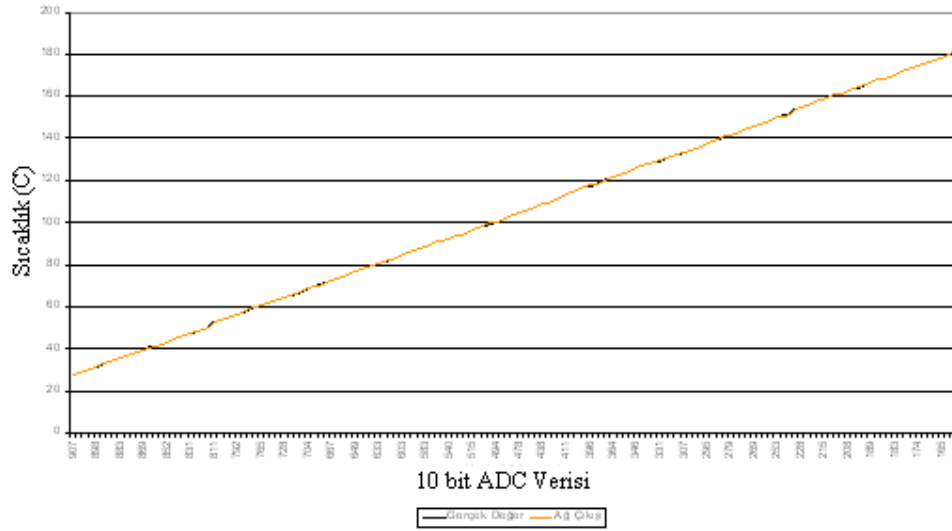
Gizli katmanında iki nöron içeren çok katmanlı YSA uygulamaları ile Şekil 5.11 ve Şekil 5.12 ile gösterilen grafikler ortaya çıkmıştır. Şekil 5.11, çamaşır makinesi NTC sıcaklık sensörü için gizli katmanında iki nöron içeren ağ uygulamaları sonucu her bir sensöre ait ADC verisine karşılık, referans değerler ve YSA çıkış sıcaklık değerlerini içermektedir. Grafikten görülebileceği gibi sensörün çalışma aralığı boyunca YSA çıkış sıcaklık değeri ile referans sıcaklık değerlerinin örtüştüğü görülebilmektedir. Şekil 5.12 grafiği ile sensörün çalışma aralığı boyunca her bir referans sıcaklık değerine karşılık YSA çıkışında elde edilen sıcaklık değerleri ile referans sıcaklık değerleri arasındaki farkın dağılımı görülebilmektedir. Grafikten görülebileceği gibi sadece üç bölgede görülen fark değerlerinin genel çalışma bölgesi göz önüne alındığında arttığı görülmüştür. Sensörün çalışma bölgelerinin uç kısımları ile diğer kısımları arasında belirgin farklılık bulunmamaktadır.



Şekil 5.11: Çamaşır makinesi NTC uygulamasında gizli katmanında iki nöron içeren YSA yapısı sonucu her bir sensör ADC verisine karşılık referans değer ve YSA çıkış sıcaklık değerleri

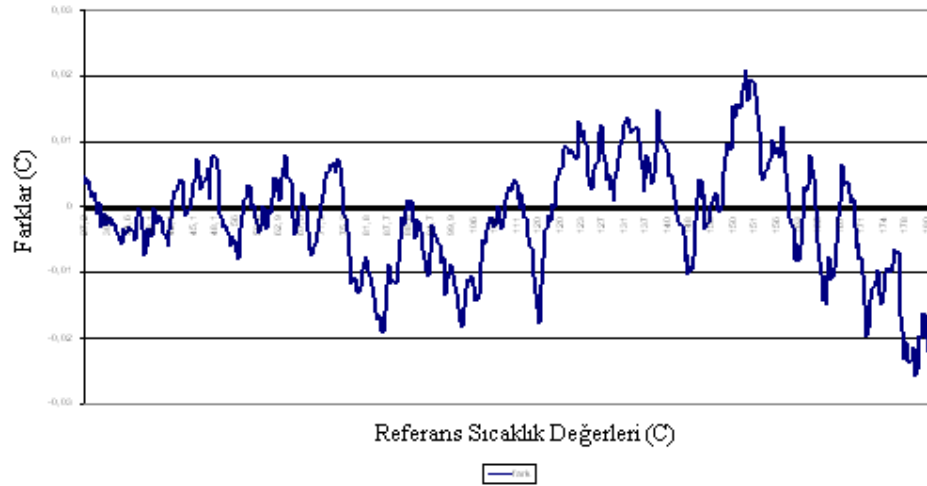
bir ADC değerine karşılık çok katmanlı ağ çıkışında elde edilen sıcaklık değerleri yer almaktadır. Grafikten, sensörün çalışma aralığı boyunca olan gerçek değer ile YSA çıkışındaki değerlerin çakıştığı görülebilmektedir.

Çok katmanlı ağ yapıları ile yapılan fırın sıcaklık sensörü uygulamalarında, referans sıcaklık sensörü ile ağ çıkışı arasındaki farkların dağılımını göstermek için Şekil 5.14 ile gösterilen grafik oluşturulmuştur. Grafikte, yatay eksen referans sıcaklık değerini, dikey eksen ise referans sıcaklık değeri ile YSA çıkışındaki sıcaklık değerlerinin farkını göstermektedir.



Şekil 5.13: Fırın NTC uygulamalarında gizli katmanında iki nöron bulunan çok katmanlı YSA için sıcaklık ADC grafiği

Çok katmanlı ağ yapıları ile yapılan çalışmanın bir diğer uygulaması, çamaşır makinelerinde kullanılan basınç sensörü için yapılmıştır. Basınç sensörü uygulaması için de beş adet aynı ürün kodlu basınç sensörü kullanılmıştır. Beş adet sensörün her biri ile yapılan uygulamaların sonucunda Tablo 5.11 ile gösterilen veriler elde edilmiştir. Tablodaki her bir hücrede yer alan ortalama karesel hata değeri, uygulamada kullanılan beş adet sensöre ilişkin ortalama karesel hata değerlerinin aritmetik ortalamasını ifade etmektedir.



Şekil 5.14: Fırın NTC uygulamalarında gizli katmanında iki nöron bulunan sistem için referans sıcaklıklar ile her bir referans sıcaklık değerindeki farklar arasındaki grafik

Tablo 5.11: Çamaşır makinesi basınç sensörü için çok katmanlı ağ yapıları ile oluşturulan YSA yapılarına ilişkin ortalama karesel hata değerleri

Gizli Katmandaki Nöron Sayısı	O.K.H.
8	0,00646
7	0,00299
6	0,00582
5	0,00624
4	0,00655
3	0,00701
2	0,00794

Tablo 5.9, Tablo 5.10 ve Tablo 5.11 içeriğindeki veriler ile şekiller incelendiğinde aşağıdaki ortak noktalar göze çarpmaktadır.

- Çamaşır makinesi sıcaklık sensörü için ortaya çıkan ortalama karesel hata değerleri, fırın uygulaması için ortaya çıkan ortalama karesel hata değerleri ile aynı seviyededir. Çamaşır makinesi sıcaklık sensörü ile yapılan uygulamaların çalışma aralığı 20 ile 90 °C arasında, fırın uygulaması için yapılan uygulamalar ise 30 ile 180 °C arasındadır. Çamaşır makinesi sıcaklık sensörünün uygulama kapsamındaki çalışma aralığı fırın uygulamasındaki çalışma aralığına göre daha dar olmasına rağmen, yöntem ile farklı sensörlerle benzer yaklaşıklıkta lineerleştirme yapılmıştır. Benzer sonuç, Bölüm 5.4 içerisinde de görülmüştür. Buna göre sensör türleri farklı olsa da benzer türdeki yapılarda yaklaşık sonuçlar alınabilmektedir.

- b. Gizli katmanda 8–5 sayılarında nöron içeren sıcaklık sensörü uygulamalarında ortalama karesel hata değerleri birbirine yakın çıkmaktadır. Bunun temel nedeni, gizli katmanda 8 – 5 arasında nöron içeren YSA yapılarında giriş katmanı ile gizli katman arasındaki nöronlar arası ağırlıkların bir kısmının birbirine çok yakın çıkmasıdır.
- c. Çamaşır makinesi basınç sensörü uygulamasında, sıcaklık sensörü ile yapılan uygulamaların sonuçlarından farklı bir sonuç ortaya çıkmıştır. Gizli katmanda sekiz nöronun kullanıldığı koşulda ortalama karesel hata değeri, gizli katmanda daha az sayıda nöron kullanılan YSA yapılarına göre daha büyük çıkmıştır. Ortaya çıkan sonucun temel nedeni, gizli katmanda gereğinden fazla kullanılan nöronların olması sonucu ıraksamanın ortaya çıkmasıdır[1-3].
- d. Çok katmanlı ağ yapıları uygulamaları sonucunda görülen en fazla hata, gizli katmanda iki nöron kullanıldığı durumda ortaya çıkmıştır. Belirtilen yapının kullanılması sonucu yukarıda belirtilen şekillerdeki grafikler ortaya çıkmıştır. Şekiller incelendiğinde her iki sıcaklık sensörünün çalışma aralıkları boyunca sensörün ADC değerlerine karşılık olarak referans sıcaklık değerleri ile YSA çıkış değerlerinin üst üste örtüştüğü görülmüştür. Referans sıcaklıklar ile ağ çıkış sıcaklıkları arasındaki farkların dağılımı incelendiğinde farkların belirgin bir bölgeye has olmadığını, farkların rastgele olarak dağıldığını söylemek mümkün olmaktadır. Böylelikle kaynaklarda belirtilen ve YSA için sorun olarak görülen çalışma bölgelerinin uç bölgelerinde hataların artması sorunu uygulamalar sırasında görülmemiştir[1-3, 8,9].

Çok katmanlı ağlar ile yapılan uygulamalarda sistemin performansını gösterebilecek bir başka kriter, tasarım için oluşturulan yazılımın mikrodenetleyici içerisinde kapladığı alandır. Çok katmanlı YSA algoritmasının ileri yol aşamasının mikrodenetleyici içerisine girildiğinde sadece algoritma için bellekte kullanılan yerin byte cinsinden değerleri Tablo 5.12 ile verilmiştir.

Tablo 5.12: Çok katmanlı Yapay Sinir Ağları ile yapılan uygulamalar sonucu gizli katmandaki her bir nöron sayısı ile yapılan uygulamaların bellekte kapladığı alan

Gizli Katman Nöron Adedi	Bellekte Kaplanan Alan(Byte)
8	3979
7	3967
6	3955
5	3943
4	3931
3	3919
2	3907

Çok katmanlı ağlar ile yapılan uygulamalar sonucu ortaya çıkan Tablo 5.12 incelendiğinde, gizli katmanında n nöron olan YSA yapısı ile n+1 nöron olan YSA yapısı arasında bellekte kaplanan alan açısından 12 byte fark vardır. Yazılıma eklenen her bir nöronun, mikrodenetleyicide kaplanan alana katkısı %0,3' tür. Gizli katmana eklenen her bir nöronun mikrodenetleyiciye getirdiği yük açısından uygulanan YSA türleri açısından fark görülmemiştir.

5.6 Tablo Tutma Yöntemi ile Yapılan Uygulamaların Sonuçları

Tablo tutma yöntemi uygulamaları, aşağıda belirtilen sensörler ile yapılmıştır.

- Çamaşır makinesi sıcaklık sensörü
- Fırın sıcaklık sensörü

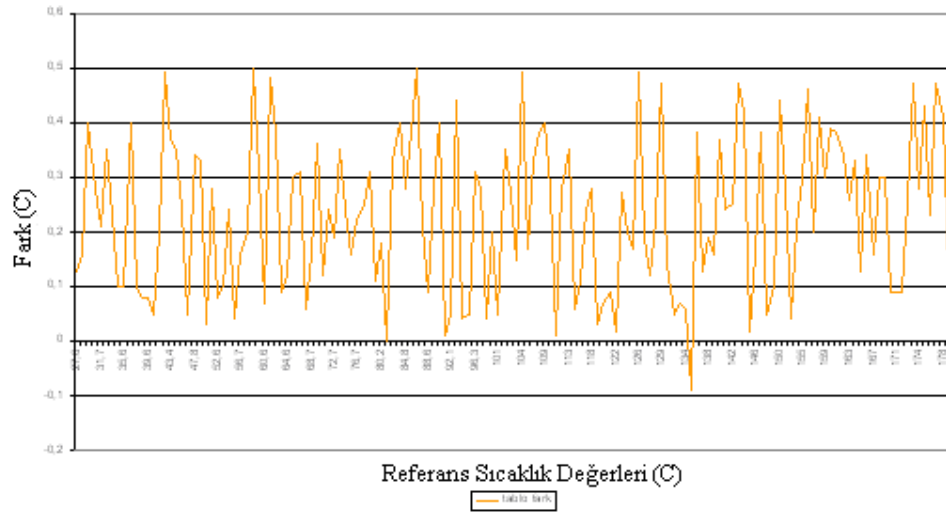
Tablo tutma yöntemi uygulamaları için oluşturulan yazılımda, pratikte yapılan uygulamalara paralel olması amacıyla, tutulan referans tablosunda her 0.5C aralıkta bir referans değer olacak biçimde veri tutulmuştur. Buna göre çamaşır makinesi sıcaklık sensörü için yapılan analizler, referans sıcaklıkların 0 ile 100 °C arasında olacağı düşünülerek tabloda tutulan 200 referans veri ile; fırın sıcaklık sensörü uygulamaları ise 0 ile 200 °C arasında çalışma aralığı olacağı düşünülerek aynı amaçla 400 veri ile yapılmıştır.

Çamaşır makinesi ve fırın sıcaklık sensörü ile yapılan çalışmaların sonucunda Tablo 5.13 ortaya çıkmıştır. Tablodaki sonuçlar, beşer adet fırın ve çamaşır makinesi sıcaklık sensörü ile yapılan uygulamaların aritmetik ortalamasını belirtmektedir. Tabloda, tutulan veri adetlerine karşılık elde edilen ortalama karesel hata değeri bulunmaktadır.

Tablo 5.13: Tablo tutma yöntemi ile yapılan uygulamalar sonucunda elde edilen ortalama karesel hata değerleri

Uygulama	Tabloda Tutulan Veri Adedi	Ortalama Karesel Hata
Çamaşı Makinesi NTC	200	0,080346
Fırın NTC	400	0,080746

Tablo tutma yöntemi ile yapılan uygulamalar sonucu Şekil 5.15 ile gösterilen fark dağılım grafiği ortaya çıkmıştır.



Şekil 5.15: Tablo tutma yöntemi ile yapılan fırın sıcaklık sensörü uygulaması sonucu referans değerler ile tabloda tutulan değerler arasındaki farkların dağılım grafiği

Tablodan görülebileceği gibi 0.5 °C aralıklarla tutulan referans verileri ile oluşturulan tablo için kullanılan her iki sensör türünde yaklaşık olarak aynı ortalama karesel hata değeri elde edilmiştir. Tasarımcı, lineerleştirme sonucunda kullandığı sensör için daha küçük ortalama karesel hata değerlerine ulaşmak istiyorsa tasarımcıya iki yol önerilebilir.

- Tuttuğu referans verilerin sıklığı 0.5 °C sıcaklıklardan küçük olmalı
- Aynı sensörler için daha küçük ortalama karesel hata değerlerine ulaşılan yöntemleri uygulayabilir. Yapılan diğer uygulamalardan görülebileceği gibi tasarımcı bu amaçla yapay sinir ağı tabanlı lineerleştirme yöntemlerine yönelebilir.

Tablo tutma yöntemi ile yapılan uygulamaların bir başka ölçütü, tablo tutma yönteminin mikrodenetleyici içerisinde kapladığı alandır. Tabloda belirtilen veri

adetlerine ek olarak 100 ile 500 arasında veri içeren bir tablonun mikrodnetleyici içerisinde kapladığı alan byte cinsinden Tablo 5.14 ile verilmiştir. Tablonun içeriğindeki her bir verinin gerçek (float) veri tipinde olduğu düşünülerek kaplanan alanlar hesaplanmıştır.

Tablo 5.14: Tablo tutma yöntemi kullanılarak yapılan uygulamalarda mikrodnetleyicide tutulan bellek miktarı

Tabloda Tutulan Veri Adedi	Bellekte Kaplanan Alan
100	701
200	1001
300	1301
400	1601
500	1901

Tablo tutma yönteminin sensör lineerleştirme uygulamalarında kullanılabileceği görülmüştür. Ancak, bu yöntemi kullanmayı hedefleyen tasarımcı, lineerleştirme amacıyla kullandığı tabloda tuttuğu veri adedi ile mikrodnetleyicide kullanılan alanın doğru orantılı olduğunu; referans veriye göre ortaya çıkan ortalama karesel hata değerinin ise belirtilen parametrelerle ters orantılı olduğunu göz önünde bulundurmalıdır.

6. SONUÇLAR VE TARTIŞMA

Sensör karakteristiğinin lineerleştirilmesi, kontrol sistemlerinde sıklıkla uygulanmaktadır. Uygulamada yapılan sistemler incelendiğinde, giriş bölümünde belirtildiği gibi, analog devreler ile yapılan sistemler, sayısal işaret işleme sistemleri ve non-lineer ADC yapıları görülmektedir. Giriş bölümünde, yöntemlerin her birinin ayrıntıları belirtilmiştir. Tasarımcılar, mikrodnetleyici içerisindeki mevcut yapıların kullanılması ile ek olarak devre tasarlama maliyetine katlanmamak, mikrodnetleyici tabanlı teknolojilerin sürekli gelişmesi ve mikrodnetleyicilerin ucuz, pratik çözümler sunabilmesi nedeni ile mikrodnetleyici tabanlı tasarımlara yönelmektedirler[9, 12,14]. Belirtilen düşünceye paralel olarak uygulanabilir bir yapı oluşturmak için tez çalışması kapsamında mikrodnetleyici tabanlı uygulamalar çalışılmıştır.

Mikrodnetleyici tabanlı sensör lineerleştirme çalışmalarında en fazla karşılaşılan uygulamalar aşağıda belirtilmiştir.

- a. Karakteristiği tam olarak bilinen bir sensör için mikrodnetleyici girişine gelen sensör verisinin, sensör karakteristiğinin ters fonksiyonundan geçirilmesi ile yapılan uygulamalar,
- b. Karakteristiği tam olarak bilinmeyen sensörler için tablo tutma yöntemi uygulamaları,
- c. Karakteristiği tam olarak bilinmeyen sensörler için yapay sinir ağı uygulamaları

Tez çalışması kapsamında, karakteristiği bilinmeyen sensörler için oluşturulan sensörlere değinilmiş; uygulamalar, tablo tutma yöntemi ve yapay sinir ağıları tekniklerine göre yapılmıştır.

Her bir uygulama türünde elde edilen sonuçlar; mikrodnetleyici üzerinde kaplanan veri alanı bilgisi, ortalama karesel hata cinsinden Bölüm 5.4, Bölüm 5.5 ve Bölüm 5.6 ile belirtilmiştir. Sonuçların yüksek doğruluklu olarak elde edilmesi için, her bir

sensör türünden aynı model numaralı beşer adet sensör kullanılmıştır. Sonuçlarla ilgili elde edilen tüm veriler, kullanılan tüm sensörlerin aritmetik ortalamasıdır.

Yapay sinir ağları tabanlı uygulamaların ortalama karesel hata değerleri aşağıdaki tablolarda verilmiştir. Tablo 6.1 ile gösterilen verilerde, çamaşır makinesi sıcaklık sensörü ile yapılan her iki YSA yöntemi uygulaması sonucu elde edilen ortalama karesel hata değerleri; Tablo 6.2 ile gösterilen verilerde ise fırın sıcaklık sensörü ile yapılan her iki YSA yöntemi sonucu oluşan ortalama karesel hata değerleri bulunmaktadır.

Tablo 6.1: Çamaşır makinesi NTC ile her iki YSA yönteminin uygulanması sonucu ortaya çıkan ortalama karesel hata tablosu

Gizli Katmandaki Nöron Sayısı	RBF tabanlı YSA	ÇKA tabanlı YSA
8	0,00077	0,00096
7	0,000785	0,00145
6	0,000982	0,00199
5	0,00148	0,00259
4	0,00185	0,00319
3	0,00212	0,00448
2	0,00235	0,00673

Tablo 6.1 ile gösterilen verilerden görülebileceği gibi çamaşır makinesi NTC sıcaklık sensörü ile her iki YSA yöntemi ile yapılan uygulamalarda, gizli katmandaki nöron sayısı arttıkça ortalama karesel hata değerinin azaldığı görülmüştür. Aynı sensörlerin kullanıldığı iki farklı YSA yöntemi kendi aralarında karşılaştırıldığında; radyal taban fonksiyonlarının kullanıldığı uygulamaların sigmoid fonksiyonlarının kullanıldığı çok katmanlı ağ uygulamalarına göre daha az ortalama karesel hataya sahip olduğu görülebilmektedir. Fırın sıcaklık sensörü ile her iki YSA yönteminin kullanıldığı Tablo 6.2 verileri incelendiğinde, Tablo 6.1 için söylenenler ile paralel sonuçların çıktığı görülmektedir.

Tablo 6.2: Fırın NTC ile her iki YSA yönteminin uygulanması sonucu ortaya çıkan ortalama karesel hata tablosu

Gizli Katmandaki Nöron Sayısı	RBF tabanlı YSA	ÇKA tabanlı YSA
8	0,00083	0,00099
7	0,00085	0,00097
6	0,00097	0,00102
5	0,00092	0,00189
4	0,00106	0,00263
3	0,0012	0,00382
2	0,00183	0,00423

Çamaşır makinesi sıcaklık sensörü ve fırın sıcaklık sensörüne ek olarak YSA tabanlı lineerleştirme algoritması çamaşır makinesi basınç sensörü için de uygulanmıştır. Uygulamalar sonucunda, Tablo 6.3 ile belirtilen veriler ortaya çıkmıştır. Tablo 6.1 ve Tablo 6.2 için söylenen yorumların aynısı frekans çıkışlı basınç sensörü için de geçerlidir. Bu yorumlara ek olarak basınç sensörü ile yapılan gizli katmanında sekiz nöronlu uygulamada, her iki YSA yönteminde ıraksama görülmüştür. Bu durum, YSA uygulamalarında gereksiz yere fazladan nöron kullanmanın ıraksamalara neden olacağı sonucunu ortaya çıkarmaktadır[1-3].

Tablo 6.3: Çamaşır makinesi basınç sensörü ile yapılan YSA tabanlı algoritmanın her iki yöntem için ortalama karesel hata değerleri

Gizli Katmandaki Nöron Sayısı	RBF tabanlı YSA	ÇKA tabanlı YSA
8	0,00526	0,00646
7	0,00292	0,00299
6	0,0047	0,00582
5	0,0056	0,00624
4	0,0062	0,00655
3	0,0064	0,00701
2	0,0067	0,00794

YSA uygulamalarına ek olarak, tablo tutma yöntemi ile uygulamalar yapılmıştır. Daha önceki bölümlerde belirtildiği gibi tablo tutma uygulamalarında çamaşır makinesi NTC sıcaklık sensörü için 0 ile 100 °C arasındaki çalışma aralığında 200; fırın NTC sıcaklık sensörü için ise 0 ile 200 °C arası için 400 referans veri ile tablo tutulmuştur. Buna göre, her 0.5 °C referans sensör değerlerine karşılık sensör tablosu tutulmuştur.

Tablo 5.13 içeriğinde görülebileceği gibi tutulan referans veri adetlerine göre ortaya çıkan ortalama karesel hata değeri 10^{-2} seviyesinde olmuştur. YSA algoritmalarının

mikrodenetleyiciye girilmesi ile bellekte kaplanan alanlar karşılaştırmalı olarak Tablo 6.4 ile verilmiştir.

Tablo 6.4: Uygulamalarda RBF ve Sigmoid fonksiyonları ile oluşturulan YSA ların mikrodenetleyici belleğinde kapladıkları alanlar

Gizli Katman Nöron Adedi	ÇKA YSA ile Bellekte Kaplanan Alan(Byte)	RBF YSA ile Bellekte Kaplanan Alan(Byte)
8	3979	4115
7	3967	4103
6	3955	4091
5	3943	4079
4	3931	4067
3	3919	4055
2	3907	4043

Tablo 6.4 verilerinden görülebileceği gibi radyal taban fonksiyonlarının kullanıldığı yapılar ile sigmoid fonksiyonlarının kullanıldığı yapıların kapladıkları alanlar açısından 136 byte fark görülmektedir. Bu fark, sigmoid fonksiyonlarına göre gaussiyen etkinleştirme fonksiyonunun derlenirken daha karmaşık olmasından kaynaklanmaktadır.

Tablo 6.4 verilerine ek olarak, tablo tutma yöntemi ile yapılan uygulamalarda, farklı referans sayıları içeren tabloların bellekte kapladıkları alan bilgisi Tablo 5.14 ile verilmiştir.

Her iki YSA yöntemi ve tablo tutma yöntemi ile yapılan uygulamalar sonucu ortaya çıkan sonuçlara göre aşağıdaki yorumlar yapılabilmektedir.

- Mikrodenetleyici ile sensör lineerleştirme amacını gerçekleştirmek için YSA yöntemleri ya da tablo tutma yöntemi kullanılabilir. Tablo tutma yöntemi ya da YSA yöntemlerinden birisini kullanma kararının verilebilmesi için ortalama karesel hata ve bellekte kaplanan alan kriterleri arasında optimizasyon yapılması gerekecektir. Tasarımcı için ortalama karesel hatanın kabul edilebilir seviyesi 10^{-2} mertebesinde ve sensörün çalışma aralığı dar ise tasarımcı tablo tutma yöntemini seçebilir. Geniş çalışma aralığına sahip bir sensörle yüksek doğrulukta ve mikrodenetleyici belleğinde daha az yer kaplayan bir tasarım için YSA yöntemlerine yönelmek önerilmektedir.
- Tablo tutma yönteminde, anahtar sensör değerleri için bir tablo oluşturulmakta ve yenileme yapılmadıkça bu değerler sabit kalmaktadır.

Yapay sinir ağıları tabanlı uygulamalarda ise ağıın eğitim aşamasında elde edilen verilere göre ağ oluşturulmakta; ağ parametreleri, eğitim yapılan sınır değerler arasında gelen ve eğitimde kullanılmayan ara değerlerde de istenilen değerleri vermiştir. Bu durum, yapay sinir ağıının genelleştirme özelliğinin sensör lineerleştirme uygulamalarına olan katkısını göstermektedir. Böylelikle, yapay sinir ağlarının kullanılması ile eğitime dahil edilmeyen sensör verileri ile de istenilen sonuca ulaşılabilir.

- c. Tablo tutma yöntemi yerine YSA yöntemlerinden birisi tercih edilmek istendiğinde Tablo 6.1 ve Tablo 6.2 incelenebilir. Tablolardan görülebileceği gibi radyal taban fonksiyonlarının kullanımı ile oluşturulan YSA, sigmoid fonksiyonunun kullanımı ile oluşturulan YSA yapılarına göre daha az ortalama karesel hatalı çıkmıştır. YSA yöntemlerinin, bellekte kapladıkları alanlar açısından birbirinden çok farklı olmadıkları görülmektedir. Tüm YSA yapılarında radyal taban fonksiyonlarının kullanımı ile oluşturulan YSA ile sigmoid fonksiyonunun kullanımı ile oluşturulan YSA arasında 136 byte değerinde bir fark görülmüştür. Tasarımcı açısından yöntemlerden birisinin tercih edilmesinin iki farklı boyutu vardır. Tasarımcı için ortalama karesel hatanın kritik olması durumunda tasarımcı radyal taban fonksiyonları ile oluşturulan YSA yapılarına yönelmelidir. Çünkü tablolar incelendiğinde oransal açıdan radyal tabanlı ağların, çok katmanlı ağlara göre daha az ortalama karesel hatalı oldukları görülebilmektedir. İkinci boyut, seçilen yöntemin bellekte kapladığı alandır. Eğer tasarımcı açısından ortalama karesel hata kritik bir öneme sahip değil ise, 136 byte ve altındaki bellek alanları tasarımcı için kritik ise bu durumda çok katmanlı ağ tabanlı YSA, tercih sebebi olabilir.
- d. Sıcaklık sensörleri ile yapılan YSA tabanlı uygulamalara göre gizli katmanda kullanılan nöron sayısı arttıkça ortalama karesel hata azalmış, buna karşılık YSA için bellekte kaplanan alan artmıştır. Basınç sensörü uygulamasında ise gizli katmanda kullanılan yüksek sayıda nöron ile sistemin ortalama karesel hata değeri, gizli katmanda daha düşük sayıda nöronun kullanıldığı uygulamalara göre daha yüksek çıkmış; ağ çıkışında ırsamalar görülmüştür. Uygulamalara göre $1*n*1$ yapısında YSA kullanılan

uygulamalarda en uygun sayıdaki n sayısı belirlenirken sistemde görülebilecek ırsamalar da dikkate alınmalıdır.

- e. Uygulamalarda, benzer karakteristikli iki farklı sıcaklık sensörü ve basınç sensörü kullanılmıştır. Sensörlerin farklı çalışma bölgelerine ve yapılarına rağmen ortaya çıkan ortalama karesel hata değerlerinin birbirine yakın olduğu görülebilir. Buna göre, YSA tabanlı uygulamalar sonucunda benzer karakteristikli sensörlere, çalışma aralıkları ve yapıları farklı olmalarına rağmen, aynı yöntemlerin uygulanması ile yaklaşık olarak aynı derecede lineerleştirme sağlanabilmektedir. Farklı çalışma karakteristiklerindeki sensörler için oluşturulan YSA ağ parametrelerinin bellekte kapladıkları alan sensörden sensöre değişmemiştir. Oysa tablo tutma yönteminin kullanılması durumunda sensörün çalışma aralığına paralel olarak bellekte kaplanan alan artmıştır.
- f. Birden fazla sensörün kullanıldığı sistemlerde, YSA uygulamaları tablo tutma yöntemi ile yapılan uygulamalara göre avantajlı durumda olduğu görülmüştür. YSA tabanlı yapılan lineerleştirmelerde söz konusu sistemde kullanılan tüm sensörler için yapay sinir ağları fonksiyonları bir kere yazılmaktadır. Bir sensörden diğer sensöre bellek alanı açısından tek fark eden, her bir sensöre ait YSA için tutulan ağırlık, ağ merkezleri gibi tablolardır. Tutulan tabloların boyutları, önceki bölümlerde belirtilmiştir. Bu durumda sistemde kullanılan sensör sayısı ile bellekte kaplanan alan doğru orantılı olarak artmamaktadır. Tablo tutma yönteminde ise her sensör için farklı tablo tutma gereksinimi vardır. Bu nedenle tablo tutma yönteminde bellekte kaplanan alan, kullanılan sensör sayısı ile doğru orantılı olarak artmaktadır.

KAYNAKLAR

- [1] **Haykin, S.**, 1999. Neural Networks, a Comprehensive Foundation, Prentice Hall, New Jersey
- [2] **Mehrotra, K., Mohan, C. K. ve Ranka, S.**, 1997. Elements of Artificial Neural Networks, MIT Pres, Massachusets
- [3] **Hagan, M. T., Demuth, H. B. ve Beale, M.**, 1995. Neural Network Design, PWS Publishing Company, Boston.
- [4] **Duman, F.ve Erdem, H.**, 2003. Hardware Implementation of Neural Networks on General Purpose Microcontrollers, IJCI Proceedings of International Conference on Signal Processing, Eylül 2003, ISSN 1304-2386, c.1, sayı 2, s. 278 – 280.
- [5] **Dempsey, G.L., Alt, N.L., Olson B.L. ve Alig, J.S.**, 1997. Control Sensor Linearization Using A Microcontroller-Based Neural Network, Proceedings of IEEE International Conference on Systems, s. 3078-3083.
- [6] **Bucci, G., Faccio M. ve Landi, C.**, 1997. The Implementation of A Smart Sensor Based on A Piece-Linear A/D Converter, IEEE Instrument and MeasurementTechnology Conference Ottawa, Kanada, Mayıs 19-21, s. 1173-1177
- [7] **Bucci, G., Faccio, M. ve Landi, C.**, 2000. New ADC with Piecewise Linear Characteristic: Case Study – Implementation of A Smart Humidity Sensor, IEEE Transactions on Instrumentation and Measurement, 49, Aralık, s. 1154 – 1166.
- [8] **Medrano-Marqués, N.J. ve Martin-del-Brio, B.**, 2000. A General Method For Sensor Linearization Based on Neural Networks, IEEE International Symposium on Circuits ans Systems, Geneva, İsviçre, Mayıs 28-31, s. 497-500.

- [9] **Medrano-Marqués, N.J. ve Martin-del-Brio, B.**, 2001. Sensor Linearization with Neural Network, IEEE Transactions on Industrial Electronics, Aralık, 48, s. 1288 – 1290.
- [10] **Amin, H., Curtis, K. M. ve Hayes-Gill, B. R.**, 1997. Piecewise Linear Approximation Applied to Nonlinear Function of a Neural Network, IEEE Proceedings on Circuits Devices Systems, Aralık, 144, s. 313-317.
- [11] **Kraska, M. P.**, 1988. Digital Linearization of Non-linear Analog(Sensor) Signals, IEEE Transactions on Instrument and Measurement, 38, Kasım, s. 45-51.
- [12] **Deosthali, A. A.**, 1998. Embeded Signal Processing on Microcontrollers, Yüksek Lisans Tezi, The University of Texas at Austin, Austin.
- [13] **Flammini, A., Marioli, D. ve Taroni, A.**, 1997. Transducer Output Signal Processing Using An Optimal Look-Up Table in Microcontroller Based Systems, Electronic Letters, 33, s. 1197-1198.
- [14] **Microchip Technology Inc.**, www.microchip.com
- [15] **Ergezer, H., Dikmen, M. ve Özdemir, E.**, 2003. Yapay Sinir Ağları ve Tanıma Sistemleri, <http://www.elyadal.org/pivolka/06/yapay.htm>
- [16] **Aleksander, I. ve Morton, H.**, 1995. An Introduction to Neural Computing, Chapman and Hall, Boston.
- [17] **Takagi, N.**, 1998. Powering by a Table Look-Up and a Multiplication with Operand Modification, IEEE Transactions on Computers, Kasım, 47, s. 1216 – 1222.
- [18] **Lopez-Martin, A. J., Zuza, M. ve Carlosena, A.**, 2003. A CMOS A/D Converter with Piecewise Linear Characteristic and Its Application to Sensor Linearization, Analog Integrated Circuits and Signal Processing, 36, s. 39-46, Kluwer Academic Publishers, Hollanda.
- [19] **Demir, T.**, 2001. Yapay Sinir Ağları ile Modelleme ve Aktif Manyetik Yataklamalı Bir Sisteme Uygulanması, *Yüksek Lisans Tezi*, İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.

EK A

Ekler için tez ile birlikte CD oluşturulmuştur. CD içeriğinde aşağıdaki bilgiler bulunmaktadır.

- a. YSA uygulamaları için bilgisayar üzerinde oluşturulan C kodları
- b. YSA ve tablo tutma uygulamaları için mikrodenetleyicide oluşturulan PIC C kodları
- c. Uygulamalarda kullanılan PIC18F252 katalog bilgileri
- d. Uygulama için tasarlanan devrenin şemaları
- e. YSA uygulamalarında ortaya çıkan ağ parametrelerini içerir dökümanlar
- f. Tablo tutma uygulamalarında kullanılan sensör dönüşüm tabloları

ÖZGEÇMİŞ

Ömer DOĞAN, 1980 yılında doğmuştur. İlköğrenimini, Yeşilyuva Hürriyet İlköğretim Okulu' nda; lise öğrenimini ise Ortaklar Anadolu Öğretmen Lisesi ve Denizli Lisesi' nde tamamlamıştır. 1998 yılında başladığı üniversite eğitimi sonucunda 2003 yılında İstanbul Teknik Üniversitesi(İTÜ) Elektronik ve Haberleşme Mühendisliği ve 2005 yılında İstanbul Teknik Üniversitesi(İTÜ) İşletme Mühendisliği lisans programlarını tamamlamıştır. 2003-2004 eğitim-öğretim yılında İTÜ Elektronik ve Haberleşme Mühendisliği Bölümü, Elektronik Mühendisliği Programı' nda yüksek lisans eğitimi alma hakkını kazanmıştır.

2003 yılından bugüne çamaşır makinesi kontrol sistemleri üzerine çalışmalar yapmaktadır.