

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**MAKİNE ÖĞRENMESİ VE DERİN ÖĞRENME YÖNTEMLERİ İLE DİZEL
MOTOR TURBO KOMPRESÖR SİSTEMİNİN MODELLENMESİ**

YÜKSEK LİSANS TEZİ

Yusuf CAN

Mekatronik Mühendisliği Ana Bilim Dalı

Mekatronik Mühendisliği Programı

HAZİRAN 2020

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**MAKİNE ÖĞRENMESİ VE DERİN ÖĞRENME YÖNTEMLERİ İLE DİZEL
MOTOR TURBO KOMPRESÖR SİSTEMİNİN MODELLENMESİ**

YÜKSEK LİSANS TEZİ

**Yusuf CAN
(518171051)**

Mekatronik Mühendisliği Ana Bilim Dalı

Mekatronik Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Metin GÖKAŞAN

HAZİRAN 2020

İTÜ, Fen Bilimleri Enstitüsü'nün 518171051 numaralı Yüksek Lisans Öğrencisi Yusuf CAN'ın, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı “MAKİNE ÖĞRENMESİ VE DERİN ÖĞRENME YÖNTEMLERİ İLE DİZEL MOTOR TURBO KOMPRESÖR SİSTEMİNİN MODELLENMESİ” başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı : **Prof. Dr. Metin GÖKAŞAN**
İstanbul Teknik Üniversitesi

Jüri Üyeleri : **Prof. Dr. Ovsanna Seta ESTRADA**
İstanbul Teknik Üniversitesi

Jüri Üyeleri : **Prof.Dr. Haluk KÜÇÜK**
Marmara Üniversitesi

Teslim Tarihi : 30 Haziran 2020
Savunma Tarihi : 23 Haziran 2020

ÖNSÖZ

Dizel motor turbo – kompresör sisteminin makine öğrenmesi ve derin öğrenme teknikleri ile modellenmesi başlıklı bu çalışmada dizel motor'un matematiksel modellenmesinde harita tabanlı modeller yerine daha kapsayıcı ve güvenilir modeller oluşturma konusu ele alınmıştır.

Tüm çalışmalarım boyunca değerli zamanını bana ayıran ve teknik bilgi birikimini benimle paylaşan saygı değer danışman hocam Prof. Dr. Metin Gökaşan'a ve bu tez çalışmamda kullandığımız dizel motor test laboratuvarının okulumuza kurulmasında katkı sağlayan tüm kişi ve kurumlara teşekkürlerimi sunarım.

Mayıs 2020

Yusuf CAN
(Mekatronik Mühendisliği)



İÇİNDEKİLER

Sayfa

ÖNSÖZ.....	v
İÇİNDEKİLER.....	vii
KISALTMALAR.....	xi
SEMBOLLER.....	xiii
ÇİZELGE LİSTESİ.....	xv
ŞEKİL LİSTESİ.....	xvii
ÖZET.....	xxi
SUMMARY.....	xxiii
1. GİRİŞ.....	1
1.1 Tezin Amacı.....	3
1.2 Literatür Araştırması.....	5
1.3 Literatür Araştırması.....	8
2. MODELLENEN SİSTEMLER VE KULLANILAN YÖNTEMLER.....	9
2.1 Dizel Motorda Turbo - Kompresör Sistemi.....	9
2.1.1 Turbo – kompresör parçaları ve görevleri.....	9
2.1.2 Turbo – kompresör kullanmanın faydaları.....	10
2.1.3 Turbo – kompresör çalışma prensibi.....	11
2.1.3 Turbo – kompresör çalışma karakteristiği.....	12
2.2 Öğrenme Algoritmaları.....	14
2.2.1 Denetimli ve yarı denetimli öğrenme.....	14
2.2.2 Denetimsiz öğrenme.....	14
2.2.3 Pekiştirmeli öğrenme.....	14
2.2.4 Diğer öğrenme algoritmaları.....	14
2.3 Makine Öğrenmesi.....	15
2.4 Derin Öğrenme.....	16

3. KULLANILAN ÖĞRENME YÖNTEMLERİNİN TEORİK ANLATIMI....	17
3.1 Polinom Regresyonu.....	17
3.2 Destek Vektör Makineleri.....	21
3.3 Regresyon Ağaçları.....	24
3.4 Yapay Sinir Ağları.....	26
3.4.1 Biyolojik sinir hücresinin yapısı.....	26
3.4.2 Yapay sinir hücresinin yapısı.....	27
3.4.2.1 Girdiler.....	27
3.4.2.2 Ağırlıklar.....	27
3.4.2.3 Toplama fonksiyonu.....	28
3.4.2.4 Aktivasyon fonksiyonu.....	30
3.4.3 Yapay sinir ağının eğitilmesi.....	34
3.4.3.1 Kayıp fonksiyonu.....	34
3.4.3.2 Gradyan iniş.....	34
3.4.3.3 Öğrenme oranı.....	36
3.4.3.4 İleriye yayılım.....	37
3.4.3.5 Geriye yayılım.....	39
3.4.3.6 Yapay sinir ağlarında kullanılan optimizasyon yöntemleri.....	41
4. KULLANILAN EĞİTİM VERİLERİ VE MODELLERİN EĞİTİMİ.....	47
4.1 Kullanılan Eğitim Verileri.....	47
4.1.1 Veri haritası 1.....	48
4.1.2 Veri haritası 2.....	48
4.1.2 Veri haritası 3.....	50
4.2 Eğitim Modellerinin Oluşturulması Ve Uygulanması.....	51
4.2.1 Polinom regresyonu yönteminin uygulanması.....	51
4.2.1.1 İkinci dereceden polinom fonksiyonun elde edilmesi.....	52
4.2.1.2 Altıncı dereceden polinom fonksiyonun elde edilmesi.....	52
4.2.2 Destek vektör makineleri yönteminin uygulanması.....	53
4.2.3 Regresyon ağaçları yönteminin uygulanması.....	53
4.2.4 Yapay sinir ağlarının oluşturulması ve eğitilmesi.....	54
4.2.4.1 Yapay sinir ağlarının oluşturulması.....	54
4.2.4.2 Yapay sinir ağlarının eğitilmesi.....	55
5. TÜM YÖNTEMLERİN SONUÇLARI.....	59
5.1 Performans Ölçütleri.....	59
5.1.1 Ortalama mutlak hata (MAE).....	59

5.1.2 Ortalama mutlak yüzde hatası (MAPE).....	60
5.1.3 Ortalama kare hatası (MSE).....	60
5.1.4 Kare Ortalamalarının Karekökü (RMSE).....	60
5.1.5 R2 Skoru (R - Squared).....	60
5.2. Polinom Regresyonu Yönteminin Sonuçları.....	61
5.2.1 İkinci dereceden polinom regresyonu ile harita 1 için sonuçlar.....	61
5.2.2 Altıncı dereceden polinom regresyonu ile harita 1 için sonuçlar.....	62
5.2.3 İkinci dereceden polinom regresyonu ile harita 2 için sonuçlar.....	64
5.2.4 Altıncı dereceden polinom regresyonu ile harita 2 için sonuçlar.....	65
5.2.5 İkinci dereceden polinom regresyonu ile harita 3 için sonuçlar.....	66
5.2.6 Altıncı dereceden polinom regresyonu ile harita 3 için sonuçlar.....	68
5.3 Karar Ağaçları Yönteminin Sonuçları.....	69
5.4 Destek Vektör Makineleri Yönteminin Sonuçları.....	71
5.5 Yapay Sinir Ağları Yönteminin Sonuçları.....	72
5.5.1 Yapay sinir ağları ile harita 1 için sonuçlar.....	72
5.5.2 Yapay sinir ağları ile harita 2 için sonuçlar.....	74
5.5.3 Yapay sinir ağları ile harita 3 için sonuçlar.....	76
KAYNAKLAR.....	79
ÖZGEÇMİŞ.....	81

KISALTMALAR

MAE	: Mean Absolute Error
MAPE	: Mean Absolute Percentage Error
MSE	: Mean Squared Error
RMSE	: Root Mean Squared Error
ANN	: Artificial Neural Network
DGS	: Diesel Generator Sets
CFD	: Computitonal Fluid Dynamic
DOE	: Design Of Experiments
RVM	: Relevance Vector Machine
YSA	: Yapay Sinir Ağları
NO_x	: Azot Oksit
CO	: Karbon Monoksit
HIL	: Hardware In The Loop
SVM	: Support Vector Machine
CART	: Classification And Regression Tree
RMS_{prob}	: Root Mean Square Error Propability
SGD	: Stochastic Gradient Descent
GNA	: Gauss Newton Algorithm
LMA	: Levenberg-Marquardt Algorithm



SEMBOLLER

$P_{in,Comp}$	: Kompresöre giren hava basıncı.
$T_{in,Comp}$	: Kompresöre giren havanın sıcaklığı.
$\dot{m}_{Comp}$	: Kompresöre giren debi.
$P_{out,Comp}$	: Kompresörden çıkan hava basıncı.
$T_{out,Comp}$	: Kompresörden çıkan havanın sıcaklığı.
$P_{in,Turbo}$	: Turboya giren hava basıncı.
$T_{in,Turbo}$	: Turboya giren havanın sıcaklığı.
$\dot{m}_{Turbo}$	: Turboya giren debi.
$P_{out,Turbo}$	: Turbodan çıkan hava basıncı.
$T_{out,Turbo}$	: Turbodan çıkan havanın sıcaklığı.
$N_{Turbo,Comp}$	: Turbo kompresör şaftının devir hızı.
η_{Turbo}	: Turbo verimi.
η_{Comp}	: Kompresör verimi.



ÇİZELGE LİSTESİ

Sayfa

Çizelge 4.1 : Birinci veri haritasının analizi.....	48
Çizelge 4.2 : İkinci veri haritasının analizi.....	49
Çizelge 4.3 : Üçüncü veri haritasının analizi.....	50
Çizelge 5.1 : Harita 1 için ikinci dereceden polinom regresyonu sonuçları.....	51
Çizelge 5.2 : Harita 1 için altıncı dereceden polinom regresyonu sonuçları.....	52
Çizelge 5.3 : Harita 2 için ikinci dereceden polinom regresyonu sonuçları.....	64
Çizelge 5.4 : Harita 2 için altıncı dereceden polinom regresyonu sonuçları.....	65
Çizelge 5.5 : Harita 3 için ikinci dereceden polinom regresyonu sonuçları.....	67
Çizelge 5.6 : Harita 3 için altıncı dereceden polinom regresyonu sonuçları.....	68
Çizelge 5.7 : Harita 3 için regresyon ağaçları modelinin sonuçları.....	69
Çizelge 5.8 : Harita 3 için destek vektör makineleri yönteminin sonuçları.....	71
Çizelge 5.9 : Harita 1 için yapay sinir ağları modelinin sonuçları.....	72
Çizelge 5.10 : Harita 2 için yapay sinir ağları modelinin sonuçları.....	75
Çizelge 5.11 : Harita 3 için yapay sinir ağları modelinin sonuçları.....	77



ŞEKİL LİSTESİ

Sayfa

Şekil 1.1 : Dört zamanlı içten yanmalı motor zamanları [10].....	1
Şekil 1.2 : Turbo-kompresör kesit görünümü.....	2
Şekil 1.3 : Basite indirgenmiş turbo-kompresör şematiği.....	4
Şekil 2.1 : Turbo-Kompresör parçaları [11].....	10
Şekil 2.2 : Turbo-Kompresör çalışma şeması [12].....	12
Şekil 2.3 : Turbo-Kompresör çalışma karakteristiği [13].....	12
Şekil 2.4 : Yapay zeka, makine öğrenmesi ve derin öğrenme.....	15
Şekil 2.5 : Makine öğrenmesi ve derin öğrenme arasındaki fark.....	16
Şekil 3.1 : Epsilon yoğun bant ile bir boyutlu doğrusal regresyon.....	21
Şekil 3.2 : Doğrusal olmayan regresyon fonksiyonu.....	22
Şekil 3.3 : Girdi örneklerinin doğrusal olmayan haritasının yüksek boyutlu özellik uzayına dönüştürülmesi.....	23
Şekil 3.4 : Örnek bir karar ağacı yapısı.....	25
Şekil 3.5 : Biyolojik sinir hücresinin temel yapısı.....	27
Şekil 3.6 : Yapay sinir hücresinin temel yapısı.....	28
Şekil 3.7 : Basamak fonksiyonu ve türevi.....	30
Şekil 3.8 : Doğrusal fonksiyon ve türevi.....	31
Şekil 3.9 : Sigmoid fonksiyonu ve türevi.....	31
Şekil 3.10 : Hiperbolik tanjant fonksiyonu ve türevi.....	32
Şekil 3.11 : ReLU fonksiyonu ve türevi.....	32
Şekil 3.12 : Sızıntı ReLU fonksiyonu ve türevi.....	33
Şekil 3.13 : Softmax fonksiyonu ve türevi.....	33
Şekil 3.14 : Öğrenme oranına karşı doğruluk grafiği.....	36
Şekil 3.15 : Dört giriş tek çıkışlı örnek bir yapay sinir ağı.....	37
Şekil 3.16 : Sinir ağında geri yayılımın görsel temsili.....	40
Şekil 3.17 : Toplu, stokastik ve mini gradyan iniş yöntemleri.....	43

Şekil 4.1 : Sıkıştırma oranı ve devir hızına bağlı olarak debiyi gösteren üç boyutlu turbo veri haritası.....	48
Şekil 4.2 : Devir hızı ve debiye bağlı olarak sıkıştırma oranını gösteren üç boyutlu kompresör veri haritası.....	49
Şekil 4.3 : Devir hızı ve debiye bağlı olarak kompresörün verimini gösteren harita.....	50
Şekil 4.4 : Harita 3 için elde edilen regresyon karar ağacı modeli.....	53
Şekil 4.5 : Harita 1 ve 2 için oluşturulan yapay sinir ağı modeli.....	54
Şekil 4.6 : Harita 3 için oluşturulan yapay sinir ağı modeli.....	55
Şekil 4.7 : Harita 1 için eğitim performansı grafiği.	56
Şekil 4.8 : Harita 2 için eğitim performansı grafiği.	56
Şekil 4.9 : Harita 3 için eğitim performansı grafiği.	57
Şekil 4.10 : Harita 1 için eğitim validasyon grafiği.	57
Şekil 4.11 : Harita 2 için eğitim validasyon grafiği.	58
Şekil 4.12 : Harita 3 için eğitim validasyon grafiği.	58
Şekil 5.1 : Harita 1 için ikinci dereceden polinom R^2 skorları.	62
Şekil 5.2 : Harita 1'in tüm verileri ile ikinci dereceden polinom regresyonun çıktısı.	62
Şekil 5.3 : Harita 1 için altıncı dereceden polinom R^2 skorları.	63
Şekil 5.4 : Harita 1'in tüm verileri ile altıncı dereceden polinom regresyonun çıktısı.....	63
Şekil 5.5 : Harita 2 için ikinci dereceden polinom R^2 skorları.	64
Şekil 5.6 : Harita 2'nin tüm verileri ile ikinci dereceden polinom regresyonun çıktısı.....	65
Şekil 5.7 : Harita 2 için altıncı dereceden polinom R^2 skorları.....	66
Şekil 5.8 : Harita 2'in tüm verileri ile altıncı dereceden polinom regresyonun çıktısı.....	66
Şekil 5.9 : Harita 3 için ikinci dereceden polinom R^2 skorları.	67
Şekil 5.10 : Harita 3'ün tüm verileri ile ikinci derecen polinom regresyonun çıktısı.....	67
Şekil 5.11 : Harita 3 için altıncı dereceden polinom R^2 skorları.	68
Şekil 5.12 : Harita 3'ün tüm verileri ile altıncı derecen polinom regresyonun çıktısı.....	69
Şekil 5.13 : Harita 3 için regresyon ağaçları modelinin R^2 skorları.....	70
Şekil 5.14 : Regresyon ağaçları modelinin harita 3 verileri ile üç boyutlu görseli....	70

Şekil 5.15 : Harita 3 için destek vektör makineleri yönteminin R2 skorları.....	71
Şekil 5.16 : Destek vektör makinelerinin harita 3 verileri ile üç boyutlu görseli.....	72
Şekil 5.17 : Harita 1 için yapay sinir ağları ile elde edilen R2 skorları.....	73
Şekil 5.18 : Harita 1'in tüm verileri ile Y.S.A modelinin çıktısı.....	74
Şekil 5.19 : Harita 2 için yapay sinir ağları ile elde edilen R2 skorları.....	75
Şekil 5.20 : Harita 2'in tüm verileri ile Y.S.A modelinin çıktısı.....	76
Şekil 5.21 : Harita 3 için yapay sinir ağları ile elde edilen R2 skorları.....	77
Şekil 5.22 : Harita 3'in tüm verileri ile Y.S.A modelinin çıktısı.....	78





MAKİNE ÖĞRENMESİ VE DERİN ÖĞRENME YÖNTEMLERİ İLE DİZEL MOTOR TURBO KOMPRESÖR SİSTEMİNİN MODELLENMESİ

ÖZET

Dizel motorlar, 130 yıla yakın bir süredir kullanılmaktadır. Dizel motorlarda kullanılan motorin yakıtının genellikle benzin fiyatına göre daha ucuz olması, aynı özelliklere sahip olan benzinli bir motorla kıyaslandığında yaklaşık olarak %30 daha az yakıt harcaması, motorinin tutuşma sıcaklığının benzinin tutuşma sıcaklığından daha yüksek olduğu için yangın tehlikesinin daha az olması, benzinli motorlarda verimin %25-30 civarında olmasına rağmen dizel motorlarda bu değer %40 oranında olması gibi önemli avantajlar sebebiyle dizel motorlar günümüzde çok fazla tercih edilmektedir.

Dizel motorların tarihi 18.yüzyılın sonlarına dayanıyor olsa da dizel motorların bilgisayar ortamında modellenmesi ve kontrolü yaklaşık olarak son 20-30 yıllık bir dönemi kapsar. Dizel motorun bilgisayar ortamında matematiksel modellenmesi ile bu dizel motorun sahip olduğu fiziksel yapı bilgisayar ortamına aktarılarak gerekli hesaplamalar ve testler yapılabilir. Burada tüm fiziksel ve kimyasal süreçler kapsamlı bir şekilde ele alınabilir. Modelleme yapılırken motorun sahip olduğu bütün özellikler doğru biçimde bilgisayara aktarılmalıdır. Motoru doğru bir şekilde modellemek, motorun otomatik kontrolü için motorun en verimli ve etkin şekilde çalışmasını sağlayan kontrolörlerin en uygun biçimde tasarlanmasını, laboratuvar ortamında oluşturulamayan fiziksel şartları bilgisayar ortamında simüle etmeyi ve diğer birçok test ve hesaplamaların yapılmasını sağlar.

Motor modeli oluşturulurken birçok fiziksel bölümün karşılığı matematiksel olarak ifade edilebilmektedir. Fakat bu işlemler çok karmaşık ve zorlu olabilmektedir. Bunun yanında bazı alt sistemlerin matematiksel olarak bilgisayar ortamına aktarılması neredeyse imkansızdır. Böyle durumlarda modellenecek olan motor üzerinden laboratuvar ortamında gerçek zamanlı veriler alınarak bu verilerden oluşan haritalar elde edilir ve modele eklenir. Fakat bu işlem için çok yüksek maliyetli bir laboratuvar ortamı gerekmektedir. Okulumuzda bu sistem mevcut olduğu için modellenecek olan motorun gerçek zamanlı verilerini elde edebiliyoruz. Ancak motor üzerinden toplanan veriler bazı durumlarda sınırlı olabilmektedir. Örneğin düşük basınçta motorun nasıl çalışacağını bu haritalar üzerinden göremeyiz. Kısaca bu haritalar üzerindeki veriler çok kapsamlı olmayabilir. Harita tabanlı modellerin çok kapsayıcı olmamasının yanı sıra sürekli zamanlı olarak her noktayı temsil edememe gibi dezavantajları da vardır. Harita üzerindeki ara değerler genellikle enterpolasyon yapılarak tahmin edilir. Harita tabanlı modelleri ortadan kaldırmak ve bunun yerine daha kapsayıcı ve sürekli zamanlı modeller oluşturabilmek bu tez çalışmasının temel hedefi olmuştur.

Bu tez çalışmasında turbo-kompresör sisteminin karmaşık yapısını bilgisayar ortamına aktarmayı sağlayan, gerçek zamanlı veriler ile eğitilen makine öğrenmesi ve derin öğrenme yöntemleri kullanılarak modeller oluşturulmuştur. Turbo kompresör sistemi modellenirken farklı alt bloklardan oluşmaktadır. Mevcut modelde turbo kompresör sistemi için farklı haritalar kullanılmaktadır. Bu tez çalışmasında bu haritaları ortadan kaldırmak ve bunların yerine turbo kompresör sistemini temsil edecek daha kapsayıcı, daha güvenilir modeller koymak esas alınmıştır. Bu tez çalışması kapsamında debi, sıkıştırma oranı ve verimi temsil eden 3 farklı harita ele alınmıştır. Polinom regresyonu yöntemi ve yapay sinir ağları ile modelleme yöntemi bu 3 veri haritasını da modellemek için ayrı ayrı kullanılmıştır. Bu yöntemlerin yanında destek vektör makineleri yöntemi ve karar ağaçları yöntemi sadece harita 3'ü modellemek için kullanılmıştır. Harita 3 diğer haritalara kıyasla daha karmaşık bir yapıya sahip olduğu için daha fazla yöntem kullanarak bu yöntemlerin sonuçları üzerinde durulmuştur.

Tüm bu yöntemlerin sonuçlarını özetleyecek olursak dizel motor turbo kompresör sisteminin makine öğrenmesi ve derin öğrenme yöntemlerini kullanarak modellenmesinde farklı yöntemlerle başarılı sonuçlar alınmıştır. Polinom regresyonu yönteminde farklı dereceden polinomlar üretilmiş ve yüksek dereceli polinomlarda daha başarılı sonuçlar elde edilmiştir. Karar ağaçları yönteminde ise destek vektör makineleri yöntemine göre daha başarılı sonuçlar elde edilmiştir. Ancak karar ağaçları yönteminin yine harita benzeri sürekli zamanlı olmayan bir yapısı olduğunu da göz önünde bulundurmakta fayda vardır. Bu tez çalışmasında yapay sinir ağları kullanılarak elde edilen modellerin diğer yöntemlere göre daha başarılı sonuçları olmuştur. Bu tez çalışmasında üç veri haritası için de eğitim verilerinin sayısı yeterli olduğu için yapay sinir ağları başarılı olmuştur. Fakat yapay sinir ağları eğitilirken özellikle de çok katmanlı yapay sinir ağları için eğitim verilerinin sayısının çok olmasına ihtiyaç duyulur. Eğer eğitim veri sayısı yeterli düzeyde değilse diğer makine öğrenmesi yöntemlerinin daha başarı olabileceği de unutulmamalıdır.

MODELING OF DIESEL ENGINE TURBO COMPRESSOR SYSTEM WITH MACHINE LEARNING AND DEEP LEARNING METHODS

SUMMARY

Diesel engines have been used for nearly 130 years. The diesel fuel used in diesel engines is generally cheaper than the gasoline price, approximately 30% less fuel consumption compared to a gasoline engine with the same features, less fire hazard because the engine's ignition temperature is higher than the ignition temperature of gasoline, although the efficiency of gasoline engines is around %25-30 on the other hand this value is about %40 in diesel engines, diesel engines are very preferred today because of the these important advantages.

After the advantages of the diesel engine, let's talk briefly about the turbo compressor system, another important subject of this thesis study. This improvement over the power output of a naturally aspirated engine is due to the compressor forcing more air and proportionally more fuel to the combustion chamber than just atmospheric pressure. The purpose of a turbo is to increase engine efficiency by increasing the density of the suction gas (usually air), thereby providing more power per engine revolution. The compressor of the turbocharger draws ambient air and compresses it before entering the intake manifold at increased pressure. This causes more air to enter the cylinders at each entry time. The power required to rotate the centrifugal compressor is derived from the kinetic energy of the engine's exhaust gases. A turbo can also be used to increase fuel efficiency without increasing power. This is achieved by recovering waste energy from the exhaust and feeding it back into the engine inlet. By using this otherwise wasted energy to increase the mass of the air, it becomes easier to burn all the fuel before it is ventilated at the beginning of the exhaust phase. Due to these benefits, turbo compressors are widely used in diesel engines and recently in gasoline engines.

Although the history of diesel engines dates back to the end of the 18th century, modeling and control of diesel engines in computer environment covers a period of approximately the last 20-30 years. With the mathematical modeling of the diesel engine in the computer environment, the necessary physical calculations and tests can be made by transferring the physical structure of this diesel engine to the computer environment. All physical and chemical processes can be comprehensively covered here. While modeling, all features of the engine should be transferred to the computer correctly. Modeling the engine correctly enables the controllers to be designed in the most efficient and efficient way for automatic control of the engine, to simulate the physical conditions that cannot be created in the laboratory, and many other tests and calculations.

While creating the engine model, the equivalent of many physical parts can be expressed mathematically. However, these processes can be very complex and challenging. In addition, it is almost impossible to transfer some subsystems to computer environment mathematically. In such cases, real-time data is obtained in the laboratory environment via the engine to be modeled, and maps consisting of these data are obtained and added to the model. However, this process requires a very high cost laboratory environment. Since this system is available in our school, we can obtain real-time data of the engine to be modeled. However, data collected over the engine may be limited in some cases. For example, we cannot see how the engine will work at low pressure through these maps. In short, the data on these maps may not be very comprehensive. In addition to being not very inclusive, map-based models also have disadvantages such as not being able to represent every point on a continuous basis. Intermediate values on the map are usually estimated by interpolation. The main goal of this thesis is to eliminate map-based models and to create more inclusive and continuous-time models instead. In the literature, mostly models related to combustion reactions outweigh using ANN. The number of studies on modeling of turbo-compressor systems using artificial neural networks is almost nonexistent.

In this thesis, models were created by using machine learning and deep learning methods, trained with real-time data, to transfer the complex structure of the turbocharger system to the computer environment. When modeling the Turbo compressor system, it consists of different sub-blocks. In the current model, different maps are used for the turbocharger system. In this thesis, it is based on eliminating these maps and replacing them with more inclusive, reliable models that will represent the turbocharger system. Within the scope of this thesis, 3 different maps representing flow, compression ratio and efficiency are discussed. Polynomial regression method and modeling method with artificial neural networks were used separately to model these 3 data maps. Besides these methods, support vector machines method and decision trees method were used only to model map 3. Since Map 3 has a more complex structure compared to other maps, the results of these methods are emphasized by using more methods.

If we briefly summarize the application of the methods used. In this thesis, two different polynomials, second and sixth degrees, were produced by applying normalization process to the input values for 3 maps and not normalizing process. While producing second order polynomials, applying normalization to input values did not contribute to improvement. However, when the degree of the polynomial increased, for example, in this thesis study, applying the normalization process to the input values for the sixth degree polynomials significantly increased the learning process. The input values are averaged from the input values for the normalization process. Then normalization was performed by dividing the standard deviation of the input values by three times. Since learning is provided by normalization process, the same process is applied to the data value during estimation using the average and standard deviation values obtained in this normalization and the result is calculated according to these values. In this thesis, Scikit Learn library was used to produce polynomials. The obtained polynomials were arranged in Matlab program and the results were obtained.

If we look at the regression method with the support vector machines applied for Map 3. Firstly, three quarters of the data, which is 35100, the rest of the training is reserved for testing. The data were not normalized and the data was mixed randomly. The functions of the regression support vector machines in the Matlab program were used. Gaussian function was preferred as the kernel. With non-normalized data, it was seen that the education was longer and the results were worse. If the model is trained by normalizing the data, the raw data should be pretreated according to these normalization values and given to the network. However, if we send the non-normalized data into the `fitcsvm` function used in the Matlab program and activate the standardization process, there is no need to pre-process the data in the prediction phase, since it will perform normalization in itself.

If we look at the regression trees applied for Map 3. Firstly, three quarters of the data, which is 35100, the rest of the training is reserved for testing. The data were not normalized and the data was mixed randomly. Using the functions in the Matlab program, a regression decision tree model consisting of 2165 nodes was obtained.

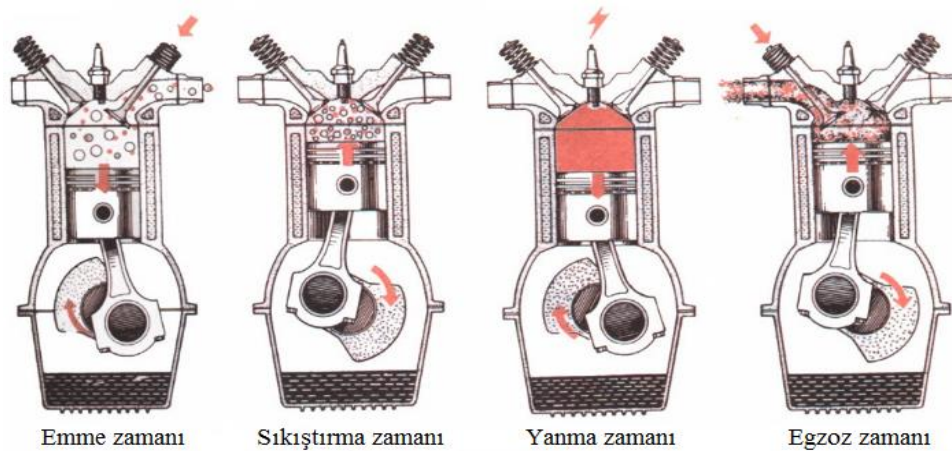
In the Matlab program for Maps 1 and 2, an artificial neural network model consisting of 32 neuron input layers, 8 hidden layers and 1 exit layer is created. One of the activation functions, hyperbolic tangent was used for the first two layers. Since Map 3 has a more complex structure, the same model has not been shown to be very effective and an additional layer with 4 neurons has been added to increase the accuracy. In addition, since softmax activation functions for map 3 give better results, hyperbolic tangent activation functions were preferred in the first three layers. Levenberg-Marquardt was chosen as the result of all trials in the optimization method in all maps. Learning rate is 0.01, verification control number is 10, maximum iteration number is 5000 and learning target value 10^{-15} .

If we summarize the results of all these methods, successful results were obtained with different methods in modeling the diesel engine turbo compressor system using machine learning and deep learning methods. In polynomial regression method, different degrees of polynomials have been produced and more successful results have been obtained in high grade polynomials. In decision trees method, more successful results were obtained compared to support vector machines method. However, it is also worth considering that the decision trees method has a map-like non-continuous structure. In this thesis study, models obtained by using artificial neural networks have had more successful results than other methods. In this thesis, artificial neural networks were successful because the number of training data was sufficient for all three data maps. However, when training neural networks, there is a need for a large number of training data, especially for multi-layered artificial neural networks. If the number of training data is not sufficient, it should be remembered that other machine learning methods may be more successful.



1. GİRİŞ

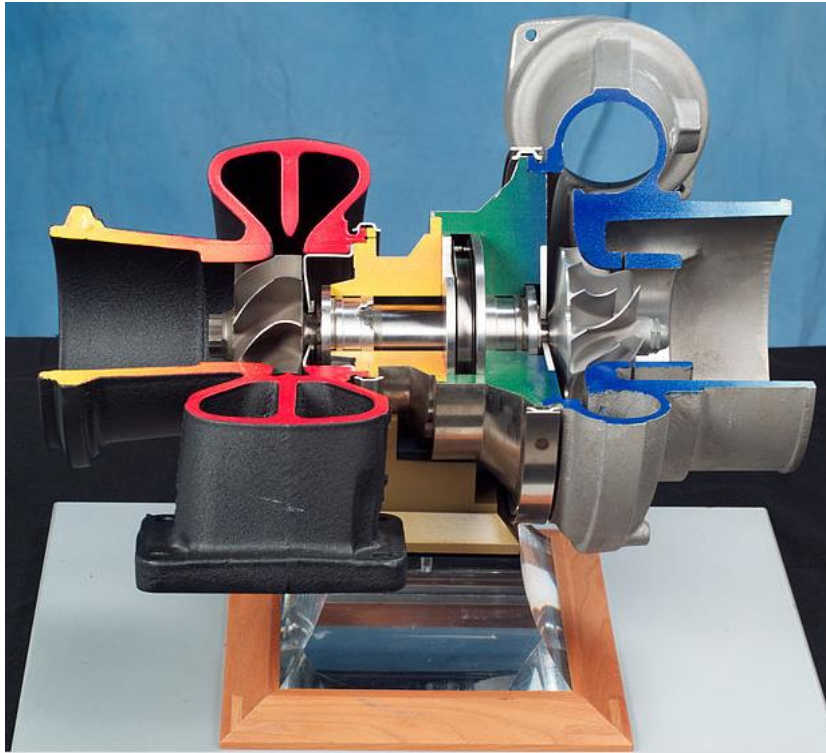
Dört zamanlı bir motor, pistonun krank milini döndürürken dört ayrı aşamayı tamamladığı bir içten yanmalı motordur. Buradaki zaman tabiri, pistonun her iki yönde silindir boyunca tam hareketini ifade eder. Dört ayrı vuruş olarak adlandırılır. Emme zamanı indüksiyon veya emme olarak da bilinir. Pistonun bu zamanı üst ölü merkezde başlar ve alt ölü merkezde biter. Bu zamanda, piston aşağı doğru hareketiyle silindire vakum basıncı üreterek piston bir hava-yakıt karışımını silindire çekerken giriş valfi açık konumda olmalıdır. Pistona karşı aşağı doğru hareketle hava emilirken piston aşağı doğru hareket eder. Daha sonra sıkıştırma zamanına geçilir. Bu zamanda piston, hava-yakıt karışımını sıkıştırır. Bu aşamada hem emme hem de egzoz valfleri kapalıdır. Üçüncü zaman yanma zamanıdır. Güç veya ateşleme olarak da bilinir. Bu, dört zamanlı döngünün ikinci devriminin başlangıcıdır. Bu noktada krank mili tam 360 derecelik bir devrini tamamlar. Basıncı hava-yakıt karışımı, bir buji (bir benzinli motorda) veya yüksek sıkıştırma (dizel motorlar) tarafından üretilen ısı ile ateşlenir ve pistonu güçlü bir şekilde alt ölü noktaya ilerler. Bu zamanda, krank milini döndürmek için motordan mekanik işler üretir. Son zaman ise egzoz zamanıdır. Piston alt ölü noktadayken egzoz valfi açılır ve piston üst ölü noktaya ilerlerken kullanılmış hava-yakıt karışımı tahliye edilir. Şekil 1.1’de görsel olarak bu zamanlar görülebilir.



Şekil 1.1 : Dört zamanlı içten yanmalı motor zamanları [24].

Teorik olarak, dizel motorlar ve benzinli motorlar oldukça benzerdir. Her ikisi de yakıtta bulunan kimyasal enerjiyi mekanik enerjiye dönüştürmek için tasarlanmış içten yanmalı motorlardır. Bu mekanik enerji pistonları silindirlerin içinde yukarı ve aşağı hareket ettirir. Pistonlar bir krank miline bağlanır ve doğrusal hareket olarak bilinen pistonların yukarı ve aşağı hareketi, bir arabanın tekerleklerini ileri doğru döndürmek için gereken dönme hareketini yaratır. Hem dizel motorlar hem de benzinli motorlar, bir dizi küçük patlama veya yanma yoluyla yakıtı enerjiye dönüştürür. Dizel ve benzin arasındaki en büyük fark, bu patlamaların gerçekleşme şeklidir. Benzinli bir motorda yakıt havayla karıştırılır, pistonlarla sıkıştırılır ve bujilerden gelen kıvılcımlar ile ateşlenir. Ancak dizel motorda önce hava sıkıştırılır ve ardından yakıt enjekte edilir. Sıkıştırıldığında hava ısınır, yakıt tutuşur.

Bir turboşarj diğer adıyla turbo, yanma odasına ekstra hava zorlayarak bir içten yanmalı motorun verimliliğini ve güç çıkışını artıran türbin tahrikli bir zorunlu endüksiyon cihazıdır. Doğal emişli bir motorun güç çıkışı üzerindeki bu gelişme, kompresörün yanma odasına sadece atmosfer basıncından daha fazla hava ve orantılı olarak daha fazla yakıt zorlamasından kaynaklanmaktadır. Şekil 1.2’de bir turbo şarjın kesit görü yer almaktadır.

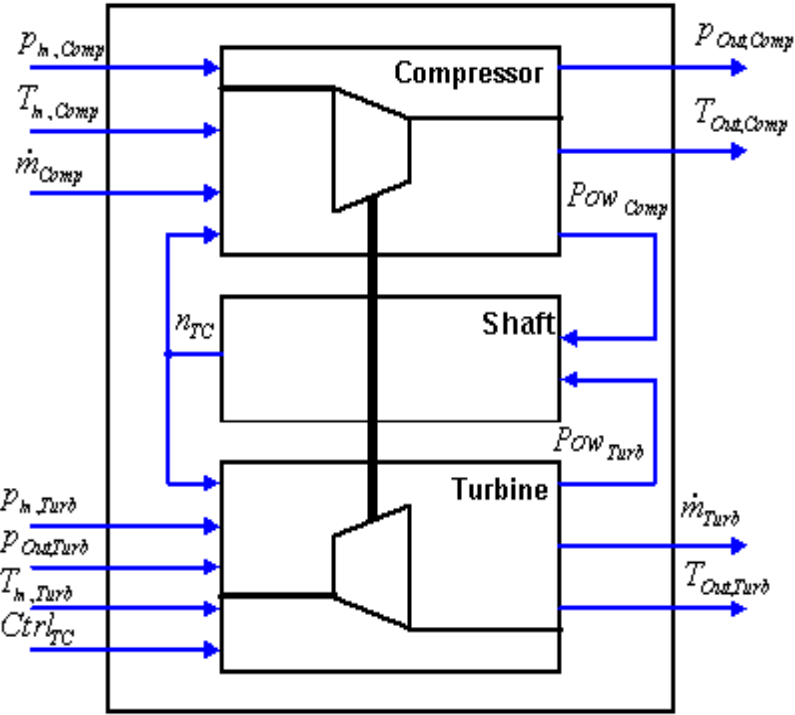


Şekil 1.2 : Turbo-kompresör kesit görünümü [25].

Bir turbonun amacı, emme gazının (genellikle hava) yoğunluğunu artırarak motor verimliliğini artırmak ve böylece motor devri başına daha fazla güç sağlamaktır. Turbo şarjının kompresörü ortam havasını çeker ve artan basınçta emme manifolduna girmeden önce sıkıştırır. Bu, her giriş zamanında silindirlere daha fazla hava girmesine neden olur. Santrifüj kompresörü döndürmek için gereken güç, motorun egzoz gazlarının kinetik enerjisinden elde edilir. Bir turbo, gücü arttırmadan yakıt verimliliğini artırmak için de kullanılabilir. Bu, egzozdaki atık enerjinin geri kazanılması ve motor girişine geri beslenmesiyle elde edilir. Havanın kütlesini arttırmak için bu başka şekilde boşa harcanan enerjiyi kullanarak, egzoz aşamasının başlangıcında havalandırılmadan önce tüm yakıtın yakılmasını sağlamak daha kolay hale gelir. İlerleyen bölümlerde bir turbo şarjın diğer adıyla turbo veya turbo-kompresör sisteminin nasıl çalıştığı detaylı olarak anlatılacaktır. Şimdilik kesit görünümde yer alan parçaları incelemek yeterli olacaktır.

1.1 Tezin Amacı

Bu tez çalışması bir dizel motorun matematiksel modellenmesine katkı sağlamak için yapılmıştır. Dizel motoru bilgisayar ortamında matematiksel modellemek, bu motorun sahip olduğu fiziksel yapıyı bilgisayar ortamına aktarmak demektir. Burada tüm fiziksel ve kimyasal süreçler kapsamlı bir şekilde ele alınmaktadır. Motorun sahip olduğu tüm özellikler doğru biçimde bilgisayara aktarılmalıdır. Motorun bilgisayar ortamında modellenmesi bize, motor en verimli şekilde nasıl çalışır, bu motorun kontrolörlerinin tasarımı nasıl olmalıdır veya motor farklı fiziksel ortamlara nasıl tepki verir gibi zor soruları bilgisayar ortamında cevaplamamızı sağlar. Motor modeli Matlab/Simulink programı üzerinde oluşturulmaktadır ve çok katmanlı bir yapıya sahiptir. Motordaki tüm alt sistemler giriş ve çıkışlara sahip bloklar halinde programa aktarılmaktadır. Her blok kendi fiziksel yapısını temsil etmektedir. Örneğin şekil 1.3’de en basite indirgenmiş giriş ve çıkışlara sahip olarak ifade edilen bir turbo – kompresörün şematik gösterimi yer almaktadır. Gerçek model üzerinde düşünüldüğünde ise çok daha karmaşık ve detaylı olarak ifade edilmektedir. Giriş çıkış değerlerinin ne anlama geldiği bu bölümde çok önemli olmadığı için ne anlama geldiği ileriki bölümlerde detaylı açıklanacaktır. Temel olarak bloğa giriş ve çıkışları görebiliriz her blok kendi içerisinde başka giriş çıkış değerlerine sahip olabilmektedir.



Şekil 1.3 : Basite indirgenmiş turbo-kompresör şematığı.

Motor modeli oluşturulurken birçok bölümün modeli matematiksel olarak ifade edilebilmektedir. Ancak bu işlem çok karmaşık ve zorlu olabilmektedir. Ayrıca bazı alt sistemlerin matematiksel olarak bilgisayar ortamına aktarılması neredeyse olanaksızdır. Bu gibi durumlarda modellenen motor üzerinden gerçek zamanlı veriler alınır ve bu verilerden oluşan bir harita elde edilir. Bu haritalar modele eklenir. Ancak bu işlem için çok yüksek maliyetli bir laboratuvar ortamı gerekmektedir. Okulumuzda bu sistem kurulduğu için modellenen motorun gerçek zamanlı verilerine sahip olabiliyoruz. Motor üzerinden toplanan veriler bazı durumlarda kısıtlı olabilmektedir. Örneğin düşük basınçta motorun nasıl çalışacağını bu haritalar üzerinden göremeyiz. Kısaca bu haritalar üzerindeki veriler çok kapsamlı olmayabilir.

Bu tez çalışması da bu problemi çözmek için gerçekleştirilmiştir. Bu tez çalışmasının temel hedefi dizel motorda kullanılan turbo-kompresör sisteminin karmaşık yapısını bilgisayar ortamına aktaracak kapsayıcı modeller oluşturmaktır. Bu modelleri oluşturmak içinde gerçek zamanlı verilerden faydalanılarak makine öğrenmesi ve derin öğrenme yöntemlerini kullanmaktır. Böylece en iyi sonucu veren modeli bilgisayar ortamına eklemektir. Turbo kompresör gibi modellenmesi güç sistemleri modellemeyi sağlamaktır.

1.2 Literatür Araştırması

1890'larda Rudolf Diesel, adını taşıyan verimli, sıkıştırma ateşlemeli, içten yanmalı bir motor icat etti. İlk dizel motorlar büyüktü ve basınçlı hava destekli yakıt enjeksiyon sistemlerinin sınırlamaları nedeniyle düşük hızlarda çalıştırıldılar. İlk yıllarında dizel motor, Akroyd-Stuart tarafından icat edilen sıcak ampul motoru olan başka bir ağır akaryakıt motoru konseptiyle rekabet ediyordu. 1920'lerde ticari araç uygulamaları için yüksek hızlı dizel motorlar ve 1930'larda da binek otomobiller için dizel motorlar tanıtıldı. Bu zamanlarda Gottlieb Daimler (1885) ve Rudolf Diesel (1896) motora verilen havanın önceden sıkıştırılmasıyla motor gücünün artacağını ve yakıt tüketimi azaltacağını tespit etti ve turbo sistemlerinin içten yanmalı motorlarda bu tarihlerden itibaren kullanmaya başladılar. Yani turboşarjlı motorun geçmişi de neredeyse içten yanmalı motorun kendisi kadar eskidir.

Dizel motorun tarihi 18.yüzyılın sonlarına dayanıyor olsa da dizel motorun bilgisayar ortamında modellenmesi ve kontrolü son 20-30 yıllık bir dönemi kapsar. Yapay sinir ağlarının doğrusal olmayan karmaşık problemleri çözme kabiliyeti ve bununla birlikte getirdiği birçok avantaj sebebiyle içten yanmalı motorların modellenmesi alanında kullanımı 1986 yılına kadar dayanmaktadır. Dizel motorun modellenmesi ile ilgili literatürde birçok yayın olmasına karşın bu tez çalışmasında sadece makine öğrenmesi ve derin öğrenme yöntemlerini içeren dizel motor modelleme çalışmaları ortaya konulacaktır.

1993 yılında Morita içten yanmalı motorların yanma parametrelerini optimize etmek amacıyla yaptığı çalışmasında geleneksel tablo inceleme yöntemi ile bulunan yanma parametrelerinin lineer olmayan karakteristiklerinin elde edilmesi rolünü YSA ile değiştirmiştir. Bunun dışında diğer tablo inceleme yöntemine göre avantaj ve dezavantajlarını sorgulamıştır. YSA ile yanma parametrelerinin gerçek zamanlı kontrol edilmesi ortaya konulmuştur. Sonuç olarak YSA'nın diğer yöntemlere göre daha uygun olduğu kanıtlanmıştır. [1]

1996 yılında Howlett ve arkadaşları yaptıkları bir çalışmada YSA'nın benzinli içten yanmalı motorlarda durum izleme veya teşhis edilmesi için uygun bir teknik içerisinde kullanılmasını ortaya koymuştur. Elde edilen modelin çıktısı ile yüzde 99 oranında sınıflandır doğruluğuna ulaşılmıştır. [2]

1997 yılında Muller ve arkadaşları algılayıcılardan gelen verileri işleyerek yanma zamanının kontrolü için kullanmışlardır ve böylece ölçülen verilerle YSA eğitilmiş ardından form faktörleri belirlemiştir. Elde edilen bu yapay sinir ağı ile gerçek zamanlı işlemler hesaplanabilir ve çok az bir çaba harcayarak yeni motorlara uygulanabilir olduğu ortaya konulmuştur. [3]

1998 yılında Ortmann ve Glesner içten yanmalı motorlarda meydana gelen vuruş (knock) olayının tespit edilmesi için bir YSA modeli geliştirmişlerdir. Bilindiği gibi yakıt tüketiminin ve emisyonun azaltılması ve performansın artırılması için motorlarda sıkıştırma oranı artırılır. Ancak bunun ile birlikte motor vuruş zamanı ve uygun motor kontrol parametrelerinin elde edilmesi gerektirmektedir. Yapılan bu çalışmada oluşturulan YSA ile her bir devirde vuruş zamanının bulunup ortaya çıkarılmasındaki rolü anlatılmıştır. [4]

2001 yılında Heister ve Froehlich yanma ve basınç verilerini kullanarak bir YSA modeli oluşturmuşlar. Bu çalışmada yapay sinir ağlarının avantajları ve dezavantajları üzerinde durulmuştur. Örneğin YSA'nın eğitimi sırasında giriş çıkış değerleri arasında fonksiyonel bağımlılık olmadığı durumda ağın yanlış eğitileceği üzerinde durulmuştur. Bir yanma devri boyunca meydana gelen ve sadece yanma odasındaki basınç değerini gösteren bir noktada gerçekleşen %50'lik enerji dönüşüm noktasının tespiti YSA ile tahmin edilmiştir. [5]

2002 yılında Yongsheng He ve arkadaşları, turboşarjlı bir dizel motoru modellemek için YSA modeli kullanıp sonuçlarını paylaşmıştır. Motor, çok çeşitli çalışma koşullarında CFD kodu (KIVA-ERC) kullanılarak simüle edilmiş ve yapay sinir ağlarını eğitmek için sayısal simülasyon sonuçları kullanılmıştır. En karakteristik motor çalışma koşullarını ve dolayısıyla yapay sinir ağlarını eğitmek için en bilgilendirici verileri seçmek için deney tasarımı (DOE) tekniklerini kullanan verimli bir veri toplama metodolojisi geliştirilmiştir. Bu yaklaşım, hesaplama veya deneysel kaynaklardan eğitim verilerinin toplanma süresini ve maliyetini en aza indirir. Ardından eğitilmiş YSA, silindir basıncı, silindir sıcaklığı, NO_x ve kurum emisyonları ve silindir ısı transferi gibi motor parametrelerini tahmin etmek için kullanılmıştır. Modelin gelecekteki kontrol geliştirme ve dizel arıtma işlemlerinde önemli olacağı belirtilmiştir. [6]

2015 yılında Ka In Wong, Pak Kin Wong ve Chun Shun Cheung, YSA'nın çoklu yerel minimum, optimal ağ yapısı seçiminde kullanıcı yükü, büyük eğitim veri boyutu ve aşırı uyum riski gibi zorluklarını öne sürerek vardır. Bu zorlukların üstesinden gelmek için, dizel motor performansını modellemek ve tahmin etmek için ortaya çıkan bir makine öğrenme tekniği olan literatürdeki ismiyle Relevance Vector Machine (RVM) uygulamayı önermektedir. Küresel optimal RVM çözümünün özelliği, modelin sadece birkaç deneysel veri seti kullanılarak eğitildiği belirtilmiştir. Bu çalışmada, modelin girdileri motor hızı, yük ve soğutma suyu sıcaklığı iken, çıkış parametreleri frene özgü yakıt tüketimi ve azot oksitler ve karbondioksit gibi egzoz emisyonlarıdır. Deneysel sonuçlar, eğitim verisi az olsa bile model doğruluğunun tatmin edici olduğunu gösterilmiştir. Ayrıca, model doğruluğu, tipik YSA kullanılan model ile kıyaslanabilecek düzeydedir. Bu çalışmanın değerlendirme sonuçları RVM'nin tipik YSA yaklaşımından daha üstün olduğunu iddia etmektedir.[7]

2017 yılında N. Acharya ve ekip arkadaşlarının YSA modeli farklı oranda dizel ve biyodizel karışımı ile beslenen bir dizel motorun farklı parametrelerini tahmin etmek için kullanılmıştır. Veriler, farklı yükleme koşullarında ve farklı dizel ve biyodizel karışım oranlarında bir çift silindirli dizel motorda yapılan bir deneyden elde edilmiştir. İki giriş verisi, yani motor yükü ve karıştırma oranı ve beş çıkış verisi, yani Fren Termal Verimliliği (BTE), Frene Özgü Yakıt Tüketimi (BSFC), Duman seviyesi, Karbon monoksit (CO) ve Azot Oksit (NOx) emisyonları için YSA modellemesi düşünülmüştür. Kullanılan ağ geri yayılımdır, gizli katmanda 10 sayı nöron içeren çok katmanlı algılayıcı ile ileri doğru beslenir. Tahmin edilen ve deneysel olarak ölçülen değerler arasında minimum fark olduğu için modelin tahmin yeteneğinin yüksek olduğu gözlenmiştir. [8]

Elektronik hız regülatörleri dizel jeneratör setlerinde (DGS) yaygın olarak kullanılmaktadır. Elektronik hız regülatörü geliştirmek ve hata ayıklamak için en iyi seçenek, döngü içinde bir donanım (HIL) simülasyon sistemi oluşturmaktır. HIL simülasyon sisteminde, fiziksel dizel motor, maliyeti azaltmak ve daha az emisyon üretmek için matematiksel modeliyle değiştirilir. Gerçek ortama kapanma gereksinimini karşılamak için, matematiksel model temsilcilerinin performansı çok önemlidir.

2018 yılında Mingxin Yu ve arkadaşları, tekrarlayan sinir ağlarına (RNN) dayanan bir dizel motor modelleme yöntemi sunmaktadır. Bu matematiksel model, bir fiziksel DGS'den elde edilen gerçek veriler kullanılarak tanımlanır ve tahmin edilir. Deneysel sonuçlar, önerilen modelin dizel motor çıkış karakteristiklerini elektrik güç yüklerindeki değişikliklerle doğru bir şekilde ürettiğini göstermiştir. Önerilen modeli doğrulamak için, simülasyon deneyi yerleşik HIL simülasyon sistemi üzerinde gerçekleştirilmiştir. Simülasyon deneyinde, rafın yer değiştirmesi ve dönme hızı, HIL simülasyon sisteminin fiziksel kısmından ölçülmüştür. Simülasyon sonucu, önerilen modelin DGS'nin yükleme ve boşaltma işlemlerini iyi bir şekilde simüle edebileceği doğrulanmıştır. [9]

Literatürde çoğunlukla YSA kullanılarak yanma reaksiyonları ile ilgili olan modeller ağır basmaktadır. Turbo-kompresör sistemlerinin YSA kullanılarak modellenmesi ile ilgili çalışma sayısı yok denecek kadar azdır.

1.3 Hipotez

Dizel motorda kullanılan çok kapsayıcı olmayan ve belirli riskler barındıran harita tabanlı modellerden kurtulmak ve bunların yerine daha kapsayıcı, örnek alınmayan/alınamayan bölgelerdeki değerlerin sürekli zamanlı olarak tahminini ortaya koyacak modeller makine öğrenmesi ve/veya derin öğrenme teknikleri ile ortaya konulabilir. Örneğin turbo-kompresör sistemleri gibi çıktıları çok karmaşık/doğrusal olmayan modeller en uygun makine öğrenmesi ve/veya derin öğrenme teknikleri ile elde edilebilir.

Dizel motorların matematiksel olarak ifade edilemeyen harita tabanlı modelleri tamamen test ortamında belirli şartlar altında elde edilen verilere dayanmaktadır. Motor çalışırken farklı fiziksel ortamlar olacaktır ve zamanla motorun fiziksel aksamaları aynı performansı göstermeyecektir. Bu gibi durumlarda harita tabanlı modellerin çok yanlış sonuçlar verebileceği zamanla tahmin edilebilirlik düzeyinin azalacağını görebiliriz. Makine öğrenmesi veya derin öğrenme modelleri ile elde edilen modeller veri haritalarının ulaşamadığı noktaları da tahmin etmesinin yanı sıra çalışma esnasında da verileri toplayarak kendini tekrar eğitebilir. Ve zamanla motorun çalışma koşulları ve fiziksel yapısına göre kendini güncelleyebilir.

2. MODELLENEN SİSTEMLER VE KULLANILAN YÖNTEMLER

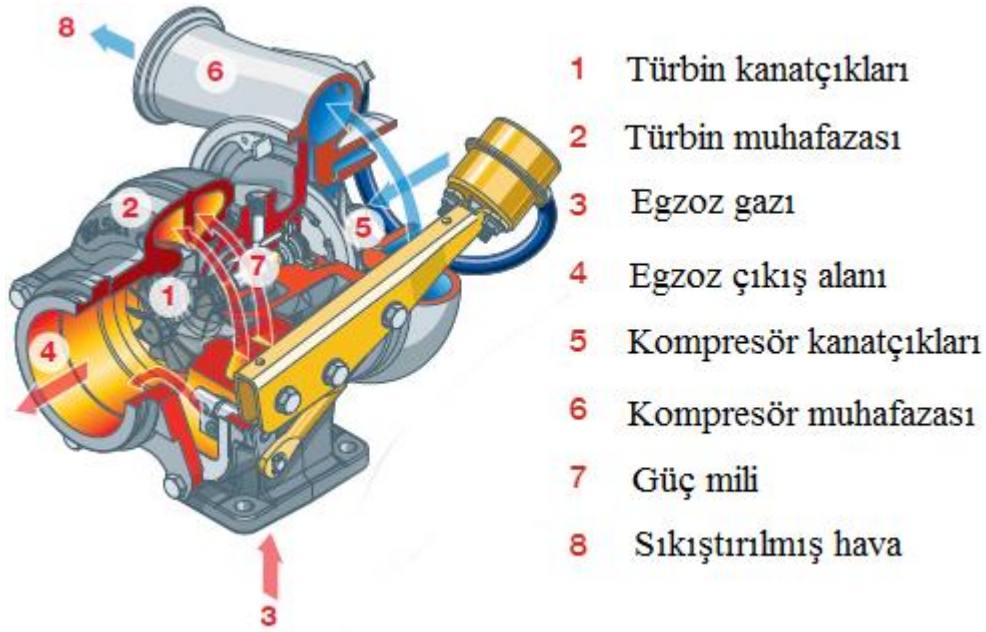
Bu bölümde, tez çalışması kapsamında modellenen dizel motorun alt sistemlerinin parçaları, çalışma prensi ve dizel motorda kullanılma nedeni gibi özellikleri anlatılacaktır. Bunun yanı sıra modellemek için kullanılan makine öğrenmesi ve derin öğrenme teknikleri hakkında temel bilgiler aktarılacaktır.

2.1 Dizel Motorda Turbo - Kompresör (Turboşarj) Sistemi

Bir turbo kompresör, yanma odasına ekstra basınçlı hava zorlayarak bir içten yanmalı motorun verimliliğini ve güç çıkışını artıran türbin tahrikli, zorlanmış bir endüksiyon cihazıdır. Doğal emişli bir motorun güç çıkışı üzerine, kompresörün yanma odasına atmosfer basıncından daha fazla hava ve orantılı olarak daha fazla yakıt almasını sağlayarak katkı sağlar. Ayrıca turboşarjlı bir motor için motor boyutu azaltılarak daha iyi paketleme, ağırlık tasarrufu ve genel olarak daha iyi yakıt ekonomisi elde edilir.

2.1.1 Turbo - kompresör parçaları ve görevleri

Bir turboşarj iki ana bölümden oluşur: türbin ve kompresör. Türbin, türbin çarkından ve türbin mahfazasından oluşur. Egzoz gazını türbin çarkına yönlendirmek türbin muhafazasının görevidir. Egzoz gazından gelen enerji türbin çarkını döndürür ve gaz daha sonra türbin muhafazasının bir egzoz çıkış alanından çıkar. Kompresör ayrıca iki parçadan oluşur: kompresör çarkı ve kompresör gövdesi. Kompresörün çalışma şekli türbinin tersidir. Kompresör çarkı türbine dövme çelik bir şaftla tutturulur ve türbin kompresör çarkını döndürdüğünde, yüksek hızlı eğirme havayı çeker ve sıkıştırır. Kompresör muhafazası daha sonra difüzyon adı verilen bir işlemle yüksek hızlı, düşük basınçlı hava akımını yüksek basınçlı, düşük hızlı hava akımına dönüştürür. Basınçlı hava motora itilir, böylece motor daha fazla güç üretmek için daha fazla yakıt yakar. Şekil 2.1’de bir turboşarj’ın temel parçalarını görebiliriz. Burada temel olarak 8 temel parça üzerinde durulmuştur.



Şekil 2.1 : Turbo - Kompresör parçaları [11].

2.1.2 Turbo - kompresör kullanmanın faydaları

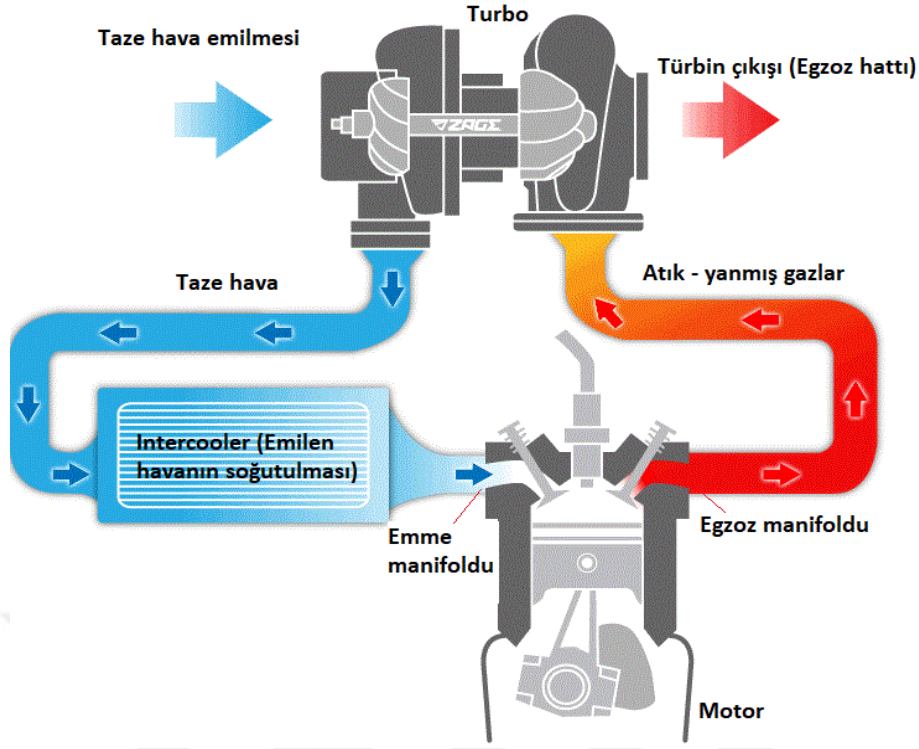
Turboşarjlar bir dizi avantaj sunduğu için modern otomobillerde çok popülerdirler. Son zamanlarda sadece dizel motorlarda değil aynı zamanda benzinli motorlarda bile kullanılmaya başlandığını söyleyebiliriz. Turboşarjlı bir motorun ana artı noktalarına bakacak olursak:

- **Güç:** Turbolar aynı ebattaki motorda daha fazla güç üretir. Bunun nedeni, pistonun her strokunun doğal emişli motorlardan daha fazla güç üretmesidir. Bu, daha fazla otomobilin artık daha büyük ve daha az ekonomik birimlerin yerine daha küçük, turboşarjlı motorlarla donatıldığı anlamına geliyor. Bunun iyi bir örneği Ford'un standart 1.6L benzinli motorunu EcoBoost adını verdiği 1L turboşarjlı bir üniteyle değiştirme kararıdır.
- **Ekonomi:** Turboşarjlar daha büyük, doğal emişli motorlarla aynı güç çıkışını üretebildiğinden, bu daha küçük, daha hafif ve daha ekonomik motorların kullanılmasına zemin hazırlar. Şimdi, tüm modern dizel otomobiller bir turboşarj ile donatılmıştır, yakıt ekonomisini iyileştirir ve emisyonları azaltır. Böylece ekonominin yanı sıra çevreyi koruma açısından da daha fazla katkı sağladığını söylemek yanlış olmayacaktır.

- Tork ve Performans: En küçük motorlarda bile, turboşarjlar daha fazla tork üretir, özellikle devir aralığını düşürür. Bu, otomobillerin şehir etrafında harika olan ve motorun otoyollarda ve A yollarında daha yüksek hızlarda daha rafine hissetmesine yardımcı olan güçlü, performansından faydalandığı anlamına gelir. Düşük hızlarda, küçük turboşarjlı motorlar ürettikleri tork nedeniyle daha büyük, doğal emişli motorlarla donatılmış araçlardan daha uzağa gidebilir.
- Ses: Turboşarjlı bir motordaki hava daha fazla boru ve parçadan filtrelendiğinden, emme ve egzoz gürültüsü azaltılır ve artırılır, böylece daha sessiz ve pürüzsüz bir motor gürültüsü elde edilir. Belki de turboşarjlı bir motorun en beklenmedik faydalarından birisi daha az gürültü çıkarmasıdır.

2.1.3 Turbo - kompresör çalışma prensibi

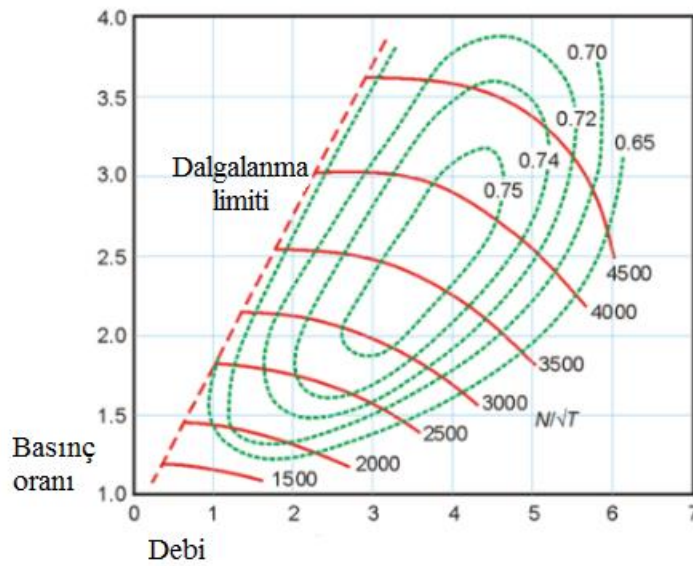
Önceki sayfalarda turbonun parçalarından, getirdiği avantajlardan ve kısaca çalışmasından bahsedilmişti. Şekil 2.2 üzerinden daha detaylı bakarsak silindirlere giren ve çıkan havanın turbo ve kompresör üzerinden nasıl bir yol izlediğini görebiliriz. Turboşarjı iki temel bölüme ayırdığımızdan söz etmiştik bunlar turbo ve kompresör olarak ifade edilir. Öncelikle şemanın sağ tarafına bakacak olursak turbonun görevini rahatlıkla görebiliriz. 4 zamanlı motorlarda egzoz gazı sıcak ve basıncı yüksek bir şekilde hareket ederek turbo kanatçıklarına çarpar böylece mili döndürür. Bu ortak mil hızla dönerek şemanın sol kısımda görülen kompresör kanatçıklarını hareket ettirir ve içeriye daha fazla hava girmesini sağlar. Bilindiği gibi soğuk havanın hacmi daha küçüktür. Dolayısıyla içeriye giren hava soğutulur ve motora öyle gönderilir. Asıl amaç silindirlere daha fazla oksijen göndererek yanmanın verimini artırmaktır. Ayrıca dikkat edilecek olursa turbonun ve kompresörün hızı ortak mil kullanıldığı için her zaman eşit olacaktır.



Şekil 2.2 : Turbo - Kompresör çalışma şeması [12].

2.1.4 Turbo - kompresör çalışma karakteristiği

Kompresör çalışma koşulları her bir kompresör için hazırlanan haritalara göre belirlenmektedir. Kompresör haritası bir motora turbo seçimi esnasında kullanılan temel verilerden birisidir. Motor performans değerlerinin sağlanıp sağlanamayacağı bu haritadan anlaşılabilir. Şekil 2.3’de örnek bir kompresör haritası görülmektedir.



Şekil 2.3 : Turbo - Kompresör çalışma karakteristiği [13].

Yatay eksen kütleli hava debisini (kg/h veya lbs/min), dikey eksen kompresörün sıkıştırma oranını (kompresörün dışındaki hava basıncının kompresörün içindeki hava basıncına oranı), ondalık olarak ifade edilen değerler kompresörün verimini, kırmızı kesik çizgiler kompresörün dalgalanma limitini ve son olarak kırmızı çizgilerin bittiği bölümlerde yer alan rakamlar da kompresörün dönme hızını ifade etmektedir.

Kompresörün çalışma limitlerini belirleyen 3 etmen vardır, bunlar dalgalanma sınırı (Surge Limit), boğulma sınırı (Choke Limit) ve maksimum hız sınırıdır.

- Dalgalanma limiti: Kompresör çarkının istikrarlı bir şekilde dönebilmesi için ihtiyaç duyulan minimum havanın oluşturduğu limitir. Düşük hava debilerinde eğer turbo hızı artarsa kompresör çarkındaki kanatçıklar üzerine hava yeterli düzeyde tutunamaz. Yaşanan bu tutunma kaybı sebebiyle kompresör içerisinde kararsız bir yapı meydana gelir ve basınç dalgalanmaları veya havanın ters yönde akması olayı yaşanır.
- Boğulma limiti: Gaz akışının kanatçıkların arasındaki en dar alandaki hızının ses hızını geçmesi olayıdır. Bu durumda kompresör hızı artmasına rağmen akış hacmi artmamaktadır. Boğulma etkisi, motorda yüksek devirlerde yaşanan bir olaydır. Bu sebepten dolayı yüksek devirlerdeki tork eğrisi arzu edilen seviyede iyileştirilememektedir.
- Maksimum hız limiti: Turboşarjın shaft hızının sınırıdır. Mekanik sınırlar çerçevesinde bu hızın aşılması durumunda turboşarj görevini yerine getiremeyecektir.

2.2 Öğrenme Algoritmaları

Öğrenme algoritmaları yaklaşımlarına, çözmeleri amaçlanan görev veya sorunlara, girdi ve çıktıların aldıkları verilere göre farklılık gösterir.

2.2.1 Denetimli ve yarı denetimli öğrenme

Denetimli öğrenme algoritmaları, eğitim verileri olarak bilinen hem girdi hem de istenen çıktıları içeren bir veri kümesinin matematiksel modelini oluşturur. Yeni girdilerle ilişkili çıktıyı tahmin etmek için kullanılabilecek bir fonksiyon veya model öğrenir.

Denetimli öğrenme algoritmaları, çıktılar sınırlı bir değer grubuyla sınırlandırıldığında sınıflandırma algoritmalarını, çıktılar bir aralık içinde herhangi bir sayısal değere sahip olduğunda regresyon algoritmalarını kullanılır. Yarı denetimli öğrenme algoritmalarında ise eğitim verilerinden bir kısmı kaybedilir. Ancak yine de bu veriler bir modelin kalitesini artırmak için sonradan kullanılabilir.

2.2.2 Denetimsiz öğrenme

Denetimsiz öğrenme algoritmaları yalnızca eğitim verisi olarak giriş değerlerini alır ve bu verilerin gruplanmasını ya da kümelenmesini sağlar. Girdi verileri arasındaki ortak noktaları bulmayı amaçlar ve yeni gelen veri değerinin bu ortak noktalara sahip olup olmadığını göre tepki verir. Denetimsiz öğrenme algoritmaları kümeleme analizini içerir.

2.2.3 Pekiştirmeli öğrenme

Bu öğrenme algoritması kümülatif ödül kavramını en üst düzeye çıkarmak için nasıl bir eylemde bulunmaları gerektiği ile ilgili olan, ödül ve ceza prensiplerine dayanan bir makine öğrenmesi alanıdır. Oyun teorisi, işlem araştırması, kontrol teorisi, bilgi teorisi, simülasyon temelli optimizasyon, küme zekası, istatistik ve genetik algoritmalar gibi birçok disiplinde incelenmiştir.

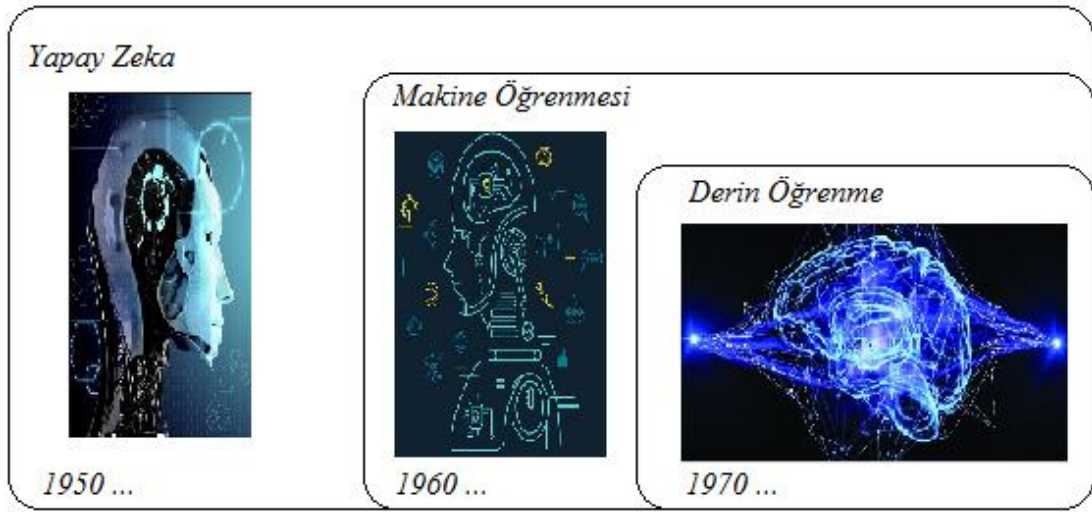
2.2.4 Diğer öğrenme algoritmaları

Kendi kendine öğrenme, özellik öğrenme, seyrek sözlük öğrenmesi gibi son zamanlarda ortaya çıkan öğrenme algoritmaları da yer almaktadır.

2.3 Makine Öğrenmesi

İlk olarak bilgisayar oyunları ve yapay zeka alanında öncü olan Amerikan Arthur Samuel tarafından 1952 yılında IBM firmasında çalışırken ortaya konulan “Makine Öğrenimi” terimi bilgisayar sistemlerinin belirli kurallar kullanmadan istenen görevi yerine getirmek için kullandığı algoritmaların ve istatistiksel modellerin bilimsel bir çalışmasıdır, formüller veya kurallar yerine kalıplara ve çıkarımlara dayanır. Şekil 2.4’den de görüleceği üzere yapay zekanın bir alt dalı, derin öğrenmenin ise bir üst sınıfı olarak görülmektedir. Makine öğrenmesi algoritmaları, görevi gerçekleştirmek için açıkça programlanmadan öngörülerde veya kararlarda bulunmak için "eğitim verisi" olarak bilinen örnek verilere dayanan matematiksel bir model oluşturur.

Makine öğrenimi algoritmaları bir bilgisayarda görevi etkin bir şekilde gerçekleştirmek için geleneksel bir algoritma geliştirmenin zor veya olanaksız olduğu durumlarda birçok problemi çözmek için kullanılabilir.



Şekil 2.4 : Yapay zeka, makine öğrenmesi ve derin öğrenme.

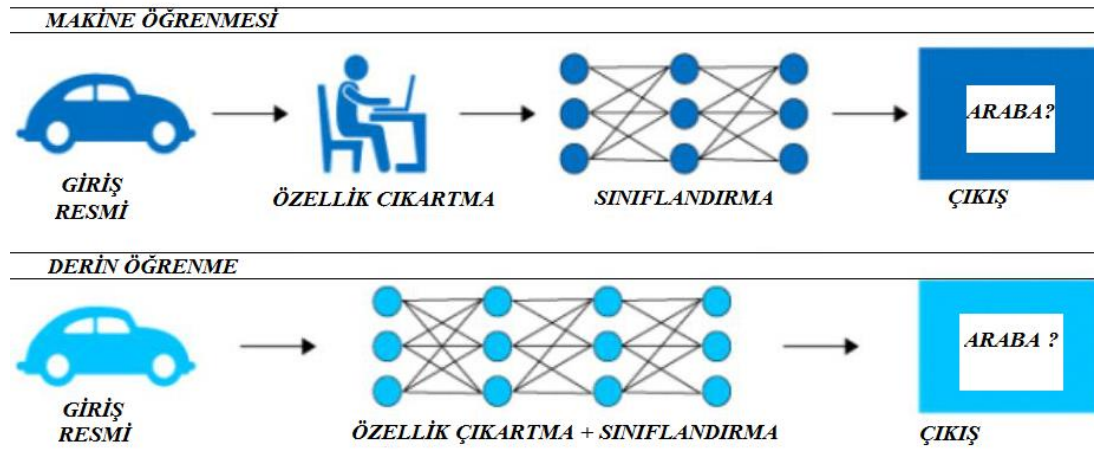
Makine öğrenmesi optimizasyon ve istatistik ile yakından ilişkilidir. Birçok öğrenme problemi, bir örnek eğitim setinde bir takım kayıp fonksiyonlarının en aza indirilmesi olarak formüle edilmiştir. Kayıp fonksiyonları, eğitilen modelin tahminleri ile gerçek problem değerleri arasındaki uyumsuzluğu ifade eder.

Optimizasyon ve makine öğrenmesi arasındaki temel fark optimizasyon algoritmaları bir eğitim setindeki kaybı en aza indirirken, makine öğrenmesi görünmeyen noktalardaki kaybın en aza indirilmesiyle ilgilidir.

Kısacası ezberlemeyi değil öğrenmeyi hedef almaktadır. Yine makine öğrenmesi ile yakından ilişkili olan istatistik kavramı bir örneklem küme üzerinden çıkarım yaparken, makine öğrenmesi genelleştirilebilir tahminde bulunabilecek kalıplar üretmeyi hedefler. Makine öğrenmesinde kullanılan karar ağaçları, destek vektör makineleri, rastgele orman, genetik algoritması , polinom regresyonu, en yakın komşuluk, bayes ağları ve yapay sinir ağları gibi birçok yöntem vardır.

2.4 Derin Öğrenme

Derin öğrenme denetimli, denetimsiz veya yarı denetlemeli algoritmaları kapsayan ve makine öğrenmesinde de kullanılan yapay sinir ağlarının şekil 2.5’den de görüleceği üzere birden fazla gizli katmandan oluşmasıyla ortaya çıkan modellere denilir. Oluşturulan yapay sinir ağı modelleri daha karmaşık olduğu için daha fazla eğitim verisine ihtiyaç duyduklarını ifade edebiliriz. Derin sinir ağları, tekrarlayan sinir ağları veya evrimsel sinir ağları bilgisayarla görme, konuşma tanıma, doğal dil işleme, ses tanıma, sosyal ağ filtreleme, makine çevirisi, ilaç tasarımı, tıbbi görüntü analizi ve malzeme kontrolü gibi birçok alanda kullanılmaktadır.



Şekil 2.5 : Makine öğrenmesi ve derin öğrenme arasındaki fark.

Bu önbilgilerin ardından bu tez çalışmasında uygulanan yöntemlere bakacak olursak, öncelikle bu tez çalışmasında regresyon problemlerini çözmek üzerinedir. Dolayısıyla sınıflandırma algoritmaları bu tez çalışmasının konusu dahilinde değildir. Makine öğrenmesi yöntemlerinden polinom regresyonu, karar ağaçları, destek vektör makineleri ve derin öğrenme yöntemi olarak birden fazla gizli katmandan oluşan yapay sinir ağları kullanılmıştır.

3. KULLANILAN ÖĞRENME YÖNTEMLERİNİN TEORİK ANLATIMI

Bu bölümde, tez çalışması kapsamında kullanılan tüm makine öğrenmesi ve derin öğrenme yöntemlerinin teorik anlatımı yer almaktadır. Bu yöntemler sırasıyla polinom regresyonu, destek vektör makineleri, karar ağaçları ve yapay sinir ağları olarak 4 başlık altında açıklanacaktır.

3.1 Polinom Regresyonu

Bu çalışmada regresyon problemlerinin çözümünde sıklıkla kullanılan ve bir makine öğrenmesi yöntemi olarak da bilinen polinom regresyonun hesaplanmasından bahsedilecektir. Polinom regresyonun nasıl hesaplandığını öğrenmek için ise doğrusal regresyon bilinmelidir.

$$y = m * x + b \quad (m = \text{Eğim}, n = \text{Sabit sayı}) \quad 3.1$$

Doğrusal ve polinom regresyonu yöntemleri gerçek değer ve tahmin edilen değer arasındaki farkların toplamının en küçük olacağı fonksiyonu elde etmeyi amaçlar. Denklem 3.2’de görüldüğü üzere birinci dereceden bir fonksiyonu ele alırsak;

$$r(x) = \sum_{i=0}^n [y_i - (mx_i + b)]^2 \quad 3.2$$

En küçük kareler fonksiyonunun toplamının minimumunu bulmak için kısmi türevler kullanılarak, y_i = gerçek değer , m = eğim , b = sabit sayı , $r(x)$ = en küçük kareler toplamı olacak şekilde denklem 3.3 ve 3.4 ele alınır.

$$\frac{\partial r}{\partial b} = \sum_{i=0}^n 2(b + mx_i - y_i) \quad 3.3$$

$$\frac{\partial r}{\partial m} = \sum_{i=0}^n 2x_i(b + mx_i - y_i) \quad 3.4$$

Elde edilen denklemlere gerekli sadeleştirme işlemleri uygulanarak denklem 3.5 ve 3.6 elde edilir.

$$2b \sum_{i=0}^n 1 + 2m \sum_{i=0}^n x_i - 2 \sum_{i=0}^n y_i = 0 \quad 3.5$$

$$2b \sum_{i=0}^n x_i + 2m \sum_{i=0}^n x_i^2 - 2 \sum_{i=0}^n x_i y_i = 0 \quad 3.6$$

Gerekli sadeleştirme ve düzenleme işlemlerinden sonra m ve b için genelleştirilmiş formülleri denklem 3.7 ve 3.8'e bakarak görebiliriz.

$$m = \frac{n \sum_{i=0}^n y_i x_i - \sum_{i=0}^n y_i \sum_{i=0}^n x_i}{\left(n \sum_{i=0}^n x_i^2 - \left(\sum_{i=0}^n x_i \right)^2 \right)} \quad 3.7$$

$$b = \frac{\sum_{i=0}^n y_i \sum_{i=0}^n x_i^2 - \sum_{i=0}^n x_i \sum_{i=0}^n y_i x_i}{\left(n \sum_{i=0}^n x_i^2 - \left(\sum_{i=0}^n x_i \right)^2 \right)} \quad 3.8$$

Bilgisayar programlarında problemleri çözmek için genellikle matrisler kullanılmaktadır. Dolayısıyla eşitliklerimizi matris olarak ifade etmemiz gerekir. Elde edilen birinci dereceden denklemin matris formu denklem 3.9'da görülmektedir.

$$\begin{bmatrix} n & \sum_{i=0}^n x_i \\ \sum_{i=0}^n x_i & \sum_{i=0}^n x_i^2 \end{bmatrix} \begin{bmatrix} b \\ m \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^n y_i \\ \sum_{i=0}^n x_i y_i \end{bmatrix} \quad 3.9$$

Bu tez çalışmasında ikinci dereceden polinom regresyonu ve altıncı dereceden polinom regresyonu elde edilmiştir. Polinom regresyonunu yüksek dereceden fonksiyonlar için genel olarak ifade etmemiz gerekmektedir. Bu aşamada ikinci dereceden, c sabit katsayılarından oluşan denklem 3.10'daki gibi bir fonksiyonu ele alalım denklem.

$$y = c_0 + c_1 x + c_2 x^2 \quad 3.10$$

Denklem 3.11'i kullanarak en küçük kareler toplamını hesaplayalım.

$$r(x) = \sum_{i=0}^n (c_0 + c_1 x_i + c_2 x_i^2 - y_i)^2 \quad 3.11$$

Doğrusal regresyonda olduğu gibi en küçük kareler toplamını minimum değere indirmek için kısmi türevleri alınmalı ve sıfıra eşitlenmelidir. Tüm c katsayılarına denklem 3.12, 3.3 ve 3.14’de görüldüğü gibi sırasıyla kısmi türevler alınarak denklem 3.15, 3.16 ve 3.17 elde edilir.

$$\frac{\partial r}{\partial c_0} = 2 \sum_{i=0}^n (c_0 + c_1 x_i + c_2 x_i^2 - y_i) \quad 3.12$$

$$\frac{\partial r}{\partial c_1} = 2 \sum_{i=0}^n (c_0 x + c_1 x^2 + c_2 x^3 - x_i y_i) \quad 3.13$$

$$\frac{\partial r}{\partial c_2} = 2 \sum_{i=0}^n (c_0 x^2 + c_1 x^3 + c_2 x^4 - x_i^2 y_i) \quad 3.14$$

$$2 \sum_{i=0}^n (c_0 + c_1 x_i + c_2 x_i^2 - y_i) = 0 \quad 3.15$$

$$2 \sum_{i=0}^n (c_0 x + c_1 x^2 + c_2 x^3 - x_i y_i) = 0 \quad 3.16$$

$$2 \sum_{i=0}^n (c_0 x^2 + c_1 x^3 + c_2 x^4 - x_i^2 y_i) = 0 \quad 3.17$$

Gerekli sadeleştirme ve matematiksel işlemlerden sonra denklem 3.18, 3.19 ve 3.20 eşitlikler elde edilir.

$$c_0 n + c_1 \sum_{i=0}^n x_i + c_2 \sum_{i=0}^n x_i^2 = \sum_{i=0}^n y_i \quad 3.17$$

$$c_0 \sum_{i=0}^n x_i + c_1 \sum_{i=0}^n x_i^2 + c_2 \sum_{i=0}^n x_i^3 = \sum_{i=0}^n x_i y_i \quad 3.18$$

$$c_0 \sum_{i=0}^n x_i^2 + c_1 \sum_{i=0}^n x_i^3 + c_2 \sum_{i=0}^n x_i^4 = \sum_{i=0}^n x_i^2 y_i \quad 3.20$$

Elde ettiğimiz bu denklemi matris formuna dönüştürürsek ikinci dereceden bir fonksiyonun katsayılarını elde edebileceğimiz bir matrisi denklem 3.21’de görebiliriz.

$$\begin{bmatrix} n & \sum_{i=0}^n x_i & \sum_{i=0}^n x_i^2 \\ \sum_{i=0}^n x_i & \sum_{i=0}^n x_i^2 & \sum_{i=0}^n x_i^3 \\ \sum_{i=0}^n x_i^2 & \sum_{i=0}^n x_i^3 & \sum_{i=0}^n x_i^4 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^n y_i \\ \sum_{i=0}^n x_i y_i \\ \sum_{i=0}^n x_i^2 y_i \end{bmatrix} \quad 3.21$$

Denklem 3.22’de kısmi türevler genelleştirilir ve daha kolay görülebilen bir hal alır.

$$\frac{\partial}{\partial c_m} \sum_{i=0}^n (c_0 + c_1 x_i + c_2 x_i^2 \dots c_m x_i^m - y_i)^2 = 2 \sum_{i=0}^n (c_0 x_i^m + c_1 x_i^{(m+1)} + c_2 x_i^{(m+2)} \dots c_m x_i^{(2m)} - x_i^m y_i) \quad 3.22$$

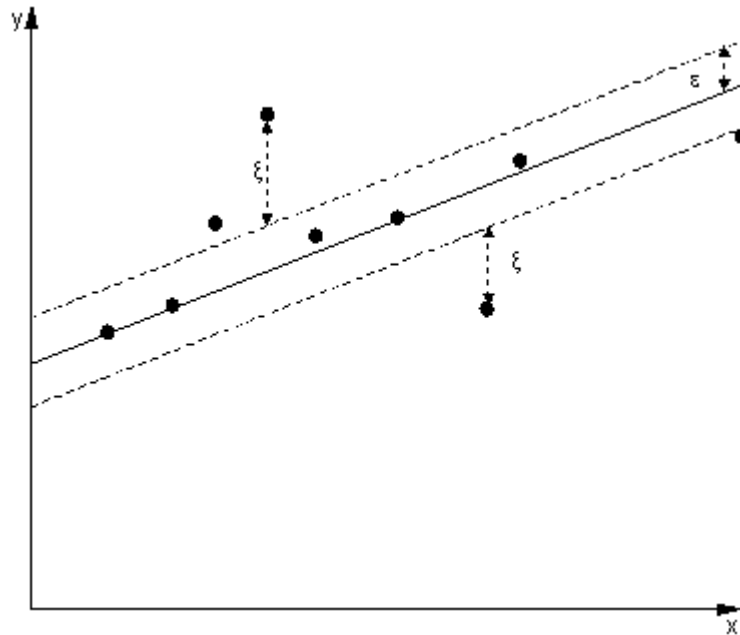
Bu genel ifadeyi hesaplamalar için daha kullanışlı olan matris formuna dönüştürürsek aşağıdaki eşitlik 3.23 ortaya çıkar. Burada m polinom derecesini n ise nokta sayısını ifade etmektedir.

$$\begin{bmatrix} n & \sum_{i=0}^n x_i & \sum_{i=0}^n x_i^2 & \dots & \sum_{i=0}^n x_i^m \\ \sum_{i=0}^n x_i & \sum_{i=0}^n x_i^2 & \sum_{i=0}^n x_i^3 & \dots & \sum_{i=0}^n x_i^{(m+1)} \\ \sum_{i=0}^n x_i^2 & \sum_{i=0}^n x_i^3 & \sum_{i=0}^n x_i^4 & \dots & \sum_{i=0}^n x_i^{(m+2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sum_{i=0}^n x_i^m & \sum_{i=0}^n x_i^{(m+1)} & \sum_{i=0}^n x_i^{(m+2)} & \dots & \sum_{i=0}^n x_i^{2m} \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_m \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^n y_i \\ \sum_{i=0}^n x_i y_i \\ \sum_{i=0}^n x_i^2 y_i \\ \vdots \\ \sum_{i=0}^n x_i^m y_i \end{bmatrix} \quad 3.23$$

3.2 Destek Vektör Makineleri

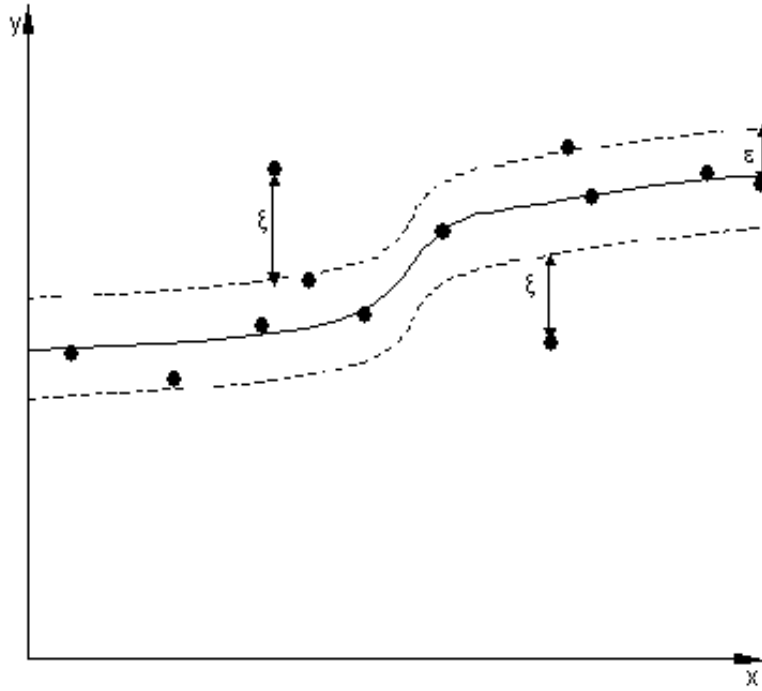
Destek vektör makineleri sınıflandırma ve regresyon problemlerini çözmek için kullanılan en popüler gözetimli makine öğrenmesi algoritmalarından birisidir. Genellikle orta ve küçük boyutlu veri setleri için tercih edilmeleri en önemli avantajıdır. Bu tez çalışması regresyon problemini çözmeye odaklandığı için bu kısımda destek vektör makinelerinin regresyon algoritması açıklanmıştır.

Destek vektör makineleri regresyon problemini çözmek sınıflandırma için sınıflandırma ile hemen hemen aynı mantıkta çalışır ancak sayısal bir sonuç üretmesi gerektiği için birkaç temel fark yer almaktadır. Maksimum marj algoritmasını karakterize eden tüm ana özellikleri içerir. Doğrusal olmayan bir fonksiyon, doğrusal öğrenme makinesi eşlemesi ile yüksek boyutlu çekirdeğin oluşturduğu özellik uzayına yasalır. Sistemin kapasitesi, özellik alanının boyutluluğuna bağlı olmayan parametrelerle kontrol edilir. Sınıflandırma yaklaşımında olduğu gibi, regresyon için verilen genelleme sınırlarını aramak ve optimize etmek için motivasyon vardır. Gerçek değerin belirli bir mesafesine yerleştirilen hataları görmezden gelen kayıp fonksiyonunu tanımlamaya dayanırlar. Bu fonksiyona genellikle epsilon yoğun kayıp fonksiyonu denir. Şekil 3.1 epsilon yoğun bant ile bir boyutlu doğrusal regresyon fonksiyonunun bir örneğini göstermektedir. Değişkenler eğitim noktalarındaki hataların maliyetini ölçer. Bunlar bandın içindeki tüm noktalar için sıfırdır.



Şekil 3.1 : Epsilon yoğun bant ile bir boyutlu doğrusal regresyon.

Şekil 3.2’de ise doğrusal olmayan regresyon durumu için görseli görebiliriz.



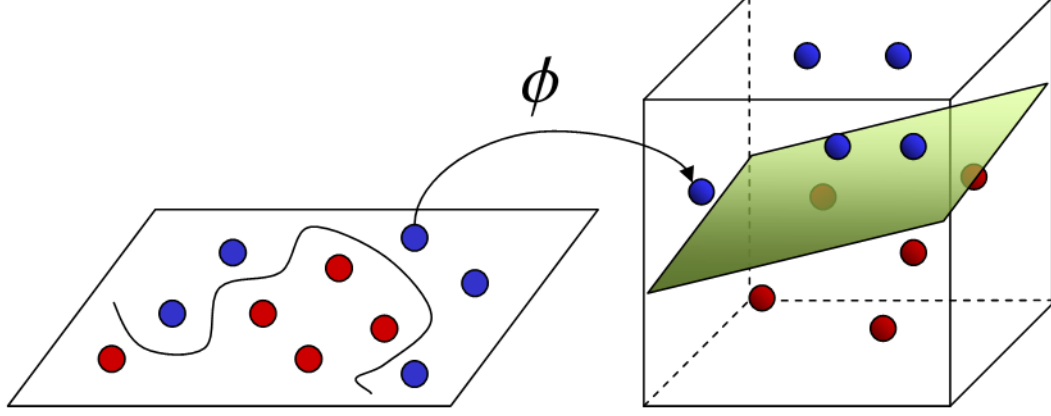
Şekil 3.2 : Doğrusal olmayan regresyon fonksiyonu.

Epsilon yoğun kayıp fonksiyonunu kullanarak global minimumun varlığını ve aynı zamanda güvenilir genelleme sınırının optimizasyonunu sağlıyoruz. Şekil 3.1 ve şekil 3.2’de koyu renkle gösterilen çizgiye hiper düzlem denir, hiper düzlemden epsilon çıkarılması ve epsilon eklenmesi ile elde edilen iki kesikli çizgiye ise sınır çizgileri denilir.

SVM regresyonunda, giriş önce bazı sabit (doğrusal olmayan) eşlemeler kullanılarak m boyutlu bir özellik alanı üzerinde haritalanır ve daha sonra bu özellik alanında doğrusal bir model oluşturulur. Denklem 3.24’de gösterilen matematiksel model kullanılır.

$$(f, w) = \sum_{j=1}^m w_j g_j(x) + b \quad 3.24$$

Denklem 3.24’de yer alan $g_j(x)$ bir doğrusal olmayan dönüşüm setini ifade eder, b değeri ise bias olarak tanımlanır. Genellikle, verilerin sıfır ortalaması olduğu varsayılır (bu, ön işleme tabi tutularak elde edilebilir), bu nedenle bias terimi düşürülür. Şekil 3.3’de yüksek boyutlu özellik uzayına geçişi görebiliriz.



Şekil 3.3 : Girdi örneklerinin doğrusal olmayan haritasının yüksek boyutlu özellik uzayına dönüştürülmesi.

Tahminin kalitesi $L(y, f(x, w))$ kayıp fonksiyonu ile ölçülür. Destek vektör regresyonu, epsilon duyarsız kayıp fonksiyonu adı verilen yeni bir kayıp fonksiyonu kullanır. Eğer $|y - f(x, w)| \leq \varepsilon$ şartını sağlıyorsa kayıp fonksiyonu 0 olacak diğer durumlarda kayıp fonksiyonu $|y - f(x, w)| - \varepsilon$ olacaktır. Denklem 3.25’de gösterilen kayıp fonksiyonlarının ortalamasına deneysel risk adı verilmektedir.

$$R(w) = \frac{1}{n} \sum_{i=1}^n L_g(y_i, f(x_i, w)) \quad 3.25$$

Destek vektör regresyonu epsilon duyarsız kayıp fonksiyonu kullanarak yüksek boyutlu özellik alanında doğrusal regresyon gerçekleştirir. Aynı zamanda $\|w^2\|$ değerini minimize ederek model karmaşıklığını azaltmayı dener. Destek vektör makineleri regresyonu, denklem 3.26 işlevselliğin en aza indirilmesi olarak formüle edilir.

$$\begin{aligned} & 0.5 \|w^2\| + C \sum_{i=1}^n (\xi_i + \xi_i^*) \\ & y_i - f(x_i, w) \leq \varepsilon + \xi_i^* \\ & f(x_i, w) - y_i \leq \varepsilon + \xi_i \\ & \xi_i, \xi_i^* \geq 0, i = 1, \dots, n \end{aligned} \quad 3.26$$

Bu optimizasyon problemi ikili probleme dönüşebilir ve çözümü denklem 3.27’de yer almaktadır. Toplam işaretinin üzerinde yer alan değişken destek vektör sayısını ifade eder.

$$f(x) = \sum_{i=1}^{ngv} (a_i - a_i^*) K(x_i, x) \quad 0 \leq a_i^* \leq C \quad 0 \leq a_i \leq C \quad 3.27$$

Çekirdek fonksiyonu ifadesini denklem 3.28’de görebiliriz.

$$K(x, x_i) = \sum_{j=1}^m g_j(x) g_j(x_i) \quad 3.28$$

Destek vektör makinelerini doğruluğu yukarıdaki formüllerde belirtilen meta parametrelerden C , epsilon ve çekirdek fonksiyonun seçimine bağlıdır. Optimum parametre seçimi problemi, destek vektör makinelerinde model karmaşıklığının (ve dolayısıyla genelleme performansının) üç parametrenin hepsine bağlı olması nedeniyle daha da karmaşılaşmaktadır. C parametresi, model karmaşıklığı (düzlük) ve epsilon’dan daha büyük sapmaların optimizasyon formülasyonunda tolere edilebildiği derece arasındaki farkı belirler. Parametre epsilon, eğitim verilerine uydurmak için kullanılan duyarsız bölgenin genişliğini kontrol eder.

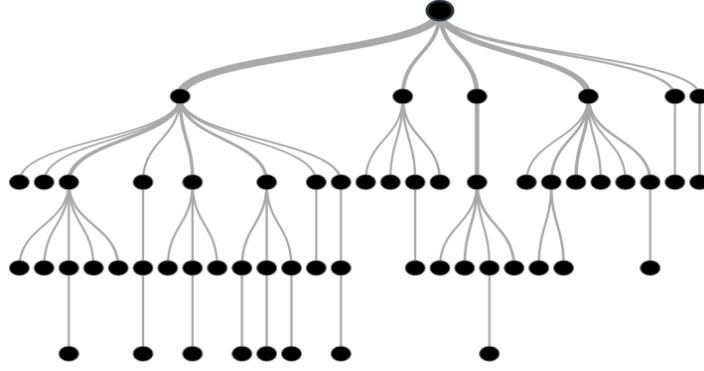
Bu tez çalışmasında kullanılan radyal tabanlı çekirdek fonksiyonu diğer adıyla Gaussian çekirdek fonksiyonu denklem 3.29’da gösterilmektedir.

$$K(x, x') = \exp\left(-\frac{\|x-x'\|^2}{2\sigma^2}\right) \quad 3.29$$

$\|x - x'\|^2$ iki özellik arasındaki Euclidean mesafesi olarak tanımlanabilir. Ayrıca σ serbest parametredir. Bu çekirdek fonksiyonun değeri 0 ile 1 arasında değişmektedir ve uzaklık arttıkça zaman çekirdek fonksiyonun değeri azalmaktadır.

3.3 Regresyon Ağaçları

Karar ağaçları istatistik, veri madenciliği ve makine öğreniminde kullanılan gözetimli öğrenme algoritmalarından biridir. Amaç, birkaç giriş değişkeni ile ilişkili olarak bir hedef değişkenin değerini tahmin eden bir model oluşturmaktır. Sınıflandırma ve regresyon problemlerini çözmek için karar ağaçları sıklıkla kullanılmaktadır. Sınıflandırma ağaçları doğru, yanlış gibi cevaplar üretirken regresyon ağaçları sayısal bir cevap verir. Şekil 3.4’de görüldüğü üzere tepeden aşağıya doğru inen bir strateji izlemektedir. Bir karar ağacı çok fazla değer içeren bir veri kümesine bir takım belirli kurallar uygulayarak verileri parçalara bölmek üzere kurulan bir yapıdır. Kısaca kolay karar verme seçenekleri uygulayarak fazla miktardaki veriyi çok küçük gruplara bölerek kullanılan bir yapı sunar.



Şekil 3.4 : Örnek bir karar ağacı yapısı.

Karar ağaçları kök, dallar ve yaprak olarak nitelenen yapılardan oluşur. Kök başlangıç noktasını temsil ederken yapraklar bize sonuçları verecektir. Bir tahmin yapmak için her adımda bir değişken değerini kontrol etmek gerekir. Karar ağaçlarında bölünmenin neye göre gerçekleşeceği modelin doğruluğunu en fazla etkileyen parametredir. Bu tez çalışmasında hem sınıflandırma ağaçlarında hem de regresyon ağaçlarında sıklıkla kullanılan CART algoritmasından yararlanılmıştır. Bu algorithmada veriler ikili olarak özyinelemeli biçimde bölünür. Dallanma ölçütü olarak bölüm ..'da açıklanan Gini bölmesinden yararlanır, kuruluş sürecinde herhangi bir durma şartı yoktur ve devamlı olarak bölünerek genişler. Sonunda yeni bir bölünmenin elde edilemeyeceği anda bu defa sondan başa doğru budama işlemi başlatılır. Muhtemel en doğru karar ağacı her budama olayı sonrasında rastgele elde edilen bir test verisi ile kontrol yapılarak belirlenmeye çalışılır.

Gini bölmesi: Gini bölmesi veya diğer adıyla Gini endeksi, belirli bir değişkenin rastgele seçildiğinde yanlış sınıflandırılmasının derecesini veya olasılığını ölçer. Denklem 3.30'dan da görüleceği üzere her bir sınıfın kare olasılıklarının toplamını birden çıkartarak hesaplanır.

$$Gini = 1 - \sum_{i=1}^n (p_i)^2 \quad 3.30$$

Gini endeksi derecesi 0 ile 1 arasında değişir. Burada 0, tüm öğelerin belirli bir sınıfa ait olduğunu veya yalnızca bir sınıfın mevcut olduğunu gösterir ve 1, öğelerin çeşitli sınıflara rastgele dağıtıldığını gösterir. 0,5'lik bir Gini endeksi, bazı sınıflara eşit olarak dağılmış elemanları belirtir. Mükemmel sınıflandırılan veriler için Gini endeksi sıfır olacaktır. Bir nesnenin belirli bir sınıfa sınıflandırma olasılığı p_i ile ifade edilir.

3.4 Yapay Sinir Ağları

Bu tez çalışmasında kullanılan diğer ve en etkili yöntem yapay sinir ağlarını kullanarak dizel motorun turbo-kompresör sistemlerinin modellenmesi olmuştur. Bu bölümde yapay sinir ağlarının matematiksel arka planı üzerinde durulacaktır. Öncelikle yapay sinir ağı modelinin nasıl oluşturulduğu hangi parametrelerde değişiklikler yapıldığı, oluşturulan bu yapay sinir ağının eğitilmesi sırasında kullanılan parametrelerin neler olduğu ve matematiksel arka planı anlatılacaktır.

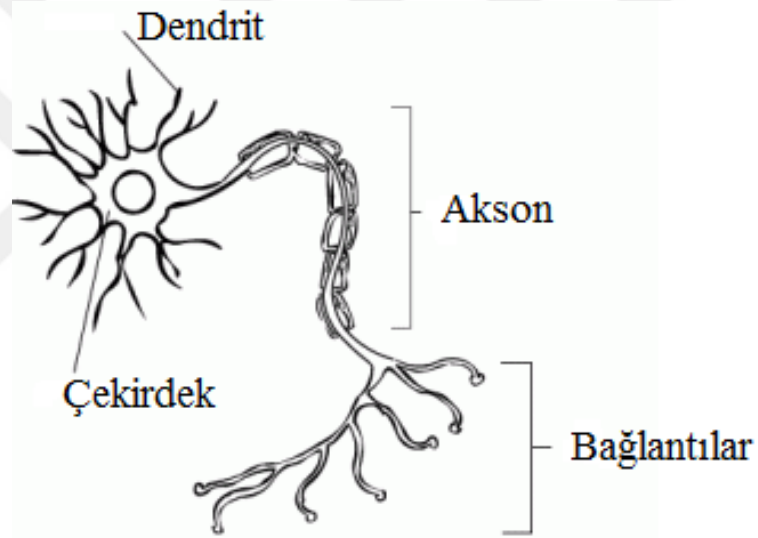
Yapay sinir ağları, insan beyninin özelliklerinden esinlenerek yeni bilgiler oluşturabilme, türetebilme ve keşfedebilme gibi yetenekleri, otomatik olarak ortaya çıkarabilmek için geliştirilen bilgisayar sistemleridir. Yapay sinir ağları; insan beyninden ilham alınarak, öğrenme sürecinin matematiksel olarak modellenmesi hedefiyle ortaya çıkmıştır. Bu yüzden, yapay sinir ağları üzerindeki çalışmalar ilk olarak beyni oluşturan biyolojik hücreler olan nöronların modellenmesi ve bilgisayar üzerinde uygulanması ile başlamış daha sonra bilgisayar sistemlerinin gelişmesiyle birlikte birçok sisteme de uygulanabilir hale gelmiştir. Ancak mevcut bilgisayar teknolojileri çok ilerlemiş olsa da insan beynini bir kenara bırakıp her hangi bir memeli canlının beyni bile ele alınarak yapılan bir değerlendirmede yapay sinir ağlarının çok ilkel kaldığını görebiliriz. Bunun esas nedenlerinden birine bakacak olursak ise insan beyninde yaklaşık 10^{11} sinir hücresinin varlığından bahsedilmektedir bu sayının bilgisayar ortamında modellenmesi ise şu an için pek mümkün değildir. Ancak karar hızı bakımından insan beyni ile yarışamaları bile, yapay sinir ağları günümüzde birçok alanda başarılı sonuçlar vermekte ve etkin olarak kullanılmaktadır. Yapay sinir ağlarının nasıl oluşturulduğundan bahsetmeden önce kısaca insan sinir sisteminin nasıl çalıştığından bahsedilecektir. Çünkü yapay sinir ağları çalışmaları bilindiği gibi insan sinir sisteminden esinlenerek ortaya çıkarılmıştır.

3.4.1 Biyolojik sinir hücresinin yapısı

İnsan beyninde yaklaşık olarak 10 milyar sinir hücresi veya diğer adıyla nöron varlığından söz edilmektedir ve bu nöronların birbirleri ile yaptığı bağlantı sayısının ise yaklaşık olarak 60 trilyon kadar olduğu öne sürülmektedir. Bu sinirler giriş verilerini duyu organlarından almaktadırlar. Daha sonra ise alıcı (taşıyıcı) sinirler bu sinyalleri işleyerek bir sonraki sinir hücresine aktarır böylece sinyalin merkezi sinir sistemine kadar ulaşmasını sağlar.

Merkezi sinir sistemi de bu gelen sinyalleri alarak yorumlar ve ardından tepki sinyalleri üretir. Bu sinyaller de ilgili organlara tepki sinirleri aracılığıyla iletilir. Böylece duyu organlarının algıladığı bilgilere göre ilgili organların tepki vermesi sağlanmış olur.

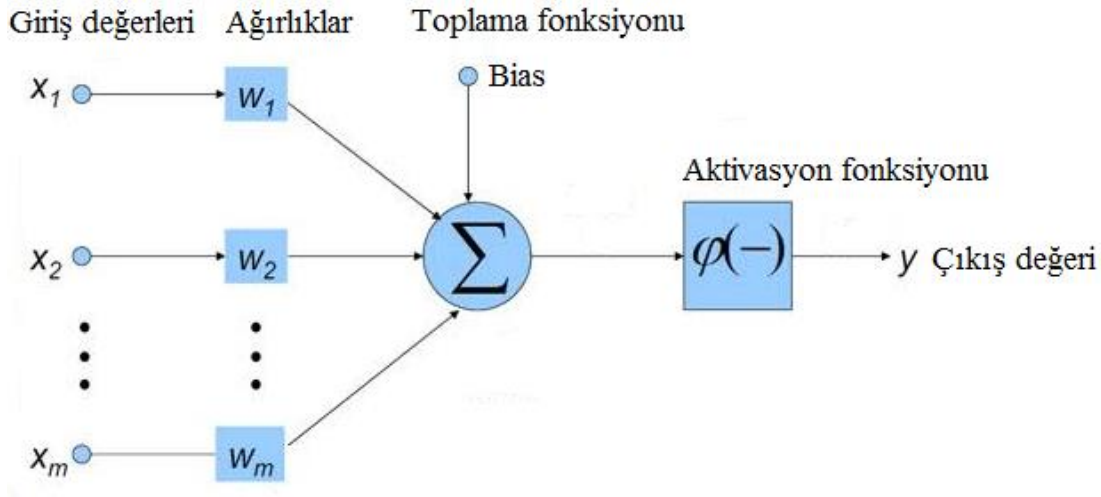
Biyolojik sinir sisteminin yapısına bakacak olursak ilk olarak nöronların yapısı incelenmelidir. Nöronlar şekil 3.5’de görüleceği üzere dendrit, akson, çekirdek ve bağlantılar olmak üzere 4 temel bölümden meydana gelmektedir. Dendritler sinir hücresinin en uç kısımlarında yer alan, bilgilerin ilk toplandığı kısım olarak düşünülebilir. Çekirdek ise dendritlerden gelen sinyalleri bir arada tutar ve bu sinyalleri aksona iletir. Akson aldığı sinyalleri işleyerek işlenen bu verilerin diğer nöronlara iletilmesini sağlayacak olan bağlantılara iletir.



Şekil 3.5 : Biyolojik sinir hücresinin temel yapısı.

3.4.2 Yapay sinir hücresinin yapısı

Yapay sinir hücreleri biyolojik sinir hücrelerinden esinlenerek oluşturulduğu için biyolojik sinir hücreleri ile benzer bölümlere sahiptir. Yapay sinir hücreleri de kendi aralarında bağ kurarak yapay sinir ağlarını oluşturmaktadırlar. Tıpkı biyolojik sinir hücrelerinde olduğu gibi yapay sinir hücrelerinde de sinyallerin alındığı, toplandığı, işlendiği ve iletildiği bölümler vardır. Yapay nöronlar; girdiler, ağırlıklar, toplama fonksiyonu, aktivasyon fonksiyonu ve çıktılar olmak şekil 3.6’dan da görüleceği üzere 5 farklı bölümden oluşmaktadır.



Şekil 3.6 : Yapay sinir hücresinin temel yapısı.

3.4.2.1 Girdiler

Yapay sinir hücresine gelen giriş değerlerine denilir. Giriş değerleri başka bir yapay sinir hücresinden gelen veri olabileceği gibi dış dünyadan alınan bir algılayıcı verisi veya haberleşme verisi olarak da düşünülebilir. Yapay sinir ağının başarılı olabilmesi için en önemli etkenlerden biri de girdilerin doğru seçilmesidir. Eğer çıkışı değiştirmeyen bir değeri ağa girdi olarak tanıtırsak büyük bir hata yapmış oluruz.

3.4.2.2 Ağırlıklar

Giriş verilerinin toplama fonksiyonuna verilmeden önce çarpıldığı katsayılara denilir. Bu katsayılar öğrenme boyunca güncellenir. Pozitif, negatif veya sıfır olabilmektedir. Ağırlık değerleri eğitim sırasında öğrenme oranı gibi birçok parametreye bağlı olarak her iterasyonda güncellenir ve ağın öğrenmesindeki en önemli dinamik değerlerden birisidir. Başlangıç değerleri sabit olarak tasarımcı tarafından belirlenebilir. Eğitim sonunda son değerlerine ulaşırlar fakat bazı uygulamalarda eğitimden sonra bile bu değerlerin güncellenmesi istenebilir.

3.4.2.3 Toplama fonksiyonu (Birleştirme fonksiyonu)

Tüm Girdi verileri ağırlıklarla çarpıldıktan sonra toplama fonksiyonu adı verilen bu fonksiyona girip toplanarak tek bir çıktı olarak aktivasyon fonksiyonuna gönderilir. Bazen girdilerin değeri önemli olurken bazen ise gelen girdilerin sayısı önemli olabilir. Bir problem için ideal bir toplama fonksiyonu yoktur.

Genellikle deneme yanılma yoluyla toplama fonksiyonu tasarımcı tarafından belirlenir. Her hücrenin toplama fonksiyonunun aynı olmadığı tasarımlar bile mevcuttur. En çok kullanılan ise denklem 3.31’de yer alan girdilerin ağırlıklar ile çarpılıp toplandığı toplama yöntemidir. En çok kullanılan toplama fonksiyonları 6 farklı denklem olarak sırasıyla verilmiştir.

Toplam

Ağırlık değerlerinin girdiler ile çarpılıp birbirleri ile toplandığı yöntem denklem x.1’de görülmektedir.

$$v = \sum_{i=1}^n x_i * w_i \quad 3.31$$

Çarpım

Ağırlık değerlerinin girdiler ile çarpılıp daha sonra tekrar birbirleri ile çarpıldığı yöntem denklem 3.32’de görülmektedir.

$$v = \prod_{i=1}^n x_i * w_i \quad 3.32$$

Maksimum

Ağırlık değerlerinin girdiler ile çarpılıp içlerinden en büyüğünün alındığı yöntem denklem 3.33’de görülmektedir.

$$v = \text{Max} (x_i * w_i) \quad 3.33$$

Minimum

Ağırlık değerlerinin girdiler ile çarpılıp içlerinden en küçüğünün alındığı yöntem denklem 3.34’de görülmektedir.

$$v = \text{Min} (x_i * w_i) \quad 3.34$$

Çoğunluk

Ağırlık değerlerinin girdiler ile çarpılıp her birinin işaretine bakılır. Pozitif ve negatif olan sayılardan büyük olanın sayısı denklem 3.35’de görüldüğü gibi alınır.

$$v = \sum_{i=1}^n \text{Sgn} (x_i * w_i) \quad 3.35$$

Kumilatif toplam

İlk yönteme ek olarak bir önceki toplama sonucunun da eklenmesiyle elde edilen yöntem denklem 3.36 da görülmektedir.

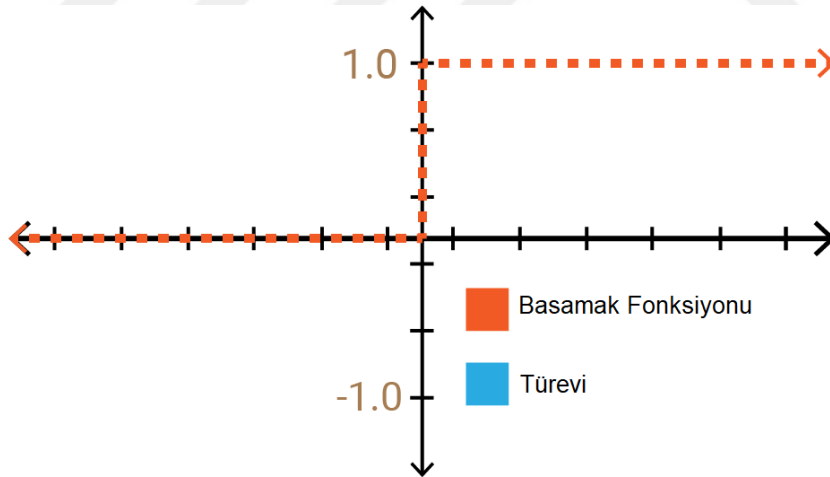
$$v_1 = v_{(0)} + \sum_{i=1}^n x_i * w_i \quad 3.36$$

3.4.2.4 Aktivasyon fonksiyonu

Öğrenme hızını ve sonucunu direkt olarak etkileyen bir başka önemli unsur ise aktivasyon fonksiyonudur. Toplama fonksiyonunun verdiği değeri işleyerek ağıın çıktısını üreten fonksiyondur. Genellikle doğrusal olmayan ve çok yüksek değerler üretmeyen fonksiyonlar seçilir. Yapay sinir ağları aktivasyon fonksiyonlarının doğrusal olmaması sebebiyle doğrusal sistemler değildir. Aktivasyon fonksiyonlarının türevleri kolay hesaplanmalıdır. Ağ eğitilirken hem geriye yayılım algoritmasında hem de kullanılacak optimizasyon yöntemlerinde türev alma işlemi sıklıkla kullanılacağı için aktivasyon fonksiyonları seçilirken türevinin kolay alınıyor olması dikkate alınmalıdır. Bu yüzden ağıın öğrenme hızını da direkt olarak etkiler. Yapay sinir ağlarında birçok aktivasyon fonksiyonu kullanılmaktadır. En sık kullanılanlar ve bu tez çalışmasında kullanılanlar bu bölümde yer almaktadır.

Basamak fonksiyonu

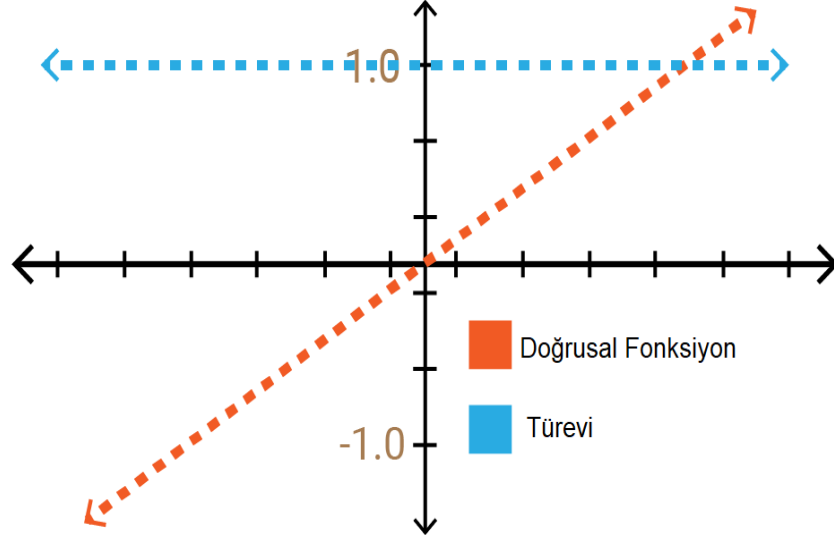
İkili sınıflandırıcı olarak kullanılır. Türevi alınmasının bir şey ifade etmemesinden dolayı ara katmanlarda değil de sadece çıkış katmanında kullanılabilir. Şekil 3.7’de grafiksel olarak görebiliriz.



Şekil 3.7 : Basamak fonksiyonu ve türevi.

Doğrusal fonksiyon

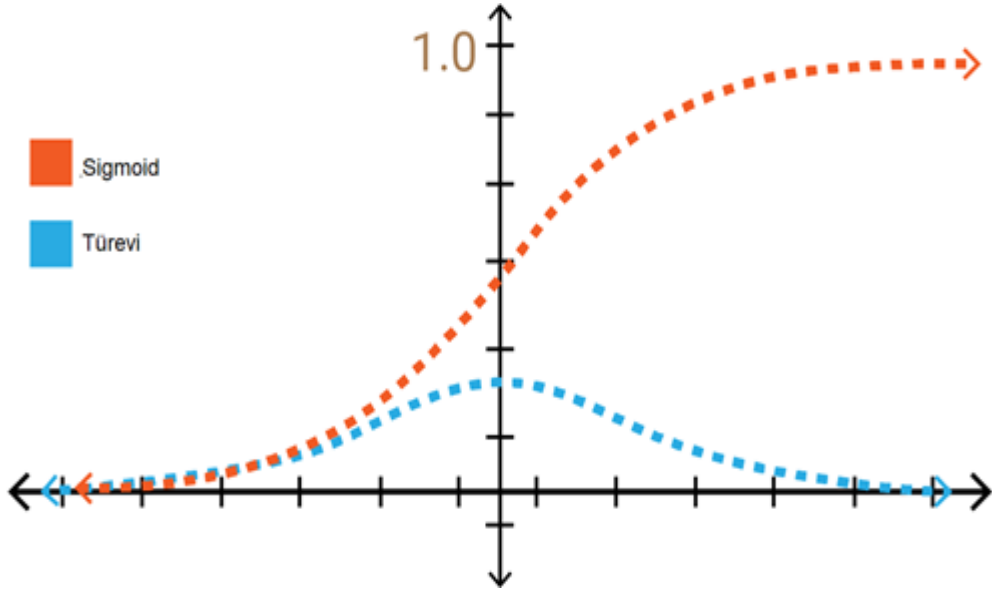
Doğrusal fonksiyonda ara katmanlarda kullanılamaz. Türevi alınır ancak sürekli sabit bir değer üretecektir dolayısıyla geriye yayılım algoritmasında giriş değerinin hiçbir etkisi kalmayacaktır. Bu fonksiyon da basamak fonksiyonu gibi sadece çıkışta kullanılabilir. Şekil 3.8’de fonksiyonu ve türevini görebiliriz.



Şekil 3.8 : Doğrusal fonksiyon ve türevi.

Sigmoid fonksiyonu

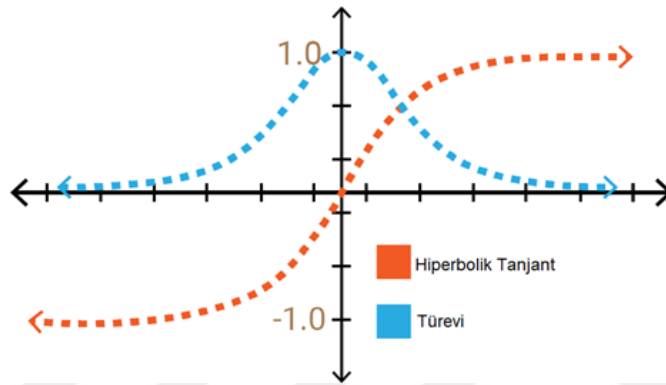
Sigmoid aktivasyon fonksiyonunu ise ara katmanlarda rahatlıkla kullanabiliriz. Bu fonksiyonun bağımsız değişkeninde yapılacak küçük değişimler bağımlı değişkende büyük değişimler yapacaktır. Ayrıca 0 ile 1 arasında değerler üretmesi de aktivasyon değerini belirli bir aralıkta tutmayı sağlayacaktır. Sigmoid aktivasyon fonksiyonu çok sık kullanılmasına rağmen şekil 3.9’da görüleceği üzere uç noktalara gidildikçe türev değerinin 0’a yakınsadığını görürüz. Yani öğrenme sırasında gradyan kaybolması denilen olayı yaşayabiliriz. Öğrenme olayı yerel minimum noktalar da kalabilir.



Şekil 3.9 : Sigmoid fonksiyonu ve türevi.

Hiperbolik tanjant fonksiyonu

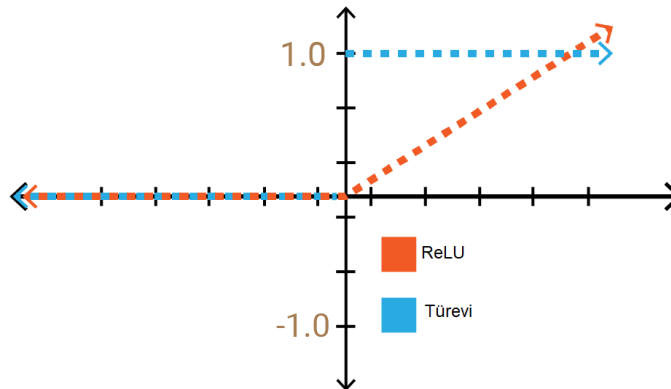
Hiperbolik tanjant fonksiyonu sigmoid aktivasyon fonksiyonuna çok benzemektedir. Ancak fonksiyonun aralığı -1 ile +1 değerleri arasındadır. Sigmoid fonksiyonuna göre ise türevinin daha dik olması sebebiyle daha çok değer alabilme avantajına sahiptir. Bu daha hızlı öğrenme yapacağı ve sınıflama işlemi için daha büyük bir aralığa sahip olacağından dolayı da daha verimli sonuç üretebileceği anlamına gelir. Ama aynı sigmoid aktivasyon fonksiyonun da olduğu gibi uçlarında gradyanların ölmesi problemi vardır. Hiperbolik tanjant aktivasyon fonksiyonunu ve türevinin yer aldığı grafiği şekil 3.10'da görebiliriz.



Şekil 3.10 : Hiperbolik tanjant fonksiyonu ve türevi.

ReLU fonksiyonu

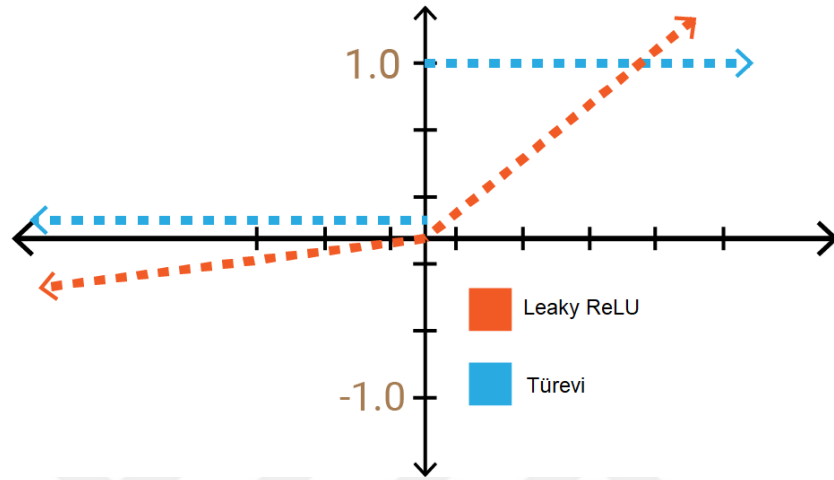
Negatif değerlerin hepsini 0'a çeken pozitif değerlerde ise doğrusal olarak görünen bir yapıya sahiptir. 0 ile $+\infty$ arasında değer alabilmektedir. ReLU fonksiyonu öğrenme hızını artırmasına rağmen . Negatif ekseninde fonksiyonun türevlerinin de 0 gelmesi sebebiyle öğrenmeyi kötü etkileyecektir. Bazı nöronlarda işlem yükünü azaltmak için de tercih edilebilir. Şekil 3.11'de bu fonksiyonu ve türevinin grafiğini görebiliriz.



Şekil 3.11 : ReLU fonksiyonu ve türevi.

Sızıntı ReLU fonksiyonu

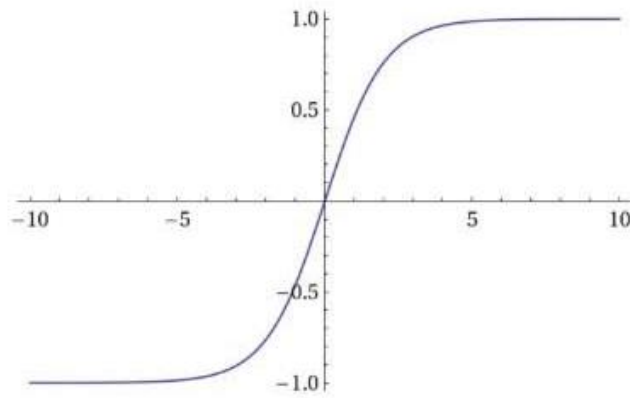
ReLU’da negatif ekseninde ölen gradyan değerlerini ağı kazandırmak için geliştirilmiş olan fonksiyona sızıntı ReLU fonksiyonu denilmektedir. Bu sızıntı değeri 0.01 olarak verilmektedir. Eğer sıfıra yakın farklı bir değer seçilir ise adı rastgele sızıntı ReLU olarak isimlendirilir. Şekil 3.12’de sızıntı ReLU fonksiyonu ve türevi yer almaktadır.



Şekil 3.12 : Sızıntı ReLU fonksiyonu ve türevi.

Softmax fonksiyonu

Softmax genellikle sinir ağlarında, bir ağın normalize edilmemiş çıktısını tahmin edilen çıktı sınıfları üzerindeki olasılık dağılımına eşlemek için kullanılır. Softmax kullanmanın ana avantajı çıktı olasılıkları aralığıdır. Aralık 0 ile 1 arasında olacaktır ve tüm olasılıkların toplamı bire eşit olacaktır. Çoklu sınıflandırma modeli için kullanılan softmax fonksiyonu, her sınıfın olasılıklarını döndürür ve hedef sınıfın olasılığı yüksektir. Softmax fonksiyonunun grafiği şekil 3.13’de yer almaktadır.



Şekil 3.13 : Softmax fonksiyonu ve türevi.

Tüm bu aktivasyon fonksiyonları dışında daha gelişmiş olanları da mevcuttur. Swish fonksiyonu sızıntı ReLU aktivasyonuna benzemekle birlikte negatif bölgede doğrusal olmayan bir yapıya sahiptir. Teorik bilgilerin dışında en uygun aktivasyon fonksiyonu deneme yanılma ile ağıın çıktısına göre belirlenmelidir.

3.4.3 Yapay Sinir Ağının Eğitilmesi

Yapay sinir ağının oluşturulmasından sonra şimdi de yapay sinir ağının eğitilmesi üzerinde durulacaktır. Yapay sinir ağlarının eğitilmesini ileriye yayılım, geriye yayılım ve tüm değerlerin güncellenmesi şeklinde 3 bölüme ayırabiliriz. Yapay sinir ağlarının eğitilmesini anlatmadan önce kayıp fonksiyonundan bahsedelim.

3.4.3.1 Kayıp fonksiyonu

Kayıp fonksiyonu, belirli bir görev için sinir ağımızın ne kadar iyi olduğunu bize söyleyen bir işlevdir. Bunu yapmanın sezgisel yolu, her eğitim örneğini alıp, ağdan geçirmek ve almak istediğimiz gerçek sayıdan çıkararak karesini almaktır. Tüm bu değerleri toplayıp örnek sayısına bölersek de denklem 3.37’de görüldüğü üzere hataların karelerinin ortalamasını elde ederiz.

$$L(y, \hat{y}) = \frac{1}{m} \sum_{i=1}^m (y_i - \hat{y}_i)^2 \quad 3.37$$

3.4.3.2 Gradyan iniş

Sinir ağımızı eğitmeye ilk olarak rastgele ağırlık değerleri kullanarak başlarız ve bu değerler başlangıçta pekiyi sonuç vermeyecektir. Eğitim sürecinde, kötü performans gösteren bir sinir ağı ile başlamak ve yüksek doğrulukta ağı ile sonuçlandırmak isteriz. Kayıp fonksiyonu açısından, eğitim sonunda kayıp fonksiyonumuzun çok daha düşük olmasını istiyoruz. Ağı geliştirmek mümkündür, çünkü ağırlıkları ayarlayarak işlevini değiştirebiliriz. Her adımda ilk işlevden daha iyi performans gösteren başka bir işlev bulmak istiyoruz. Eğitim sorunu, kayıp fonksiyonunun en aza indirilmesi sorununa eşdeğerdir. İşlevleri optimize eden birçok algoritma vardır. Bu algoritmalar, yalnızca işlev tarafından sağlanan bilgileri değil, aynı zamanda onun gradyanını da kullandıkları anlamıyla gradyan tabanlı olabilir veya olmayabilir. En basit gradyan tabanlı algoritmalardan birine stokastik iniş denir. Gradyan uzayda bir yöne işaret eden vektördür.

Aslında fonksiyonun en dik artış yönüne işaret eder. İşlevimizi en aza indirmek istediğimiz için, eğimin ters yönünde bir adım atacağız. Sinir ağında girişleri ve çıkışları sabit sayılar olarak düşünüyoruz. Türevlerimizi alacağımız değişken, ağırlıklardır. Çünkü bunlar ağırmızı geliştirmek için değiştirmek istediğimiz değerlerdir. Eğer kayıp fonksiyonunun gradyanını ağırlıklarımızla hesaplar ve gradyanın ters yönünde küçük adımlar atarsak, kaybımız bazı yerel minimalara yaklaşıp kadar kademeli olarak azalacaktır. Bu algoritmalara “Gradyan İnişi” denir. Gradyan İnişin her yinelemesinde ağırlıkların güncellenmesi kuralı denklem 3.38’de yer almaktadır. Aşağıdaki formülde L kayıp fonksiyonunu, lr ise öğrenme oranını temsil etmektedir.

$$w_j = w_{j-1} - lr * \frac{\partial L}{\partial w_{j-1}} \quad 3.38$$

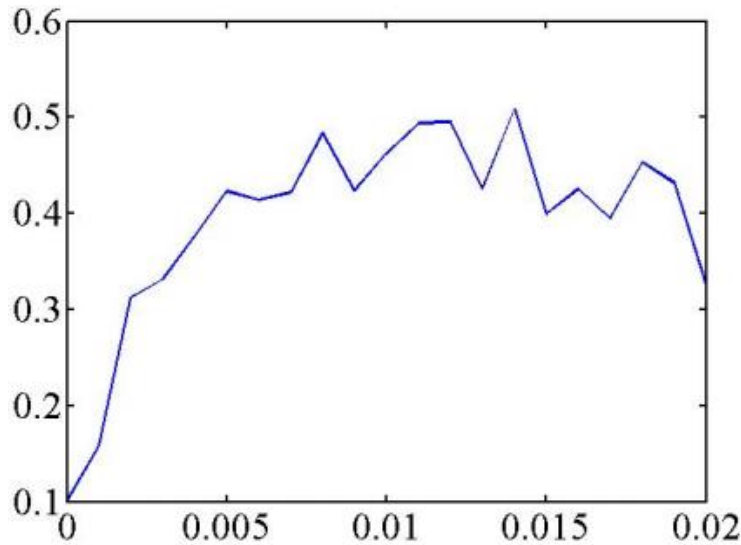
Ne yazık ki, bu algoritmayı sinir ağlarını eğitmek için gerçekten uygulayamayız. Bunun nedeni denklem 3.37’de yer alan kayıp fonksiyonunun formülünde yatmaktadır. Matematikten biliyoruz ki, toplamın türevi, türevlerin toplamıdır. Bu nedenle, kaybın türevlerini hesaplamak için, veri setimizin her bir örneğini gözden geçirmemiz gerekir. gradyan inişin her yinelemesini yapmak çok verimsiz olacaktır, çünkü algoritmanın her bir yinelemesi sadece küçük bir adımla kaybımızı iyileştirir. Bu sorunu çözmek için Mini-Toplu Gradyan inişi adlı başka bir algoritma var. Ağırlıkları güncelleme kuralı aynı kalır, ancak tam türevi hesaplamayacağız. Bunun yerine, veri kümesinin bazı küçük mini partilerindeki türevi yaklaşık olarak hesaplayacağız ve bu türevi ağırlıkları güncellemek için kullanacağız. Mini partinin en uygun yönde adımlar atması garanti edilmez. Aslında, genellikle olmaz. Gradyan inişi ile yeterince küçük öğrenme oranı seçersek, kaybımızın her yinelemeyi azaltması garanti edilir. Mini parti ile bu doğru değil. Kayıp zamanla azalacak, ancak dalgalanacak ve çok daha “gürültülü” olacaktır.

Türevleri tahmin etmek için kullanılacak partinin boyutu, seçilmesi gereken başka bir hiper parametredir. Genellikle, belleğin işleyebileceği kadar büyük bir toplu boyut seçilmek istenir. Ancak genel olarak bu sayı 100’den küçük seçilmektedir. Sonuç olarak toplu gradyan boyutu 1’e eşit olan mini toplu gradyan inişin aşırı sürümüne Stokastik Gradyan İniş denir. Literatürde birçok yeni optimizasyon yöntemi geliştirilmiş ve gelişmekte olmakla birlikte bunlardan çoğu gradyan inişi yönteminden faydalanmaktadır.

Gradyan tabanlı algoritmalar hakkında söylenecek tek şey, gradyanı nasıl hesapladığımızdır. Hesaplamanın en hızlı yöntemi, her bir sinir ağı mimarisi için analitik olarak türev bulmak olacaktır. Ancak, sinir ağı söz konusu olduğunda bu çok fazla vakit alan bir işlem olacaktır. Çok basit bir sinir ağı için yukarıda tanımladığımız formül, tüm türevleri bulmaya çalışırken sıkışacak kadar zordu ve modern mimarilerde milyonlarca kez kullanılması gerekecektir. Bu problemlerin çözümü için geliştirilen geriye yayılım algoritmasından bahsedilecektir.

3.4.3.3 Öğrenme oranı

Öğrenme oranı her bir yinelemeyi ne kadar büyük bir adımda attığımızı kontrol etmek için var olan parametredir. Sinir ağlarını eğitirken ayarlanması gereken en önemli hiper parametrelerden birisidir. Çok büyük seçilirse, adım sayısı azalır. Ancak lokal minimum noktasını kaçırma olasılığı artar. Çok küçük öğrenme hızı seçilirse, bazı yerel minimalara yaklaşmak çok fazla zaman alabilir. İnsanlar, optimal bir öğrenme oranı bulmak için bazı çok iyi teknikler geliştirmişlerdir. Leslie N. Smith “Cyclical Learning Rates for Training Neural Networks.” [2] bölüm 3.3 makalesinde iyi bir öğrenme oranını tahmin etmek için kesin bir yol önermektedir. Bunu yapmak için çok düşük bir öğrenme oranı ile eğitim yapmalı ve her yinelemede doğrusal (veya katlanarak) artırmalıyız. Kayıp fonksiyonu büyük ölçüde artmaya başladığında eğitim durdurulmalıdır. Her yinelemede öğrenme hızını ve kaybını (veya doğruluğunu) kaydetmeliyiz. Daha sonra, öğrenme oranını kayba (veya doğruluğa) karşı çizmeliyiz. Muhtemelen şekil 3.14’e benzeyen grafik ortaya çıkacaktır.

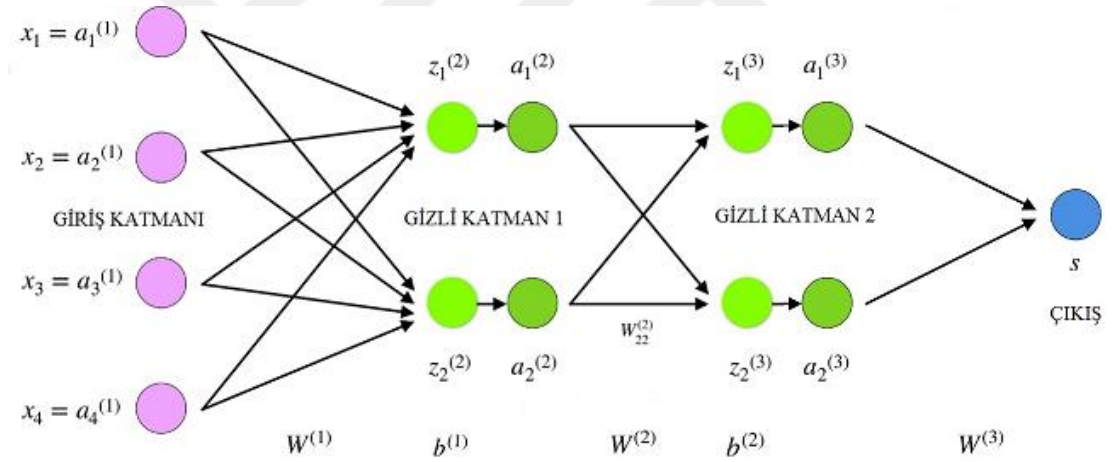


Şekil 3.14 : Öğrenme oranına karşı doğruluk grafiği.

Bu örnekte doğruluğa karşı çizim yaparken, doğruluk azaldığında, düzensizleştğinde veya bozulmaya başladığında öğrenme hızına dikkat edersek. İstedığınız değer budur. Doğru değer 0.005-0.006 civarında olduğu görülür. Bu şekilde öğrenme oranı belirlemek iyi bir çözüm olarak değerlendirilebilir. Ancak uzun ve manuel bir yöntem olduğu görülmektedir. Bunun dışında alternatif olarak birçok öğrenme oranı seçme yöntemi geliştirilmiştir. Üçgenleme yöntemi yeniden başlamalı stokastik gradyan inişi kosinüs yöntemi bunlardan bazılarıdır.

3.4.3.4 İleriye yayılım

Şekil 3.15’de görülen yapay sinir ağını ele alarak yapay sinir ağına ileriye yayılım aşamalarından bahsedilecektir. Mor renkli x değerleri ağına giriş değerlerini temsil etmektedirler. Bunlar skaler değerler olabileceği gibi vektörler veya matrisler de olabilir. Yeşil renkli daireler gizli katmanları mavi renkli daire ise çıkış değerini gösterir.



Şekil 3.15 : Dört giriş tek çıkışlı örnek bir yapay sinir ağı.

Giriş değerleri denklem 3.39’da görüldüğü üzere matematiksel olarak ifade edilebilir.

$$x_i = a_i^{(1)}, i \in 1, 2, 3, 4 \quad 3.39$$

Giriş değerleri başlangıçta rastgele belirlenen ağırlıklar ile çarpılarak gizli katmana verilecektir. Bilindiği gibi w ağırlıkları b ise bias değerlerini temsil etmektedirler. Denklem 3.40 giriş değerlerinin ağırlıklar ile çarpılıp bias değerleri ile toplanmasını ifade etmektedir. Yani toplama fonksiyonlarına karşılık gelen işlemdir.

$$z^{(2)} = w^{(1)}x + b^{(1)} \quad 3.40$$

Daha sonra toplama fonksiyonundan geçen değerler denklem 3.41’de görülen aktivasyon fonksiyonundan geçecektir.

$$a^{(2)} = f(z^{(2)}) \quad 3.41$$

Birinci gizli katmanın aktivasyon fonksiyonunun çıktıları denklem 3.42’de görüldüğü üzere ikinci gizli katmanın toplama fonksiyonuna gönderilir.

$$z^{(3)} = w^{(2)}x + b^{(2)} \quad 3.42$$

İlk katmanın ağırlıklarını matris olarak ifade edecek olursak denklem 3.43’de görüldüğü gibi 2 satır 4 sütundan oluşan bir matris elde ederiz.

$$\begin{bmatrix} w_{11}^{(1)} & w_{12}^{(1)} & w_{13}^{(1)} & w_{14}^{(1)} \\ w_{21}^{(1)} & w_{22}^{(1)} & w_{23}^{(1)} & w_{24}^{(1)} \end{bmatrix} \quad 3.43$$

Giriş değerleri ağırlık değerleri ve bias değerlerini toplama fonksiyonundan geçecek şekilde matris olarak ifade edersek denklem 3.44 karşımıza çıkacaktır.

$$\begin{bmatrix} z_1^{(2)} \\ z_2^{(2)} \end{bmatrix} = \begin{bmatrix} w_{11}^{(1)} x_1 & w_{12}^{(1)} x_2 & w_{13}^{(1)} x_3 & w_{14}^{(1)} x_4 \\ w_{21}^{(1)} x_1 & w_{22}^{(1)} x_2 & w_{23}^{(1)} x_3 & w_{24}^{(1)} x_4 \end{bmatrix} + \begin{bmatrix} b_1^{(1)} \\ b_2^{(1)} \end{bmatrix} \quad 3.44$$

Tüm sistemi matrisel olarak ifade edebiliriz. Ancak bu bölüm için ilk katmanı göstermek yeterli olacaktır. Bir sinir ağının son kısmı, öngörülen değeri üreten çıktı katmanıdır. Basit örneğimizde, mavi renkte tek bir nöron olarak gösterildi ve denklem 3.45’de matematiksel ifadesini görülebilir.

$$s = w^{(3)}a^{(3)} \quad 3.45$$

Bir ileri yayılım işleminde son adım, öngörülen çıktıları beklenen bir çıktıya karşı değerlendirmektir. Ağın tahminini gösteren s ve gerçek sonuç değerini ifade eden y arasındaki değerlendirme bir maliyet fonksiyonu ile gerçekleşir. Bu ortalama kare hatası kadar basit veya çapraz entropi gibi daha karmaşık olabilir. Bu maliyet fonksiyonuna c adını veririz ve denklem 3.46’daki gibi gösterebiliriz. Maliyet fonksiyonunun değerine göre, geri yayılım algoritması kullanılarak model beklenen çıktıya yaklaşmak için parametrelerini ne kadar ayarlayacağını bilir.

$$c = cost (s, y) \quad 3.46$$

3.4.3.5 Geriye yayılım

1989 tarihli makalede yer alan tanıma göre, geri yayılım algoritması ağın gerçek çıkış vektörü ile istenen çıkış vektörü arasındaki farkın bir ölçüsünü en aza indirecek şekilde ağdaki bağlantıların ağırlıklarını tekrar tekrar ayarlar ve yararlı yeni özellikler yaratma yeteneği, geri yayılımı daha önceki daha basit yöntemlerden ayırır. Neden gradyanları hesaplamamız gerektiği sorusunun cevabını ise denklem 3.47 üzerinden açıklayalım.

Maliyet fonksiyonunun gradyanı giriş değerlerinin kısmi türevlerinden oluşan bir vektördür.

$$\frac{\partial c}{\partial x} = \left[\frac{\partial c}{\partial x_1}, \frac{\partial c}{\partial x_2}, \dots, \frac{\partial c}{\partial x_3} \right] \quad 3.47$$

Bir maliyet fonksiyonunun türevi, giriş değerlerinin değişimine göre çıkış değerinin değişim duyarlılığını ölçer. Yani gradyan maliyet fonksiyonunu minimize etmek için x değerinin ne kadar değişmesi gerektiğini gösterir. Bu gradyanları hesaplamak için zincir kuralı adı verilen bir teknik kullanılır. Öncelikle tek bir ağırlık için gradyan değerine denklem 3.48 üzerinden ulaşabiliriz.

$$\frac{\partial c}{\partial w_{jk}^l} = \frac{\partial c}{\partial z_j^l} \frac{\partial z_j^l}{\partial w_{jk}^l} \quad 3.48$$

Toplam fonksiyonunun çıktısını ise denklem 3.49’da görüldüğü gibi genelleştirelim. Burada m nöron sayısını l ise katman sayısını ifade etmektedir.

$$z_j^l = \sum_{k=1}^m w_{jk}^l a_k^{l-1} + b_j^l \quad 3.49$$

Denklem 3.49’un ağırlıklara göre kısmi türevini alırsak karşımıza denklem 3.50 çıkacaktır.

$$\frac{\partial z_j^l}{\partial w_{jk}^l} = a_k^{l-1} \quad 3.50$$

Final değeri olarak ise denklem 3.51’e bakabiliriz.

$$\frac{\partial c}{\partial w_{jk}^l} = \frac{\partial c}{\partial z_j^l} a_k^{l-1} \quad 3.51$$

Aynı işlemleri bias değeri için yapacak olursak karşımıza denklem 3.52 çıkacaktır.

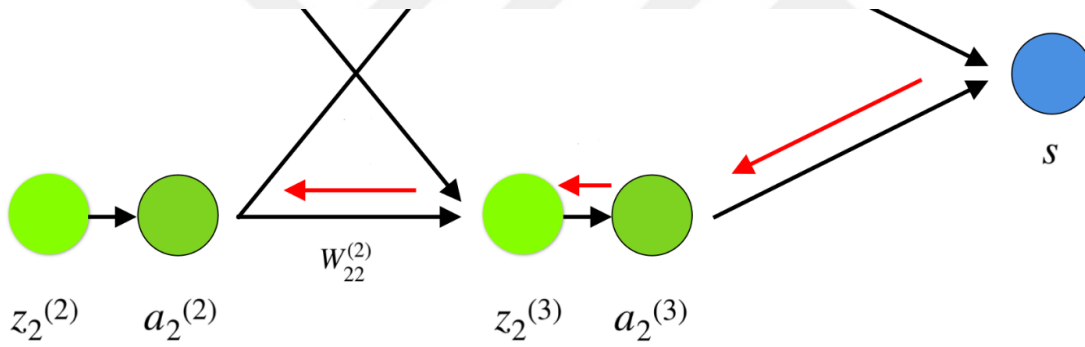
$$\frac{\partial c}{\partial b_j^l} = \frac{\partial c}{\partial z_j^l} \quad 3.52$$

Her iki denklemdeki ortak kısma genellikle yerel gradyan denir ve bu ifadeyi denklem 3.53’de görebiliriz. Yerel gradyan zincir kuralı kullanılarak kolayca belirlenebilir.

$$\delta_j^l = \frac{\partial c}{\partial z_j^l} \quad 3.53$$

Gradyanlar, modelin parametrelerini optimize etmemize izin verir. Genel bir ifade ortaya koyacak olursak bir sonraki ağırlık ve bias değeri bir önceki değerine gradyan değerinin öğrenme oranıyla çarpılması sonucu elde edilen değer eklenerek bulunur diyebiliriz. Denklem 3.54’de görebiliriz. Epsilon öğrenme oranını ifade eder.

Şekil 3.15’de yer olan örnek sinir ağıımızın alt kısmına odaklanarak zincir kuralını uygulayalım. Şekil 3.16’da ilgili kısmı görebiliriz.



Şekil 3.16 : Sinir ağında geri yayılımın görsel temsili.

İkinci gizli katmanın ve çıkış katmanının ağırlık katsayısını, toplama fonksiyonunu ve aktivasyon fonksiyonunu ele alarak uygulanan zincir kuralını denklem 3.55’de görebiliriz.

$$w_1 = w_0 - \epsilon \frac{\partial c}{\partial w_0} \quad 3.54$$

$$\frac{\partial c}{\partial w_{22}^{(2)}} = \frac{\partial c}{\partial z_2^{(3)}} \frac{\partial z_2^{(3)}}{\partial w_{22}^{(2)}} = \frac{\partial c}{\partial a_2^{(3)}} \frac{\partial a_2^{(3)}}{\partial z_2^{(3)}} a_2^{(2)} = \frac{\partial c}{\partial a_2^{(3)}} (z_2^{(3)}) a_2^{(2)} \quad 3.55$$

Bu denklem ikinci gizli katmanın ağırlığının kayıp fonksiyonuna olan etkisini nasıl iyileştireceğimizi hesaplamamızı sağlar. Adım adım geriye giderek önce kayıp fonksiyonunun, aktivasyon fonksiyonuna göre sonra toplama fonksiyonu göre ve son olarak da ağırlığa göre kısmi türevleri alınır ve giriş değeri ile çarpılır. Daha sonra öğrenme oranı ile çarpılır ardından bir önceki değere eklenerek yeni ağırlık değeri hesaplanır burada gradyanın yönü bize pozitif veya negatif bir değer döndürür.

3.4.3.6 Yapay sinir ağlarında kullanılan optimizasyon yöntemleri

Makine öğrenmesi veya derin öğrenme yinelemeli bir süreçtir. Ayarlanacak çok fazla parametre ve denenecek çok fazla yöntem vardır. Yinelemeli döngüyü hızla tamamlamak için modelleri hızlı bir şekilde eğitmek önemlidir. Bu bir makine öğrenimi ekibinin hızını ve verimliliğini artırmanın anahtarıdır. Hızlı eğitim ile birlikte yüksek doğruluk da elde etmek isteriz. Yapay sinir ağlarını eğitirken çok fazla dinamik ve statik parametre işin içine girmektedir ve bu parametreler en iyi şekilde seçilmelidir. Bu parametrelerin bir kısmından önceki sayfalarda bahsedildi. Şimdi ise yapay sinir ağlarının eğitilmesinde en önemli unsurlardan birisi olan optimizasyon yönteminin seçilmesi kısmından bahsedilecektir.

Yapay sinir ağlarında optimizasyon yöntemleri belirlenen hatayı en aza indirmek için ağırlık ve bias değerlerinin güncellenmesinden sorumlu algoritmalarlardır. Aslında bir önceki sayfada en bilinen optimizasyon yöntemi olan geriye yayılım algoritmasından bahsedilmiştir. Yapay sinir ağlarında çoğu işe yarar algoritma geriye yayılım algoritmasında olduğu gibi gradyan tabanlıdır. Geriye yayılım algoritması temel alınarak geliştirilen birçok optimizasyon yöntemi de vardır. En başarılı sonuç veren birkaç optimizasyon yönteminden bahsedelim. Ayrıca bu tez çalışmasında yapay sinir ağlarının en başarılı sonucunu veren optimizasyon yönteminden de ek olarak bahsedilecektir. Optimizasyon yöntemleri tıpkı aktivasyon fonksiyonları gibi deneme yanılma ile netlik kazanır. Burda birçok etken yer almaktadır. Örneğin aynı optimizasyon yöntemi için regresyon veya sınıflandırma problemlerinin çözümü birbirinden tamamen farklı sonuçlar üretebilir.

Genellikle YSA optimizasyonu hakkında konuştuğumuzda, objektif fonksiyon ortalama kare hata fonksiyonudur (kayıp / maliyet fonksiyonu). Objektif fonksiyonu en aza indirmek için sinir ağına değerlerini optimize etmeliyiz.

Geri yayılma gibi gradyan tabanlı arama teknikleri şu anda sinir ağlarını eğitmek için en yaygın kullanılan optimizasyon teknikleri olmasına rağmen, bu gradyan tekniklerinin küresel çözümler bulma yeteneklerinde ciddi şekilde sınırlı olduğu gösterilmiştir. Burda temel mesele yerel minumum noktalara takılmamak, mutlak minumum noktayı bulmaktır. Yapay sinir ağlarının optimizasyonu için en bilinen yöntemlerden biri gradyan inişidir.(Gradient descent).

Tek iterasyonda kullanılan veri setinin uzunluğuna bağlı olarak üç adet gradyan inişi yöntemi (Batch Gradient Descent, Stochastic Gradient Descent, Mini-Batch Gradient Descent) vardır. Gradyan inişi yöntemini esas alan birçok algoritma (Rmsprop, Adagrad, Adam, Nadam) vardır. Bu yöntemleri anlatmadan önce tanımlarda kullanılan epoch kavramını kısaca açıklamakta fayda var. Epoch yapay sinir ağlarında tüm verilerin kaç defa kullanıldığı gösteren parametredir.

Toplu gradyan inişi

Toplu Gradyan İnişinde, tüm eğitim verileri tek bir adım atmak için dikkate alınır. Tüm eğitim örneklerinin gradyanlarının ortalamasını alıp parametrelerimizi güncellemek için bu ortalama gradyanı kullanıyoruz. Yani bu bir epoch'da gradyan inişinin sadece bir adımı. Toplu gradyan iniş dışbükey veya nispeten düzgün hata manifoldları için mükemmeldir. Bu durumda, doğrudan doğruya optimum bir çözüme doğru hareket ediyoruz.

Stokastik gradyan inişi

Toplu Gradyan İnişinde, gradyan İnişinin her adımı için tüm örnekleri değerlendiriyorduk. Ama veri setimiz çok büyükse. Derin öğrenme modelleri veri istemektedir. Veri ne kadar fazla olursa modelin iyi olma şansı o kadar artar. Veri kümemizin 5 milyon örneğe sahip olduğunu varsayalım, o zaman sadece bir adım atmak için modelin tüm 5 milyon örneğin gradyanlarını hesaplaması gerekecektir. Bu durumlarda etkili bir yol değildir. Bu sorunu çözmek için Stokastik Gradyan İniş yöntemi geliştirilmiştir. Stokastik gradyan inişte (SGD), tek bir adım atmak için her seferinde sadece bir örnek ele alıyoruz. SGD için bir epoch'da aşağıdaki adımları gerçekleştiriyoruz:

1. Bir örnek seçilir.
2. Yapay sinir ağı beslenir.

3. Gradyanı hesaplanır.
4. Ağırlıkları güncellemek için 3. adımda hesaplanan gradyan kullanılır.
5. Eğitim veri kümesindeki tüm örnekler için ilk 4 adım tekrarlanır.

Eğitim verileri tek tek kullanılarak güncelleme yapıldığı için dalgalanmalı bir grafik ortaya çıkması beklenir. Ancak zamanla dalgalanmaların şiddetinin azalması beklenir. Bu yöntem dinamik (online) eğitim ve büyük veri kümeleri için tercih edilebilir.

Mini toplu gradyan inişi

Mini toplu gradyan inişi ilk iki yöntemin karışımı gibi düşünülebilir. Ne tüm veri kümesini aynı anda kullanıyoruz ne de tek seferde tek örneği kullanıyoruz. Gerçek veri kümesinden daha az olan sabit sayıda eğitim örneğinin bir kümesini kullanırız ve buna mini toplu olarak adlandırırız. Bunu yapmak, diğer iki yöntemin her ikisinin de avantajlarına ulaşmamıza yardımcı olur. Yani, sabit boyutlu mini partilveri kümelerini oluşturduktan sonra, aşağıdaki adımları bir epoch'da yapıyoruz:

1. Mini veri kümesi seçilir.
2. Sinir ağını beslenir.
3. Mini kümenin ortalama gradyanı hesaplanır.
4. Ağırlıkları güncellemek için bu gradyan kullanılır.
5. Oluşturulan diğer veri kümesi seçilerek 1-4 arasındaki adımlar tekrarlanır.

Bu yöntemde de dalgalanmalar görülür ancak daha hızlı minimum noktaya ilerler iki yöntemde avantaj ve dezavantajlarını içerir. Veri kümeleri 32, 64 gibi sayılar seçilebilir. Şekil 3.17'de bu 3 yöntemin farkını görsel olarak da görebiliriz. Tabi bu her veri kümesi için genelleştirilemez. Bazı problemlerde sadece bu 3 yöntemden birisi başarılı sonuç verebileceği gibi hepsi de hedefe ulaşabilir.



Şekil 3.17 : Toplu, stokastik ve mini gradyan iniş yöntemleri.

Adagrad

Adagrad, her parametre için her iterasyonda farklı öğrenme çarpanı kullanarak, her bir parametre için farklı güncelleme yapmaktadır. Böylece, öğrenme çarpanını manuel olarak ayarlama ihtiyacını ortadan kaldırmaktadır. Adagrad'da her parametrenin kendi öğrenme hızı vardır ve güncelleme işleminde öğrenme çarpanının değerinin bölüldüğü ifadenin eğitim süresince büyümeye devam etmesi sonucunda öğrenme çarpanı aşırı küçülmektedir.

Rmsprop

Rmsprop, Adagrad algoritmasında öğrenme çarpanının çok fazla küçültme problemine bir yanıt yolu olarak geliştirilmiştir. Adagrad'daki önceki bütün eğimlerin karelerinden elde edilen değerlerin hepsini kullanmak yerine, değer miktarını belli bir çerçeve büyüklüğü ile sınırlandırmıştır.

Adam

Bu yöntemde, Rmsprop'ta yapıldığı gibi geçmiş eğimlerin karelerinin üssel olarak ağırlıklandırılmış ortalamalarının kullanılmasının dışında, momentum değişikliklerini de ön hafızada saklar. Kısaca Rmsprop ve momentumu birleştirir. Varsayılan beta değerleri β_1 için 0.9, β_2 için 0.999 olarak belirlenmiştir. Adam'ı dışbükey olmayan optimizasyon problemlerinde kullanmanın cazip faydaları:

- Uygulaması kolaydır.
- Hesaplama açısından verimlidir.
- Az bellek gereksinimi duyar.
- Veri ve / veya parametre açısından büyük sorunlar için çok uygundur.
- Çok gürültülü / veya seyrek gradyanlı problemler için uygundur.
- Hiper parametrelerin sezgisel bir yorumu vardır ve tipik olarak çok az ayarlama gerektirir.

Nadam

Nadam, Adam ve Nesterov accelerated gradienti (NAG) yöntemlerinin birleşiminden oluşmaktadır. NAG algoritmasını Adam algoritmasına eklemek için, momentum ifadesinde değişiklik yapılarak elde edilen sonuçta yer alan güncelleştirme kuralı elde edilir. Şekil 3.18'de formüller görsel olarak ifade edilmiştir.

Sgd	$\theta_{t+1} = \theta_t - \eta \nabla_{\theta} J(\theta_t; x^{(i)}; y^{(i)})$	Adam	$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$
Adagrad	$g_{t,i} = \nabla_{\theta_t} J(\theta_{t,i})$		$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$
	$\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{G_{t,ii} + \epsilon}} g_{t,i}$		$m'_t = \frac{m_t}{1 - \beta_1^t}, v'_t = \frac{v_t}{1 - \beta_2^t}$
Rmsprop	$E[g^2]_t = 0.9E[g^2]_{t-1} + 0.1g_t^2$	Nadam	$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{v'_t + \epsilon}} m'_t$
	$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} g_t$		$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{v'_t + \epsilon}} (\beta_1 m'_t + \frac{(1 - \beta_1) g_t}{1 - \beta_1^t})$
	$g_t = \nabla_{\theta_t} J(\theta_t)$		

Şekil 3.18 : Farklı optimizasyon yöntemleri.

Levenberg-Marquardt algoritması

Matematik ve hesaplamada, Sönümlü en küçük kareler (DLS) yöntemi olarak da bilinen Levenberg-Marquardt algoritması (LMA veya sadece LM), doğrusal olmayan en küçük kareler problemlerini çözmek için kullanılır. Bu minimizasyon problemleri özellikle en küçük kareler eğrisine uydurmada ortaya çıkar. LMA, genel eğri uydurma sorunlarını çözmek için birçok yazılım uygulamasında kullanılır. LMA, Gauss – Newton algoritması (GNA) ile gradyan iniş yöntemi arasında enterpolasyon yapar. LMA, GNA'dan daha sağlamdır, bu da birçok durumda nihai minimumdan çok uzakda başlasa bile bir çözüm bulduğu anlamına gelir. İyi davranan fonksiyonlar ve makul başlangıç parametreleri için LMA, GNA'dan biraz daha yavaş olma eğilimindedir.

Yarı Newton yöntemleri gibi Levenberg-Marquardt algoritması da Hessian matrisini hesaplamak zorunda kalmadan ikinci dereceden eğitim hızına yaklaşmak için tasarlanmıştır. Performans işlevi toplam kareler biçiminde olduğunda (ileri beslemeli ağların eğitiminde tipik olarak olduğu gibi), Hessian matrisi denklem 3.56'daki gibi düşünülebilir ve gradyan denklem 3.57'deki gibi hesaplanabilir.

$$H = J^T J \quad 3.56$$

$$g = J^T e \quad 3.57$$

Burada J, ağırlık ve bias açısından ağ hatalarının ilk türevlerini içeren Jacobian matrisidir ve e, ağ hatalarının bir vektörüdür.

Jacobian matrisi, Hessian matrisini hesaplamaktan daha az karmaşık olan standart bir geri yayılım tekniği ile hesaplanabilir. Levenberg-Marquardt algoritması, bu yaklaşımı Hessian matrisine benzeyen denklem 3.58’de görülen Newton benzeri bir formülle hesaplar.

$$x_{k+1} = x_k - [J^T J + \mu I]^{-1} J^T e \quad 3.58$$

Skaler μ sıfır olduğunda, bu yaklaşık Hessian matrisi kullanılarak sadece Newton’un yöntemidir. Ve μ büyük olduğunda, bu küçük adım boyutuyla gradyanlı iniş haline gelir. Newton’un yöntemi minimum bir hataya yakın daha hızlı ve daha doğrudur, bu yüzden amaç Newton’un yöntemine olabildiğince çabuk geçmektir. Böylece, her başarılı adımdan sonra (performans fonksiyonunda azalma) μ azaltılır ve sadece geçici bir adım performans fonksiyonunu artıracak zaman artar. Bu şekilde, performans işlevi algoritmanın her yinelenmesinde daima azaltılır. Bu algoritma orta büyüklükteki ileri beslemeli sinir ağlarını (birkaç yüz ağırlığa kadar) eğitmek için en hızlı yöntem gibi görünmektedir. MATLAB® yazılımında etkili bir uygulamaya sahiptir, çünkü matris denkleminin çözümü yerleşik bir işlemdir, bu nedenle özellikleri bir MATLAB ortamında daha da belirgin hale gelir.

Matlab programında kullanılan LMA yönteminin bazı kurallarına bakacak olursak: Ağa verilen verileri eğitim, doğrulama ve test olarak üç bölümde değerlendirir. Doğrulama vektörleri, eğer art arda kötü sonuçlar alınırsa eğitimi durdurmak için kullanılır. Eğitim üzerinde herhangi bir etkisinin olmadığı test vektörleri, ağı iyi bir şekilde genelleştirilip genelleştirilmediğini gösterir. Bu fonksiyon Matlab’da “trainlm” olarak geçmektedir ve yapay sinir ağına ağırlıkları, net girişleri ve transfer fonksiyonları türev fonksiyonlarına sahip olduğu sürece bu fonksiyon tarafından eğitilebilir. Jacobian J_x ’in hesaplanması için geriye yayılım algoritmasını kullanır. Aşağıdaki koşullardan birisi gerçekleşirse eğitim sonlandırılır.

- Maksimum epoch sayısına (tekrar) ulaşırsa.
- Maksimum süre aşırsa.
- Performans hedefine ulaşırsa.
- Performans gradyanı minimum gradyanın altına düşerse.
- Doğrulama verisine göre art arda belirlenen sayıya kadar kötü sonuç üretilirse.

Bu tez çalışmasında Matlab’deki tüm yöntemlerden en iyi sonuç veren LMA olmuştur.

4. KULLANILAN EĞİTİM VERİLERİ VE MODELLERİN EĞİTİMİ

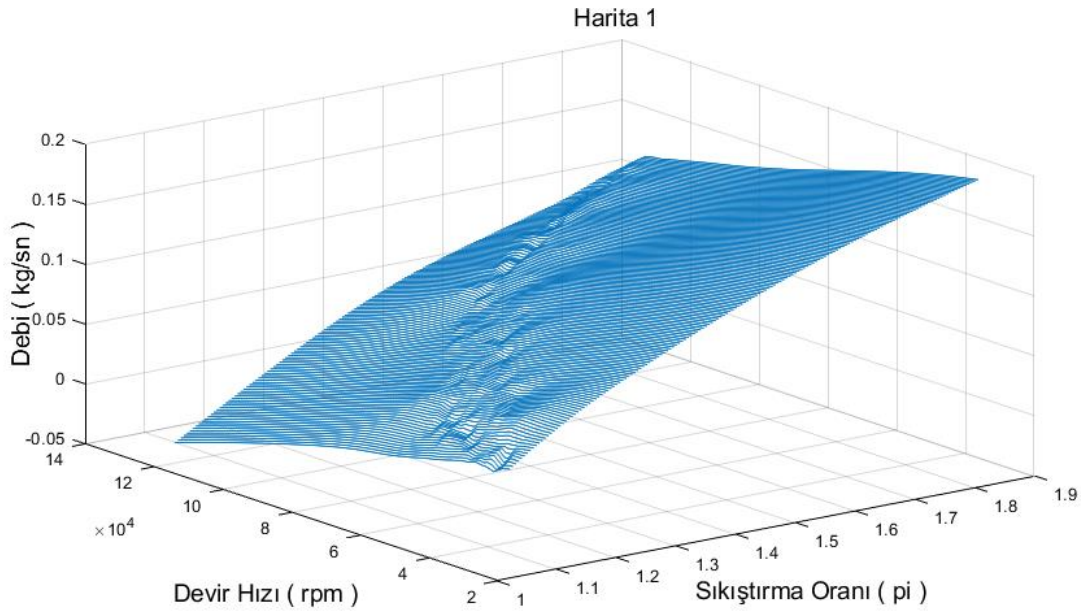
Bu tez çalışmasında turbo kompresör ile ilgili olarak modellenmesi gereken 3 farklı harita üzerinde çalışılmıştır. Bu haritalarda yer alan verileri kullanarak makine öğrenmesi ve derin öğrenme yöntemleriyle uygun modeller oluşturulmuştur. Bu bölümde öncelikle kullandığımız verilerin analizini yaparak inceleme yapılacak ve ardından öğrenme yöntemlerinin modelleri ve eğitim süreçleri açıklanacaktır. Bu tez çalışmasında kullanılan veriler yüksek maliyetli bir laboratuvar ortamında çalışan dizel motordan gerçek zamanlı olarak toplanmasıyla elde edilmiştir.

4.1 Kullanılan Eğitim Verileri

Kullanılan eğitim verileri üç farklı harita olarak verilecektir. Bu üç harita da iki giriş bir çıkışlı eğitim verilerinden oluşmaktadır. İlk veri haritası sıkıştırma oranı ve devir hızına bağlı olarak debiyi gösteren üç boyutlu turbo veri haritasıdır. İkinci veri haritası devir hızı ve debiye bağlı olarak sıkıştırma oranını gösteren üç boyutlu kompresör veri haritası ve üçüncü olarak kullanılacak veri haritası devir hızı ve debiye bağlı olarak kompresörün verimini gösteren haritadır. Bu üç veri haritası birbirinden bağımsız olarak düşünülebilir. Ancak her birinin çıkışı dolaylı olarak bir birini etkileyeceği için tüm sistem düşünüldüğünde bir bağ söz konusu olacaktır. Ancak eğitim modelleri olarak hepsi ayrı düşünülmelidir. Bu üç veri haritasını detaylı olarak şekil 4.1, 4.2 ve 4.3'de göreceğiz. Bu haritaların sayısal analizi ise sırasıyla çizelge 4.1, 4.2 ve 4.3'de verilecektir.

Bu tez çalışmasında makine öğrenmesi ve derin öğrenme tekniklerini uygulamak için ortam olarak Python'da yazılmış açık kaynaklı bir sinir ağı kütüphanesi olan Keras, istatistik ve makine öğrenmesinde sıklıkla kullanılan Scikit-learn kütüphanesi ve Matlab programından yararlanılmıştır. Laboratuvar ortamında elde edilen veriler ilk olarak matris formatında kaydedilmişti. Eğitim verileri olarak kullanılması için Excel ortamında tablo olarak ve Matlab ortamında vektör olarak bu veriler iki giriş bir çıkış temsil edecek şekilde saklandı. Modeller eğitilirken matris formatından tablo ve vektör formatına çevrilen bu veriler üzerinden işlemler yapıldı.

4.1.1 Veri Haritası 1



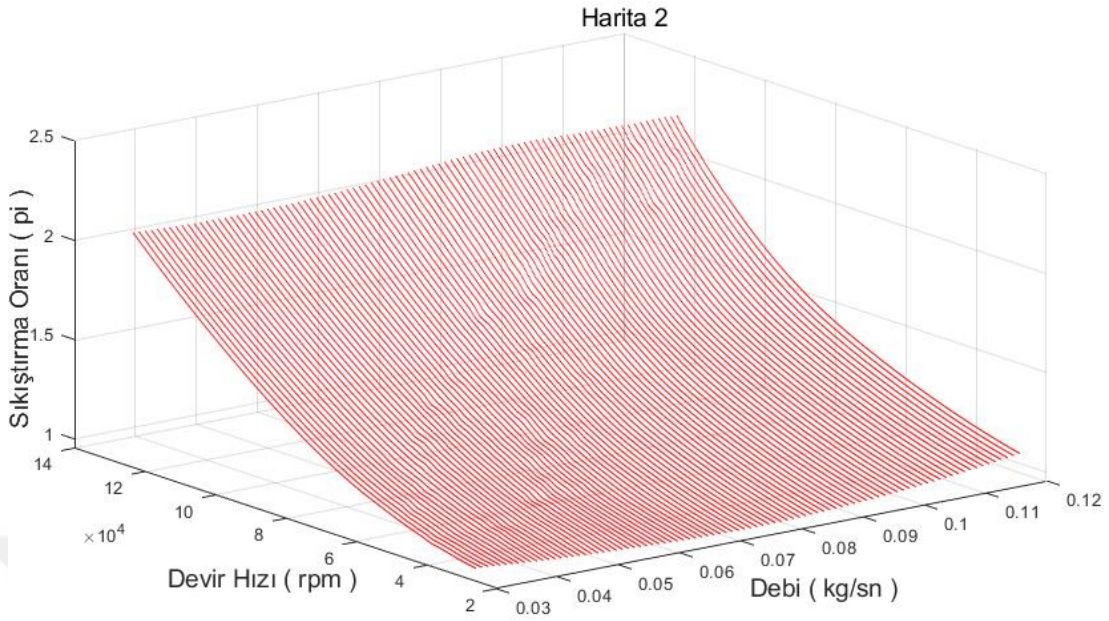
Şekil 4.1 : Sıkıştırma oranı ve devir hızına bağlı olarak debiyi gösteren üç boyutlu turbo veri haritası.

Şekil 4.1’de görüldüğü üzere genel olarak üç boyutlu doğrusal bir veri haritamız var ancak orta kısımlarda oluşan girintili değerleri üretmek burada önemli olan kısım olacaktır. Bu veri haritası için en iyi öğrenme algoritması bu kıvrımlı noktaları da üretmelidir. Harita 1 için detaylı analizi çizelge 4.1’de görebiliriz.

Çizelge 4.1 : Birinci veri haritasının analizi

HARİTA 1	Birinci Giriş	İkinci Giriş	Çıkış
Turbo	Basınç Oranı (pi)	Devir Sayısı (rpm)	Debi (kg/sn)
Veri sayısı	61620	61620	61620
Tekrarsız veri	158	390	61620
En küçük değer	1.055	26000	-0.038337428
En büyük değer	1.840	123250	0.197315119
Artış miktarı	0.0055	250	
Ortalama	1.4475	74625	0.0919262592
Standart sapma	0.228050	28145.9614	0.0501403844
Varyans	0.05200709	792195147.80343	0.0025140581

4.1.2 Veri Haritası 2



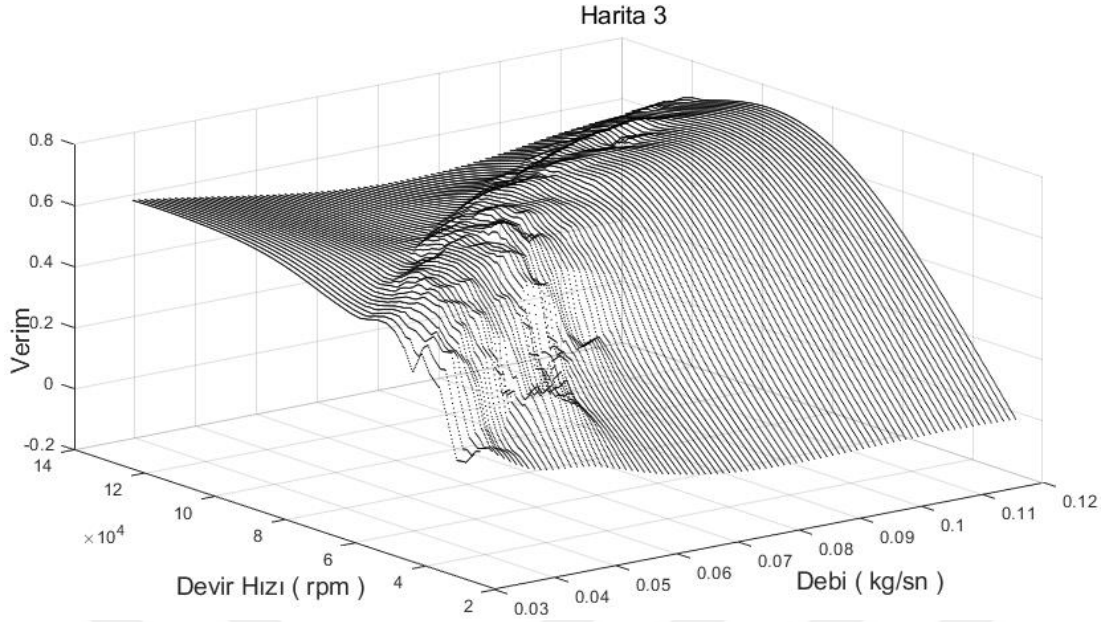
Şekil 4.2 : Devir hızı ve debiye bağlı olarak sıkıştırma oranını gösteren üç boyutlu kompresör veri haritası.

Şekil 4.2’de görüldüğü üzere genel olarak üç boyutlu doğrusal bir veri haritamız var. Öğrenilmesi kolay olan bu veri haritasında bazı noktalarda bükülmeler görülmektedir. Bu bükülme noktalarını da üreten bir öğrenme algoritması başarıyı artıracaktır. Harita 2 için detaylı analizi çizelge 4.2’de görebiliriz.

Çizelge 4.2 : İkinci veri haritasının analizi

HARİTA 2	Birinci Giriş	İkinci Giriş	Çıkış
Kompresör	Devir Sayısı (rpm)	Debi (kg/sn)	Basınç Oranı (pi)
Veri sayısı	35100	35100	35100
Tekrarsız veri	90	390	35100
En küçük değer	0.03	26000	0.9541747
En büyük değer	0.119	123250	2.202148558
Artış miktarı	0.001	250	
Ortalama	0.0745	74625	1.4244603020
Standart sapma	0.02597952	28145.9614	0.3428567767
Varyans	0.00067493589	792195147.80343	0.1175507693

4.1.3 Veri Haritası 3



Şekil 4.3 : Devir hızı ve debiye bağlı olarak kompresörün verimini gösteren harita.

Diğer iki veri haritamız ile kıyaslarsak Şekil 4.3’de görüldüğü üzere daha karmaşık ve öğrenilmesi daha zor olan bir veri haritamız mevcuttur. Bu arada kompresörün ve turbonun aynı mil üzerinden hareket eylemini aldığını düşünürsek devir hızlarının da aynı olduğunu rahatlıkla anlayabiliriz. Harita 3 için detaylı analizi çizelge 4.3’de görebiliriz.

Çizelge 4.3 : Üçüncü veri haritasının analizi

HARİTA 2	Birinci Giriş	İkinci Giriş	Çıkış
Kompresör	Devir Sayısı (rpm)	Debi (kg/sn)	Basınç Oranı (pi)
Veri sayısı	35100	35100	35100
Tekrarsız veri	90	390	35100
En küçük değer	0.03	26000	-0.01381762
En büyük değer	0.119	123250	0.744724855
Artış miktarı	0.001	250	
Ortalama	0.0745	74625	0.5334885886
Standart sapma	0.02597952	28145.9614	0.2025243781
Varyans	0.00067493589	792195147.80343	0.0410161237

4.2 Eğitim Modellerinin Oluşturulması ve Uygulanması

Bu tez çalışması kapsamında 3 farklı veri haritasına polinom regresyonu yöntemi, yapay sinir ağları ile öğrenme yöntemi uygulanmıştır. Harita 3 daha karmaşık bir yapıya sahip olduğu için en iyi öğrenme yöntemini bu harita üzerinden kıyaslamak için sadece polinom regresyonu ve yapay sinir ağları yöntemi değil bunların yanında karar ağaçları ve destek vektör makineleri yöntemleri de uygulanmıştır. Ayrıca eğitim verilerinin normalize edilip edilmediği durumlar ele alınmış normalize veriler ile eğitilen modellerin sonuca katkısı olmadığı durumlarda normalize edilmeyen veriler ile devam edilmiştir.

4.2.1 Polinom regresyonu yönteminin uygulanması

Bu tez çalışmasında 3 farklı veri haritası için polinom regresyonu yöntemi kullanılarak fonksiyonlar elde edilmiştir. Aynı veri kümelerine farklı dereceden polinom regresyonu yöntemi uygulanmış ve sonuçlar kıyaslanmıştır. Bunun yanı sıra bu verileri normalize ederek ve normalize etmeyerek elde edilen polinom regresyonu yönteminin sonuçları kıyaslanmıştır.

Bu tez çalışmasında 3 harita için giriş değerlerine normalizasyon işlemi uygulanarak ve normalizasyon işlemi uygulanmayarak her bir harita verileri için ikinci dereceden ve altıncı dereceden olmak üzere iki farklı polinom türü üretildi. İkinci dereceden polinomlar üretilirken giriş değerlerine normalizasyon işleminin uygulanması sonucu iyileştirme konusunda herhangi bir katkı sağlamadı. Ancak polinomun derecesi arttığı zaman örneğin bu tez çalışmasında altıncı dereceden polinomlar için giriş değerlerine normalizasyon işlemini uygulamanın öğrenme işlemini oldukça arttırdığı görüldü. Normalizasyon işlemi için giriş değerlerinden yine giriş değerlerinin ortalaması çıkarıldı. Daha sonra giriş değerlerinin standart sapmasının üç katına bölünerek normalizasyon işlemi gerçekleştirildi. Normalizasyon işlemi yapılarak öğrenme sağlandığı için tahmin sırasında gelen veri değerine bu normalizasyonda elde edilen ortalama ve standart sapma değerlerini kullanarak aynı işlem uygulanır ve sonuç bu değerlere göre hesaplanır. Bu tez çalışmasında polinom üretmek için Python programlama dili ile Scikit Learn kütüphanesinden yararlanıldı. Elde edilen polinomlar Matlab programında düzenlenerek sonuçlar elde edildi.

4.2.1.1 İkinci dereceden polinom fonksiyonun elde edilmesi

Normalizasyon işlemi uygulanmayan veriler rastgele karıştırıldıktan sonra dörtte üçü eğitim geri kalanı test olarak ayrılmıştır. Böylece 46215 adet eğitim verisi ve 15405 adet test verisi elde edilmiş oldu. İkinci dereceden polinom üretmek için her bir giriş değeri için denklem 4.1’de görüldüğü gibi ikinci dereceden özellikleri çıkarıldı.

$$[x_1, x_2, x_1^2, x_1x_2, x_2^2] \quad 4.1$$

Elde edilen yeni değişkenler ile bölüm 3.1’ de açıklandığı gibi en küçük kareler yöntemi kullanılarak 6 adet katsayı elde edildi. İlk terim sabit toplama katsayısını denk gelirken diğer terimler sırasıyla denklem 4.1’de gösterilen girişlerle çarpılacak katsayıları ifade etmektedir. Bu şekilde çarpma işlemi uygulanarak ikinci dereceden polinom üretildi. Bu yöntemle harita 1, harita 2 ve harita 3 verileri kullanılarak 3 farklı ikinci dereceden polinom üretildi.

4.2.1.2 Altıncı dereceden polinom fonksiyonun elde edilmesi

Polinomun derecesi arttığı zaman normalize işleminin gerekli olduğundan bahsedilmişti bu nedenle altıncı dereceden bir polinom elde ederken de öncelikle giriş değerleri normalize edildi ve rastgele karıştırıldı. İkinci dereceden polinom yönteminde olduğu gibi 46215 adet eğitim verisi ve 15405 adet test verisi kullanıldı. Altıncı dereceden polinom üretmek için iki girişli bir veri yapısının altıncı dereceden özellikleri denklem 4.2’ de görüldüğü gibi elde edildi. Derecesi altı olan iki girişli bir verinin en büyüğü altıncı dereceden olmak üzere iki girişin birbirleri ile çarpım kombinasyonlarını da içeren 27 adet özelliği oluşturuldu.

$$[x_1, x_2, x_1^2, x_1x_2, x_2^2, x_1^3, x_1^2x_2, x_1x_2^2, x_2^3, x_1^4, x_1^3x_2, x_1^2x_2^2, \dots, x_2^6] \quad 4.2$$

Altıncı dereceye kadar özellikleri çıkarılan veriler en küçük kareler yöntemi kullanılarak ilk terimi sabit toplama katsayısı olmak üzere 28 adet katsayı elde edildi. İlk terimden sonraki 27 terim denklem 4.2’de belirtilen katsayılar ile sırasıyla çarpılarak altıncı dereceden bir polinom üretildi. Harita 1 , harita 2 ve harita 3 için altıncı dereceden polinomlar üretildi.

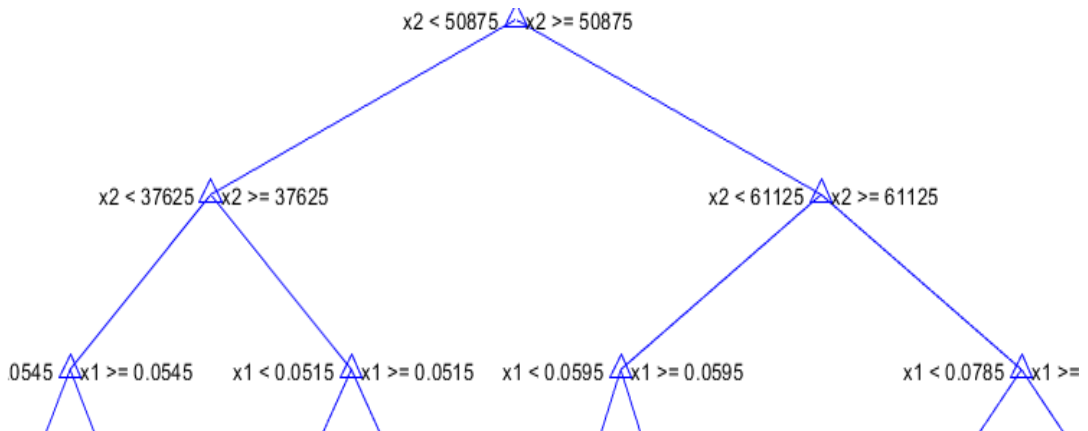
4.2.2 Destek vektör makineleri yönteminin uygulanması

Bu bölümde harita 3 için uygulanan destek vektör makineleri ile regresyon yönteminin nasıl uygulandığı anlatılacaktır.

Önceki sayfalarda harita 3'ün analizi yapılmıştı. Öncelikle 35100 adet olan veri sayısının dörtte üçü eğitim geri kalanı ise test için ayrıldı. Veriler normalize edilmedi ve rastgele olacak şekilde veriler karıştırıldı. Matlab programında yer alan regresyon destek vektör makineleri sınıfının fonksiyonları kullanıldı. Çekirdek olarak Gaussian fonksiyonu tercih edildi. Normalize edilmeyen veriler ile eğitimin daha uzun ve sonuçların daha kötü olduğu görüldü. Eğer verileri normalize ederek model eğitilirse tahmin aşamasında ham veriler bu normalizasyon değerlerine göre ön işlemden geçirilip öyle ağa verilmelidir. Ancak Matlab programında kullanılan “fitcsvm” fonksiyonu içerisine normalize edilmeyen verileri gönderir ve standartlaştırma işlemini aktif edersek kendi içerisinde normalizasyon işlemi yapacağı için tahmin aşamasında verileri ön işleme sokmaya gerek kalmaz.

4.2.3 Regresyon ağaçları yönteminin uygulanması

Bu bölümde harita 3 için uygulanan regresyon ağaçları yöntemi anlatılacaktır. Öncelikle 35100 adet olan veri sayısının dörtte üçü eğitim geri kalanı ise test için ayrıldı. Veriler normalize edilmedi ve rastgele olacak şekilde veriler karıştırıldı. Matlab programında yer alan fonksiyonlar kullanılarak 2165 düğümden oluşan ve sadece başlangıç kısmı şekil 4.4'de görülen bir regresyon karar ağacı modeli elde edilerek eğitildi.



Şekil 4.4 : Harita 3 için elde edilen regresyon karar ağacı modeli.

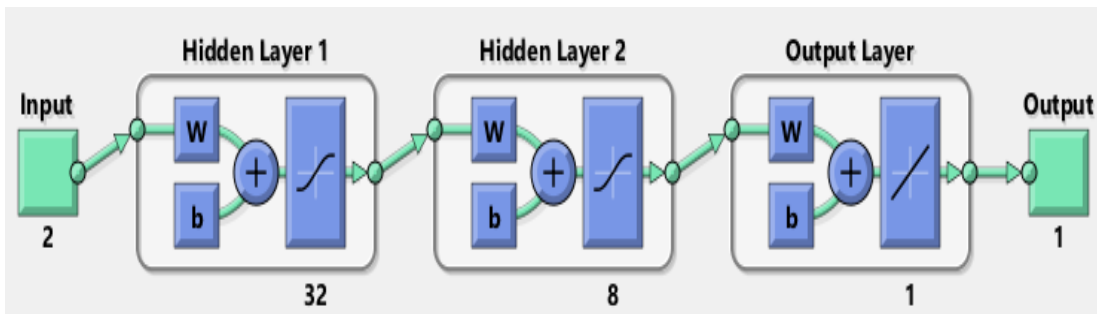
4.2.4 Yapay sinir ağlarının oluşturulması ve eğitilmesi

Bu bölümde önceki bölümlerde detaylı analizi verilen 3 veri haritası için yapay sinir ağlarının oluşturulması ve eğitilmesinden bahsedilecektir. Bu tez çalışmasında yapay sinir ağlarını oluşturmak için Matlab programı kullanılmıştır. Ağın oluşturulması ve eğitilmesinde hazır araç kutusu kullanılmamıştır. Ancak Matlab programının yapay sinir ağları için yazılan özel fonksiyonlarından yararlanılmıştır.

4.2.4.1 Yapay sinir ağlarının oluşturulması

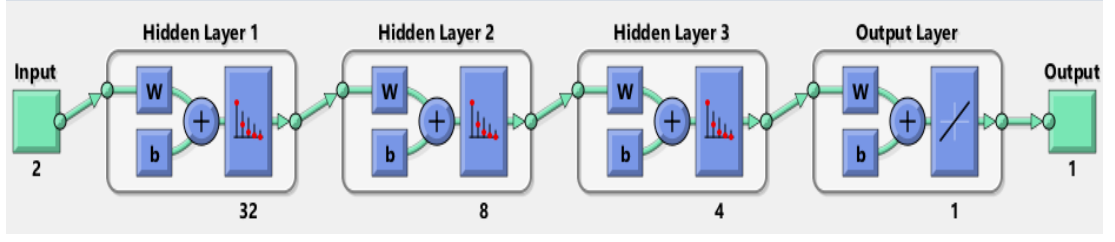
İlk haritada basınç oranını ve devir hızını gösteren 2 farklı sensör verisi ağa giriş olarak verilmiş ağın çıkışı ise debiyi gösteren sensör verileri olarak belirlenmiştir. Kısaca yapay sinir ağında basınç oranı ve devir hızı bağımsız girişler debi ise bağımlı çıkış olarak kullanılmıştır. İkinci haritada ise bağımsız giriş değişkenleri debi ve devir hızıyken bağımsız çıkış değişkeni sıkıştırma oranıdır. Üçüncü haritada giriş değişkenleri debi ve devir hızıyken çıkış değişkeni verimdir.

Harita 1 ve 2 için Matlab programında oluşturulan modelin görseli şekil 4.5’de görülmektedir. Giriş katmanı 32 adet, gizli katmanı 8 adet ve çıkış katmanı 1 adet nörondan oluşan bir yapay sinir ağı modeli oluşturulmuştur. Önceki bölümde detaylı açıklanan aktivasyon fonksiyonlarından hiperbolik tanjant ilk iki katman için kullanılmıştır.



Şekil 4.5 : Harita 1 ve 2 için oluşturulan yapay sinir ağı modeli.

Harita 3 daha karmaşık bir yapıya sahip olduğu için aynı modelin çok etkili sonuç vermediği görülmüştür ve 4 nörona sahip bir gizli katman daha eklenerek doğruluk oranı artırılmıştır. Ayrıca harita 3 için softmax aktivasyon fonksiyonları daha iyi sonuç verdiği için ilk üç katmanda hiperbolik tanjant aktivasyon fonksiyonları yerine tercih edilmiştir. Harita 3 için oluşturulan model şekil 4.6’da görülmektedir.



Şekil 4.6 : Harita 3 için oluşturulan yapay sinir ağı modeli.

4.2.4.2 Yapay sinir ağlarının eğitilmesi

Oluşturulan yapay sinir ağları eğitilmeden önce veriler eğitim, doğrulama ve test verisi olarak 3 kategoriye ayrılmalıdır. Eğitim verileri geriye yayılım ve optimizasyon algoritmalarında ağı öğrenmesi için kullanılırken doğrulama verisi ağı ezberleyip ezberlemediğinin kontrolü için kullanılmaktadır. Test verileri ise ağı eğitimi tamamlandıktan sonra öğrenme oranının tespit edilmesi için kullanılır. Genellikle % 60 – 80 arası eğitim % 10 – 20 arası doğrulama ve % 10 – 20 arası test verisi olarak tercih edilmektedir. Eğitilen tüm haritalar için bu oranlar ileriki bölümde yer alan çizelgelerde detaylı olarak görülecektir.

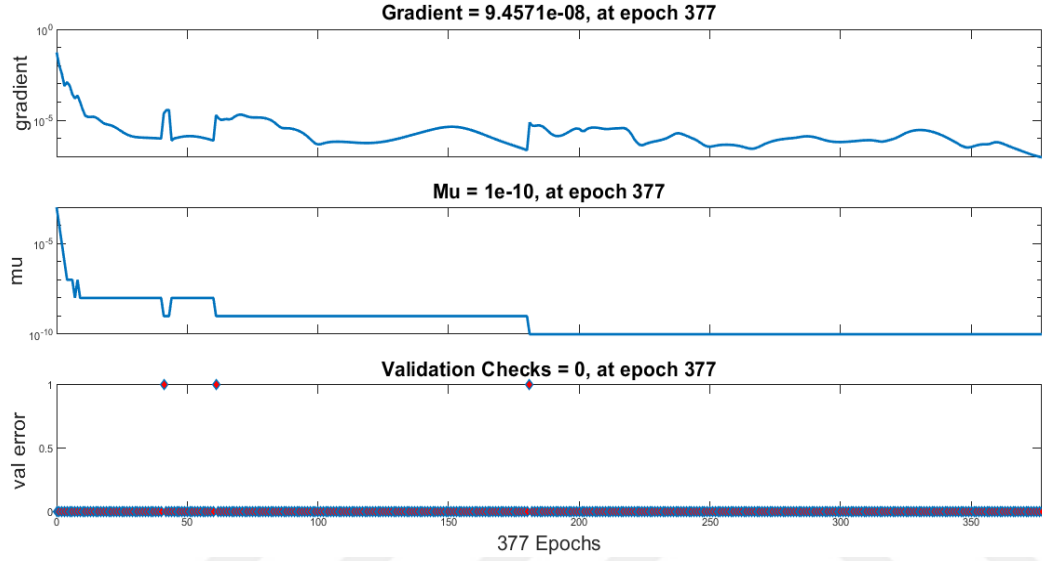
Ağ eğitilmeden önce eğitim hızını artırmak için ve ağı eğilirken öğrenme noktasının en iyi temsil edildiği global minimum noktaya ulaşmak için verilerin normalize edilmesi gerekmektedir. Normalizasyon işlemi için tüm veriler -1, 1 arasına gelecek şekilde tekrar düzenlenmiştir. Tüm verilerin ortalamasının 0 olacağı normalizasyon işlemi de tercih edilen diğer bir yöntemdir. Denklem 4.3’de bu tez çalışmasında kullanılan normalizasyon formülü görülmektedir. Girişler ve çıkışlar arasında normalizasyon yapılacağı gibi sadece giriş değerleri de birbirine göre normalize edilebilir. Bu tez çalışmasında iki giriş değeri birbirine göre normalize edilmiştir. Burada x ilk giriş y ikinci giriş olarak düşünülebilir.

$$y_{normalize} = \frac{(y_{max}-y_{min}) \times (x-x_{min})}{(x_{max}-x_{min})} + y_{min} \quad 4.3$$

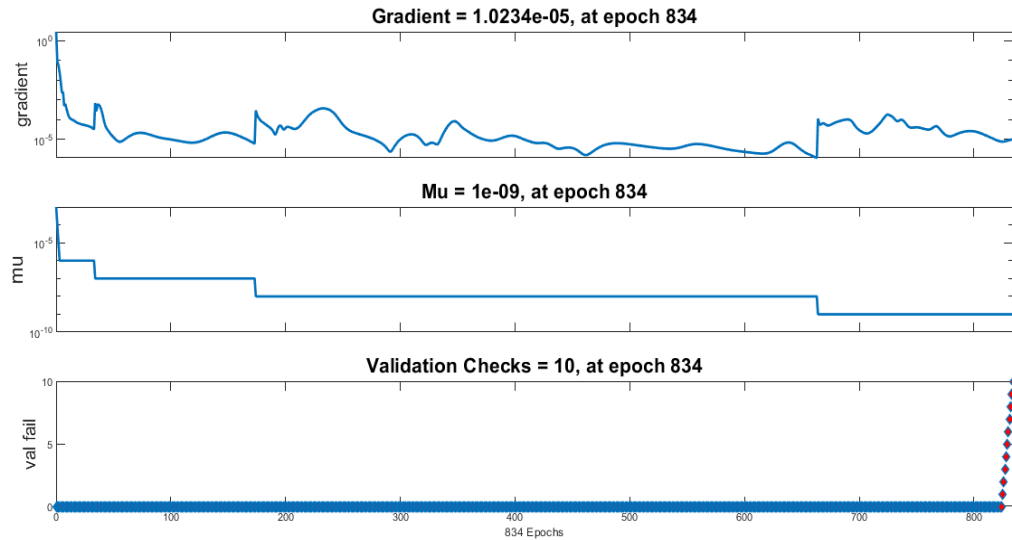
Yapay sinir ağlarının eğitilmesi için kullanılan birçok parametre vardır. Bunlardan en önemlileri optimizasyon yöntemi, ağı ağırlık katsayılarının değişme büyüklüğünü belirleyen öğrenme oranı, eğitimin maksimum tekrar sayısını belirleyen iterasyon (epoch) sayısı, ezberlemeyi kontrol etmeye yarayan doğrulama kontrol sayısı ve hedef öğrenme oranını temsil eden öğrenme hedef değeridir.

Tüm haritalarda optimizasyon yöntemi tüm denemeler sonucunda en iyi sonuç veren Levenberg-Marquardt seçilmiştir. Öğrenme oranı 0.01, doğrulama kontrol sayısı 10, maksimum iterasyon sayısı 5000 ve öğrenme hedef değeri 10^{-15} seçilmiştir.

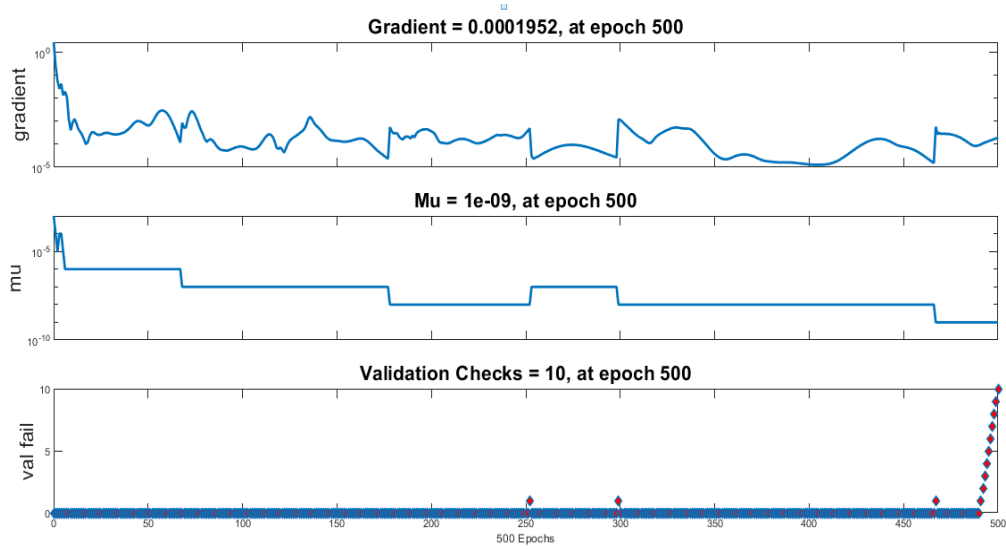
Tüm haritaların eğitim performansları şekil 4.7, 4.8 ve 4.9 üzerinden yorumlanabilir. Harita 1 en iyi doğruluk noktasına 377. İterasyonda ulaşırken harita 2 için bu değer 834 ve harita 3 için ise 500 olarak görülmektedir.



Şekil 4.7 : Harita 1 için eğitim performansı grafiği.

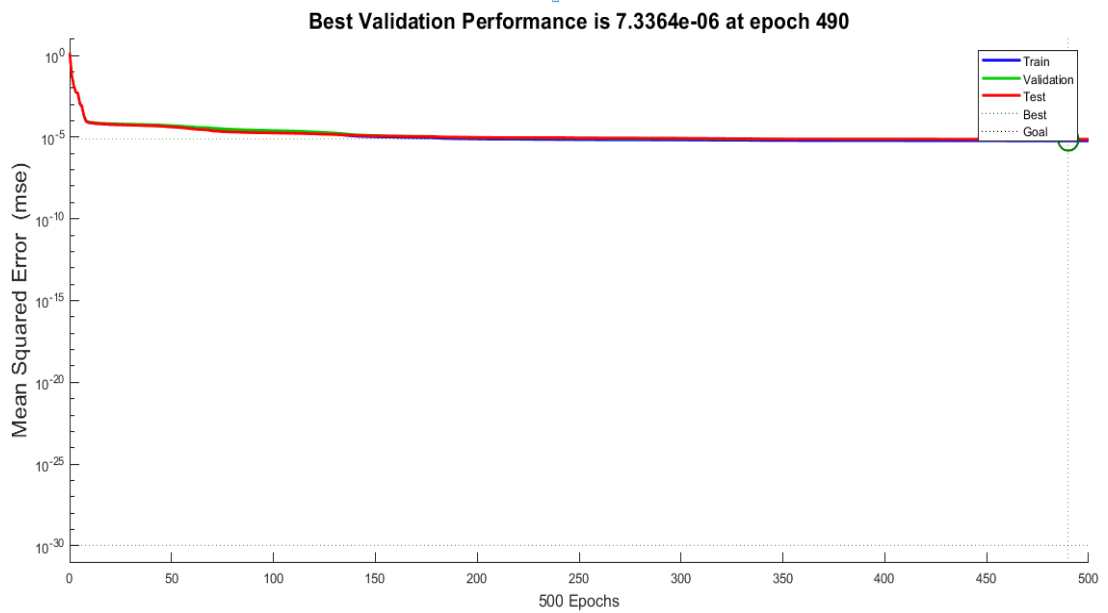


Şekil 4.8 : Harita 2 için eğitim performansı grafiği.

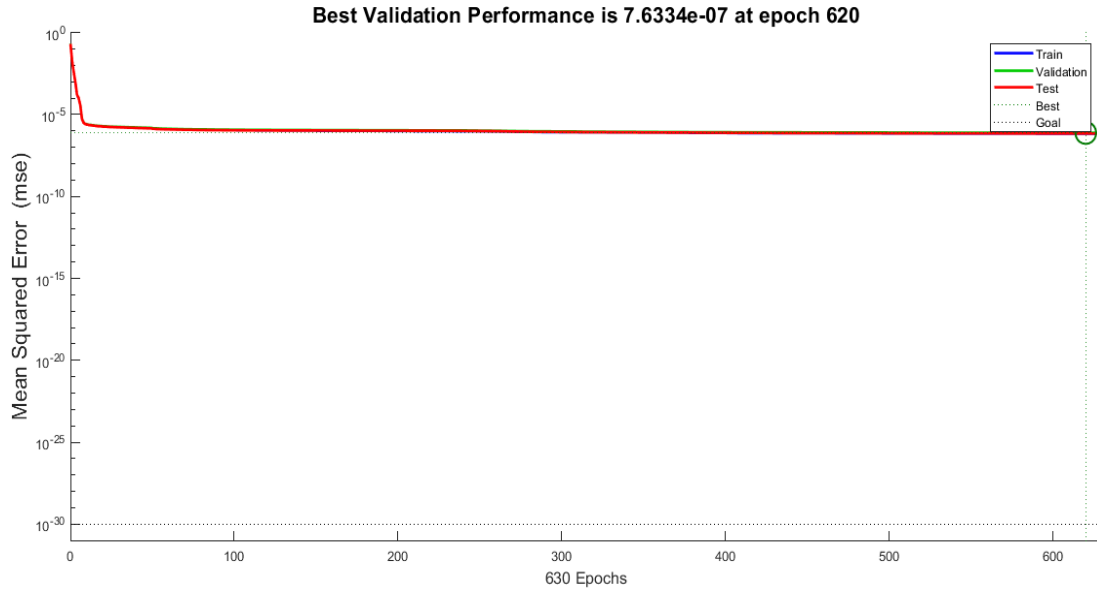


Şekil 4.9 : Harita 3 için eğitim performansı grafiği.

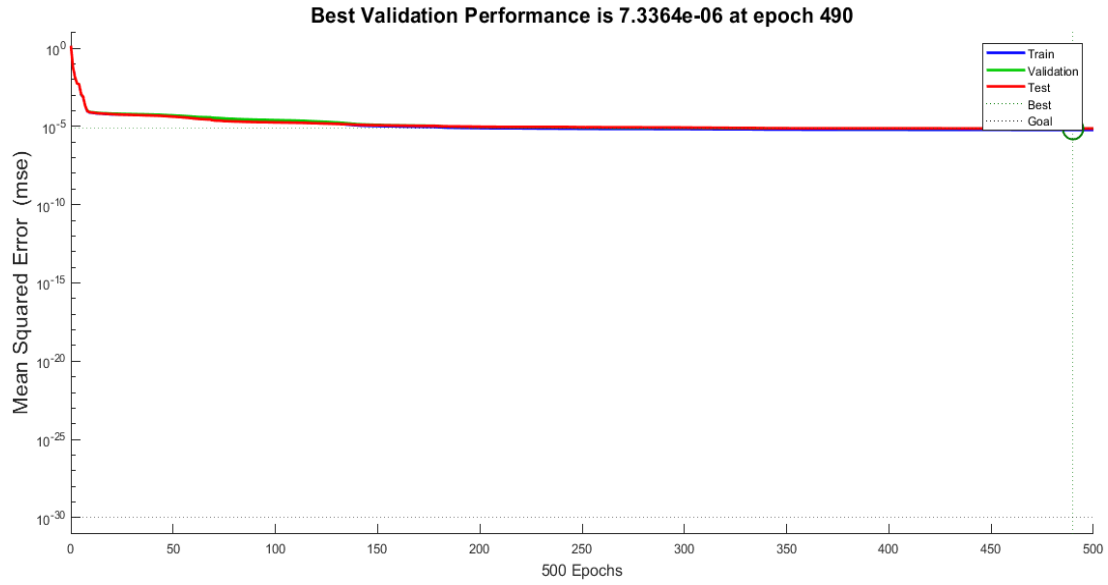
Son olarak iterasyon sayısı ile öğrenme arasındaki ilişkiyi gösteren grafikler 3 harita için sırasıyla şekil 4.10, 4.11 ve 4.12 üzerinden görülmektedir. Yatay eksen iterasyon sayısını gösterirken dikey eksen ortalama hataların karekökünü temsil etmektedir. Bu grafikte eğitim, doğrulama ve test verileri kullanılmış ve hepsi ayrı renklerde ifade edilmiştir. Ancak benzer değerleri işaret ettikleri için tek bir renk gibi ortaya çıkmaktadır.



Şekil 4.10 : Harita 1 için eğitim validasyon grafiği.



Şekil 4.11 : Harita 2 için eğitim validasyon grafiği.



Şekil 4.12 : Harita 3 için eğitim validasyon grafiği.

3 harita için de hemen hemen aynı iterasyon sayılarında ortalama hataların karekökü aynı değerlere ulaşmıştır. Ancak sadece ortalama hataların karekökü değerine bakarak öğrenmeyi garanti etmediğimiz için bu grafikler ilk etapta bir ön fikir elde etmemizi sağlamaktadır.

5. TÜM YÖNTEMLERİN SONUÇLARI

Bu tez çalışmasında modellerin sonuçlarını değerlendirmek için birçok performans göstergesi üretildi. Sonuçları vermeden önce bu performans göstergelerinin formülleri ve anlamları açıklanacaktır.

5.1. Performans Ölçütleri

Öncelikle hata değerimizin ne olduğunu belirtelim. Elimizde her harita için binlerce giriş ve çıkış değerleri yer almaktadır. Bunlara ek olarak eğittiğimiz her modelin tahmin sonuçları olacaktır. Herhangi bir modeli ele alacak olursak bu model için hata değeri elimizde bulunan gerçek değerden modelin tahmin ettiği değeri çıkararak bulunur. Bu işlem tüm veriler için sırasıyla uygulanarak bütün hata değerleri elde edilir. Hata değerini nasıl elde ettiğimizi açıkladıktan sonra şimdi kullanılan performans göstergelerini tanımlayabiliriz. Formüllerde kullanılan kısaltmaları denklem 5.1’den görebiliriz.

$$y_t = \text{Gerçek değer} \quad n = \text{Eleman sayısı} \quad e_t = \text{Hata değeri} \quad 5.1$$

5.1.1 Ortalama mutlak hata (Mean absolute error - MAE)

Bir modelin ortalama hatasını bulmak için basitçe tüm hata değerlerini topla ve eleman sayısına böl diyebiliriz. Ancak burada negatif ve pozitif sayılar birbirlerini etkileyeceği için doğru sonuç alamayız. Bu problemi çözmek için hatanın mutlak değerini almak gerekir. Ortalama mutlak hata performans ölçütü her bir hatanın mutlak değerini alıp birbirleriyle toplayarak daha sonra da eleman sayısına bölerek bir sonuç verir. Kısaca mutlak hata ortalaması olarak tanımlayabiliriz. Denklem 5.2’de ortalama mutlak hatanın formülünü görebiliriz. Ortalama mutlak hatanın en büyük dezavantajı çıkan sonucun genelleştirilememesidir yani veri kümesi değiştiği zaman çıkan sonucun o veri kümesine göre yorumlanması gerekir.

$$MAE = \frac{1}{n} \sum_{t=1}^n |e_t| \quad 5.2$$

5.1.2 Ortalama mutlak yüzde hatası (Mean absolute percentage error - MAPE)

Ortalama mutlak hatanın genelleştirilememesi sorununun çözümü için ortalama mutlak yüzde hatasını kullanabiliriz. Bu yöntemdeki en önemli fark denklem 5.3’de görüldüğü üzere hata değerlerinin toplanmadan önce gerçek değerlere bölünmesidir. Çıkan sonucu 100 ile çarparak cevabı yüzde olarak ifade edebiliriz. Bu tez sonuçlarında 100 ile çarpılmadan tablolara yazılmıştır. Mape değerinin düşük çıkması bize öğrenme ile ilgili iyi bir gözlem verebilir. Bu yöntemde dikkat edilmesi gereken bir nokta gerçek değerlerden birisi 0 olursa 0’a bölünme hatası ortaya çıkacaktır.

$$MAPE = \frac{100\%}{n} \sum_{t=1}^n \left| \frac{e_t}{y_t} \right| \quad 5.3$$

5.1.3 Ortalama kare hatası (Mean squared error - MSE)

Diğer bir hata ölçüm yöntemi denklem 5.4’de de görüldüğü üzere hataların karelerinin ortalamasını almaktır. Sıfıra yakın sonuçlar iyi yorumlanabilir. Fakat bu yöntemin en büyük dezavantajı ise her hangi bir yüksek hata değerinin karesini almanın toplam sonuçta büyük bir etkiye sahip olmasıdır.

$$MSE = \frac{1}{n} \sum_{t=1}^n e_t^2 \quad 5.4$$

5.1.4 Kare Ortalamalarının Karekökü (Root mean square - RMSE)

Rmse genellikle verilerin hareketini anlamak ve yorumlamak için kullanılan bir yöntemdir. Denklem 5.5’ten de görüldüğü gibi Mse’den tek farkı çıkan sonucun karekökünü almaktır.

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n e_t^2} \quad 5.5$$

5.1.5 R² Skoru (R - Squared)

Bu dört yöntem sonuçlarımızı yorumlamak için bize yardımcı olmaktadır. Bunların yanı sıra istatistik, ekonomi, makine öğrenmesi ve derin öğrenme gibi birçok alanda modelin çıktılarını değerlendirmek için en çok kullanılan yöntemlerden biri R-kare skorudur.

R-kare skorunu hesaplamak için İngilizce kısaltmasıyla SSt ve SSe değerlerinin bulunması gerekir. Denklem 5.6’dan görüleceği üzere SSt her bir eleman için gerçek değerlerin gerçek değerlerin ortalamasından çıkartılıp toplanmasıyla elde edilen değerdir. Sse ise denklem 5.7’den görüleceği üzere her bir eleman için gerçek değerlerin modelin tahmin ettiği çıkıştan çıkarılıp birbirleri ile toplanmasıyla elde edilen değerdir.

$$SSt = \sum_{t=1}^n (y_t - y_{ortalama})^2 \quad 5.6$$

$$SSe = \sum_{t=1}^n (y_t - y_{tahmin})^2 \quad 5.7$$

SSe ve SSt elde edildikten sonra denklem 5.8'den de görüleceği üzere bu iki değer oranlanır ve eksi 1 ile toplanır çıkan sonuç R-kare skorunu verecektir. R-kare skorunun 1'e yakın olması modelin iyi öğrendiğini gösterir.

$$R^2 = 1 - \frac{SSe}{SSt} \quad 5.8$$

5.2. Polinom Regresyonu Yönteminin Sonuçları

Bu başlık altında 3 veri haritası için ikinci ve altıncı dereceden polinomların sonuçları verilecektir. Polinom regresyonu yönteminde hangi dereceden polinomun daha iyi sonuç vereceği deneme yanılma yöntemiyle belirlenir. Bu tez çalışması için kullanılan verilerde en iyi sonuç altıncı dereceden polinomdan elde edilmiştir. Derecenin artması hesaplama karmaşıklığını da artırmaktadır. Ayrıca ikinci dereceden polinomlarda normalizasyon işlemine gerek kalmamış altıncı dereceden polinomlarda normalizasyon işlemi sonucu ciddi anlamda iyileştirmiştir.

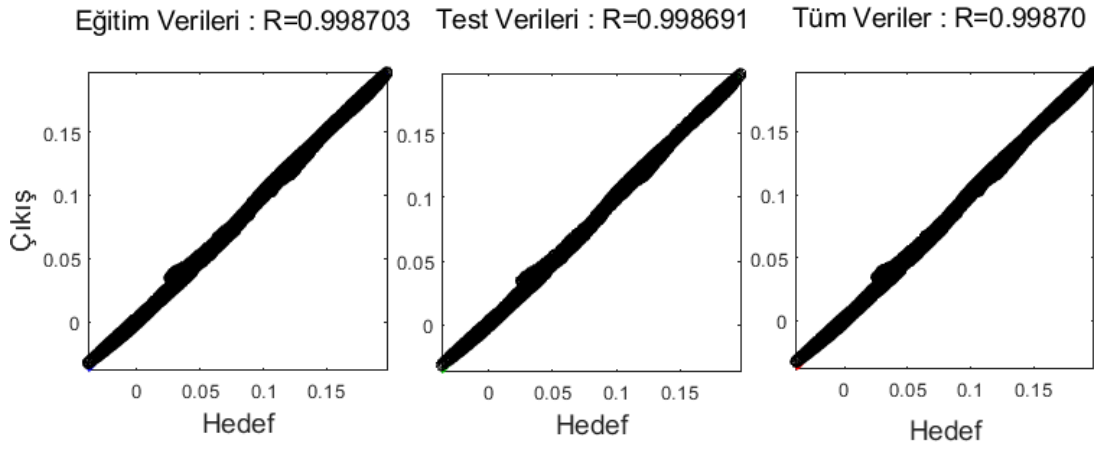
5.2.1 İkinci dereceden polinom regresyonu ile harita 1 için sonuçlar

Harita 1 ile ikinci dereceden polinom için sadece eğitim verisi, sadece test verisi ve tüm veriler kullanılarak elde edilen sonuçlar çizelge 5.1'de görülmektedir.

Çizelge 5.1 : Harita 1 için ikinci dereceden polinom regresyonu sonuçları.

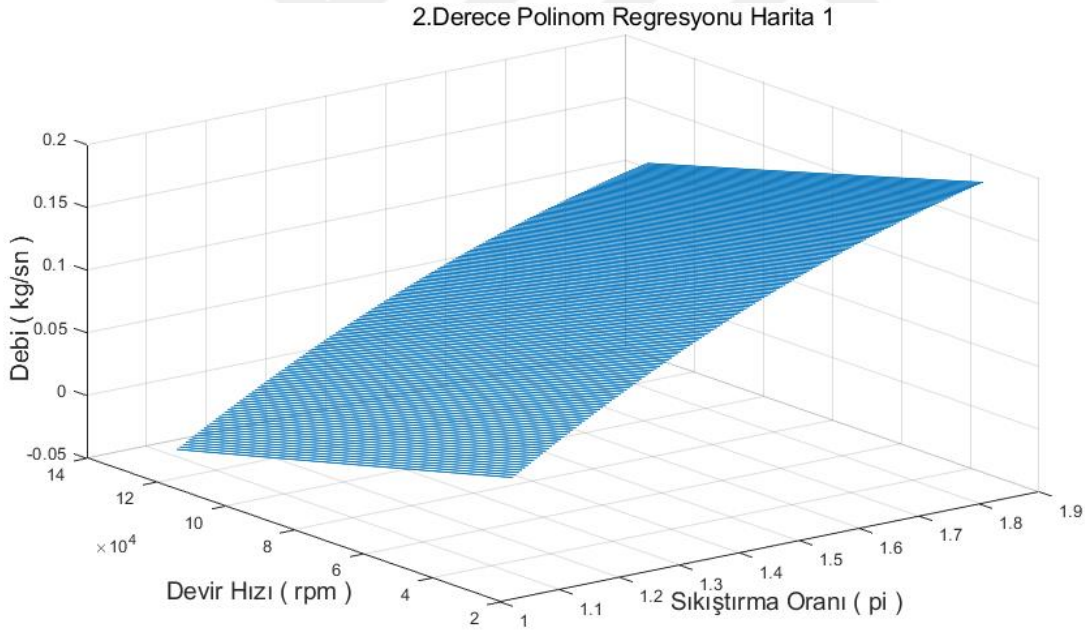
Harita 1	Eğitim Verisi	Test Verisi	Tüm Veriler
İkinci Derece	(46125)	(15405)	(61620)
MAE	0.0013781	0.001383836	0.001379601
MAPE	0.020272899	0.005020023	0.0164374023
MSE	3.264×10^{-6}	3.72×10^{-6}	3.26×10^{-6}
RMSE	0.0018067	0.00180905	0.00180733

Çizelge 5.1'de görüldüğü üzere harita 1 çok karmaşık bir yapıya da sahip olmadığı için ikinci dereceden bir polinom bile iyi sonuç verdiği görülüyor. Daha doğru bir yorum yapmak için şekil 5.1'de yer alan R2 skorlarını da incelemeliyiz. Sonuçlar grafik üzerinde yuvarlak olarak işaretlenmektedir ancak veri sayısı çok fazla olduğu ve sonuçlar üst üste bindiği için biz bunu düz bir çizgi olarak görüyoruz.



Şekil 5.1 : Harita 1 için ikinci dereceden polinom R^2 skorları.

R^2 skorları da yorumumuzu doğruladı ve son olarak şekil 5.2’de ikinci derece polinomumuzun tüm veriyi kullanarak ürettiği üç boyutlu çıktısını görebiliriz. Orijinal verilerden oluşan harita 1 ile kıyaslarsak bazı kıvrımlı noktaların öğrenilmediğini görebiliyoruz.



Şekil 5.2 : Harita 1’in tüm verileri ile ikinci dereceden polinom regresyonun çıktısı.

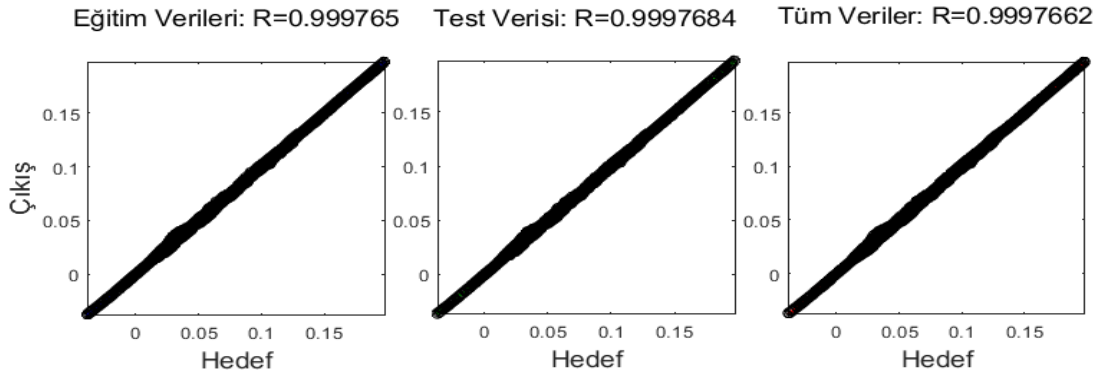
5.2.2 Altıncı dereceden polinom regresyonu ile harita 1 için sonuçlar

Bu kısımda ise aynı verileri kullanarak elde ettiğimiz altıncı dereceden polinom regresyonu sonucu yer almaktadır. Çizelge 5.2 ile çizelge 5.1 kıyaslanırsa altıncı dereceden polinom regresyonu sonuçlarının daha iyi olduğu anlaşılabilir.

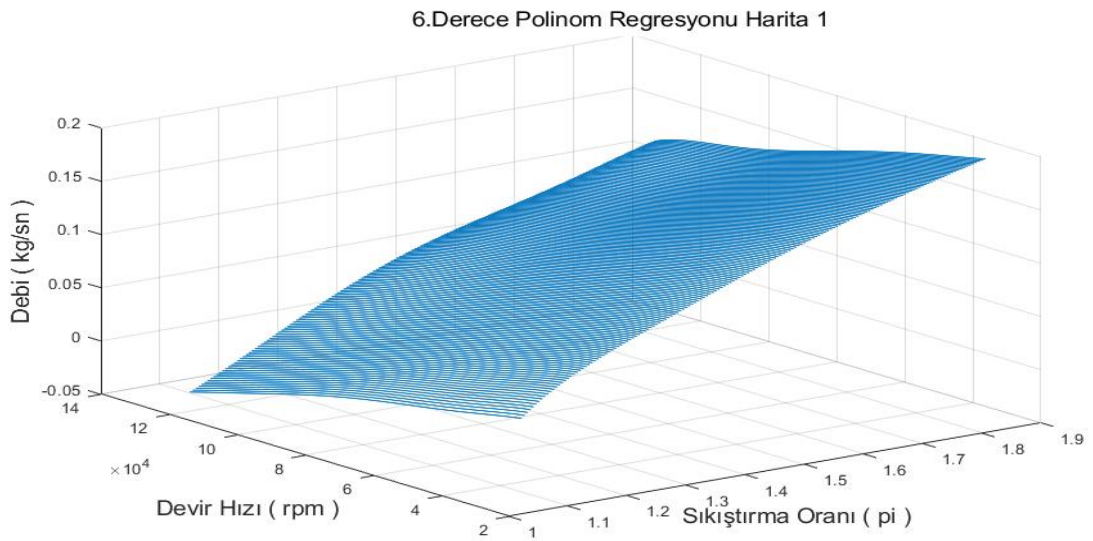
Çizelge 5.2 : Harita 1 için altıncı dereceden polinom regresyonu sonuçları.

Harita 1	Eğitim Verisi	Test Verisi	Tüm Veriler
Altıncı Derece	(46125)	(15405)	(61620)
MAE	5.078×10^{-4}	5.0508×10^{-4}	5.0716×10^{-4}
MAPE	0.008752	0.0071039	0.0083375
MSE	5.907×10^{-7}	5.788×10^{-7}	5.877×10^{-7}
RMSE	7.685×10^{-4}	7.608×10^{-4}	7.666×10^{-4}

R2 skorları üzerinden yorum yapmak daha doğru olacağı için şekil 5.1 ile şekil 5.3 kıyaslanırsa altıncı dereceden polinomun ürettiği sonuçların daha doğru olduğu görülür. Son olarak şekil 5.4'e bakılırsa kıvrımlı yapıların tamamen öğrenilmediğini ancak şekil 5.2'den daha doğru bir sonuç ürettiğini görebiliriz.



Şekil 5.3 : Harita 1 için altıncı dereceden polinom R2 skorları.



Şekil 5.4 : Harita 1'in tüm verileri ile altıncı dereceden polinom regresyonun çıktısı.

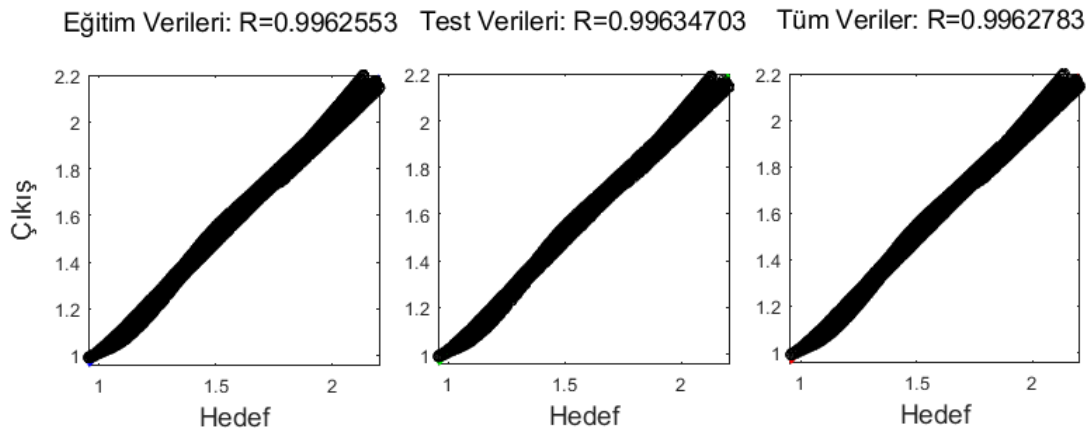
5.2.3 İkinci dereceden polinom regresyonu ile harita 2 için sonuçlar

Harita 1 için yaptığımız tüm işlemleri harita 2'nin verilerini kullanarak bu kısımda gerçekleştireceğiz. Öncelikle çizelge 5.3'ten sonuçları görelim. Çok fazla kıvrımlı yapılara ve aniden artış azalış değerlerine sahip olmadığı için harita 2 diğerlerine göre öğrenilmesi nispeten daha kolay diyebiliriz.

Çizelge 5.3 : Harita 2 için ikinci dereceden polinom regresyonu sonuçları.

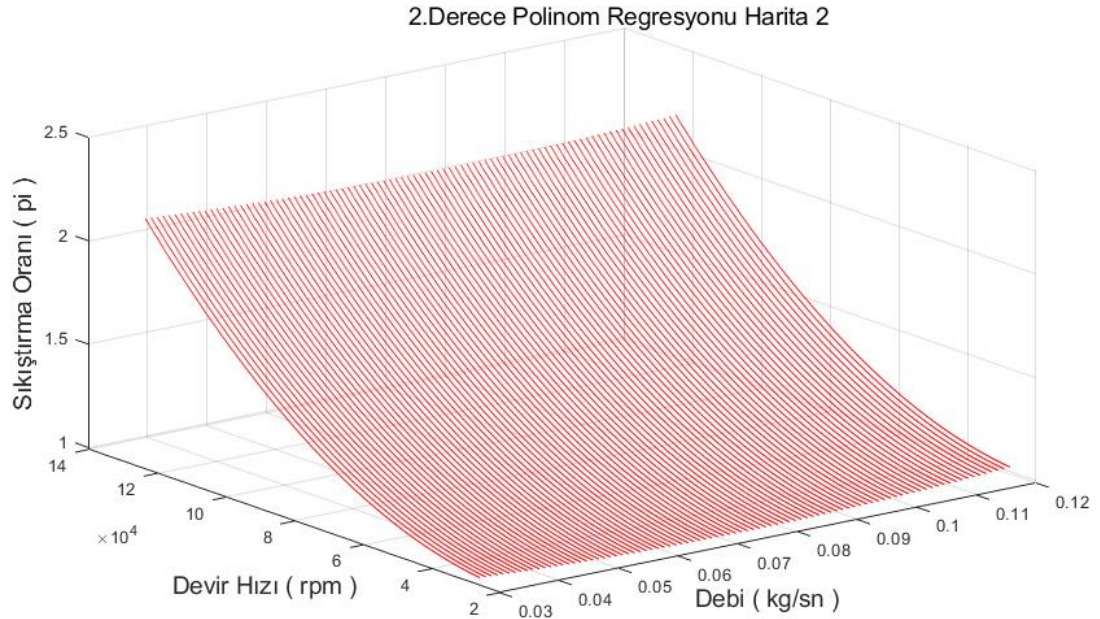
Harita 2	Eğitim Verisi	Test Verisi	Tüm Veriler
İkinci Derece	(26325)	(8775)	(35100)
MAE	0.016792073	0.016668277	0.016761124
MAPE	0.011750337	0.011685826	0.011734209
MSE	4.3935×10^{-4}	4.3179×10^{-4}	4.37465×10^{-4}
RMSE	0.02096084	0.02077955	0.02091567

Şekil 5.5'e bakarak ikinci dereceden polinomun R2 skorlarını görebiliriz. Sonuçların düz bir çizgi gibi doğrusal olması ve R2 skorlarının 1'e yakın olması öğrenme oranının yüksek olduğunu gösterir. Bu şekilden görüldüğü üzere sonuçlar tatmin edici boyutta diyebiliriz. Eğitim verileri polinomun üretilirken kullanılan verileri göstermektedir. Test verileri polinom katsayıları elde edildikten sonra modelin sonuçlarını görmek için eğitim sırasında hiç kullanılmayan verileri ifade etmektedir. Son olarak tüm veriler ise hem test verilerini hem de eğitim verilerini kapsayan tüm bu değerlerin birlikte olduğu verileri kapsamaktadır.



Şekil 5.5 : Harita 2 için ikinci dereceden polinom R2 skorları.

Son olarak şekil 6.6'ya bakarak harita 2 için eğitilen ikinci dereceden polinomun tüm verileri kullanarak elde edilen üç boyutlu görselini görebiliriz.



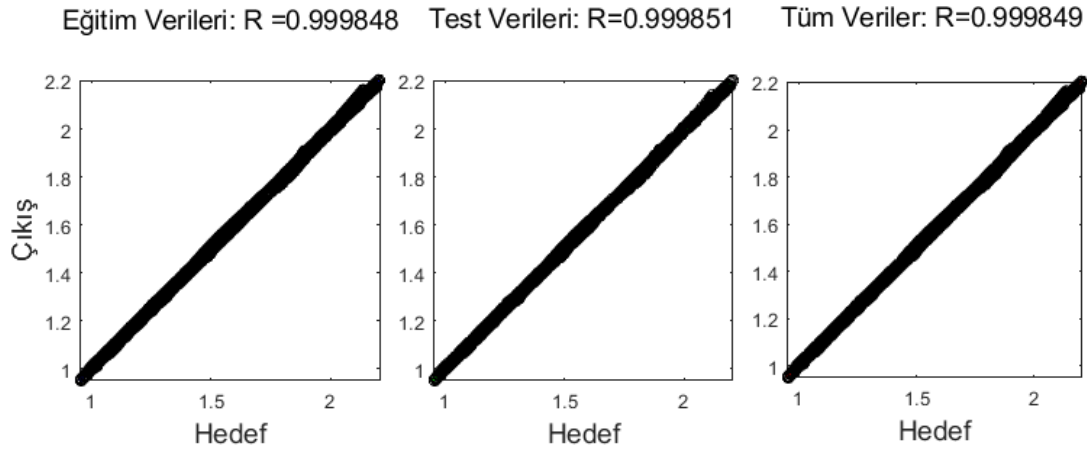
Şekil 5.6 : Harita 2'nin tüm verileri ile ikinci dereceden polinom regresyonun çıktısı.

5.2.4 Altıncı dereceden polinom regresyonu ile harita 2 için sonuçlar

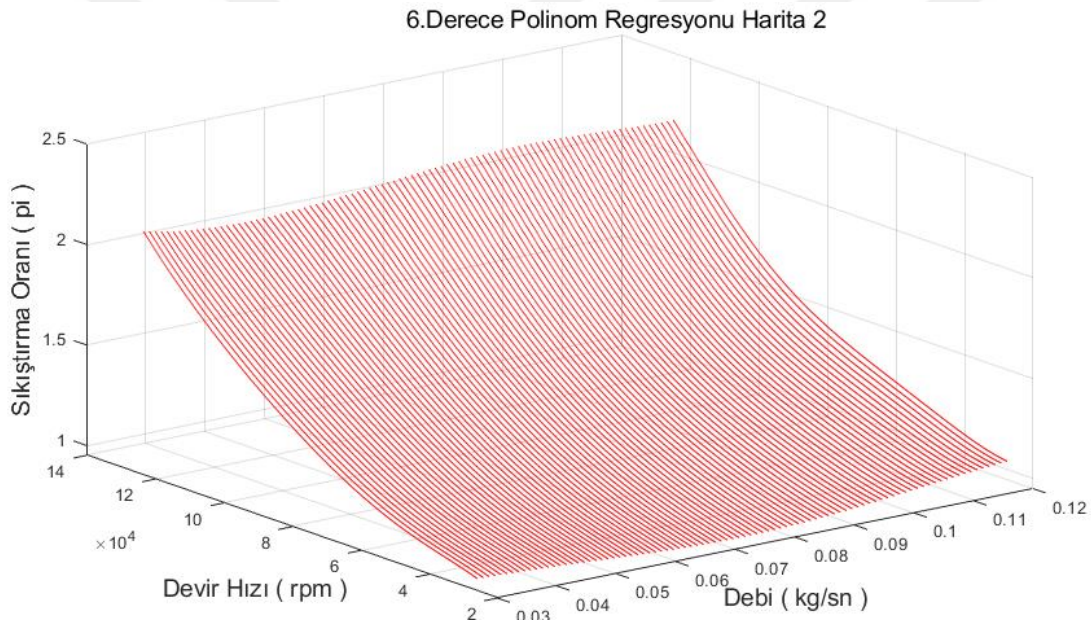
Harita 2 için altıncı dereceden elde edilen polinom sonuçlarını çizelge 5.4 ve şekil 5.7'den görebiliriz. Tıpkı harita 1 için olan ikinci ve altıncı dereceden polinomların sonuçları gibi harita iki içinde polinomun derecesinin artması sonucu ciddi anlamda iyileştirmiştir diyebiliriz. Çizelge 5.3 ve çizelge 5.4 arasında tüm metriklerde anlamlı farklar vardır. Bunun yanında R2 skorlarına bakacak olursak da şekil 5.5 ve şekil 5.7 aradaki farkı gösterecektir. Yine şekil 5.6 ile 5.8'e bakarak üç boyutlu grafikler üzerinden bir kıyaslama yaparsak iki polinomun ürettiği çıktılarda özellikle bitme noktalarındaki fark görsel olarak ayırt edilebilir.

Çizelge 5.4 : Harita 2 için altıncı dereceden polinom regresyonu sonuçları.

Harita 2	Eğitim Verisi	Test Verisi	Tüm Veriler
Altıncı Derece	(26325)	(8775)	(35100)
MAE	0.003265032	0.003259364	0.003263615
MAPE	0.0023298759	0.0023287818	0.0023296024
MSE	1.77859×10^{-5}	1.75265×10^{-5}	1.77210×10^{-7}
RMSE	0.00421733	0.004186467	0.004209639



Şekil 5.7 : Harita 2 için altıncı dereceden polinom R2 skorları.



Şekil 5.8 : Harita 2'in tüm verileri ile altıncı dereceden polinom regresyonun çıktısı.

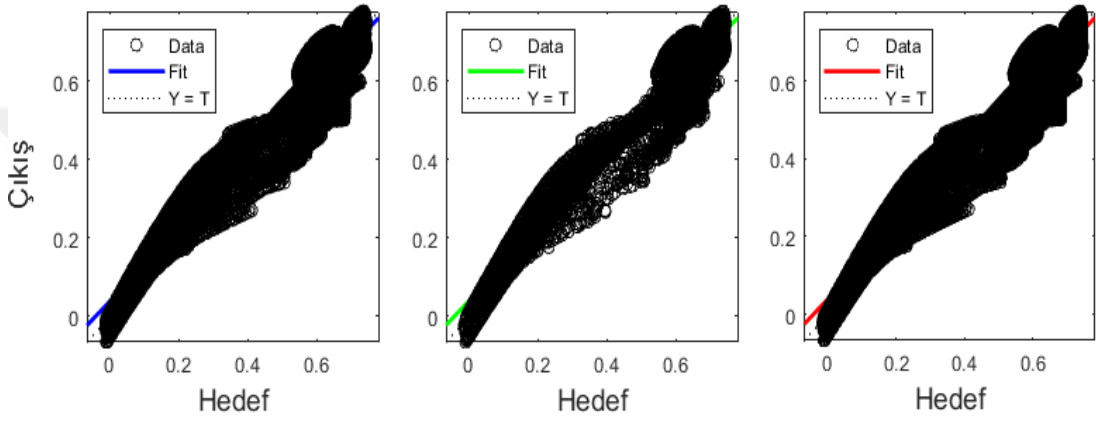
5.2.5 İkinci dereceden polinom regresyonu ile harita 3 için sonuçlar

Harita 3'ün diğerlerine göre daha karmaşık bir yapıya sahip olduğu belirtilmişti. Bu nedenle harita 3'ün verileri ile eğitilen ikinci dereceden bir polinomun iyi sonuç vermediğini çizelge 5.5 ve şekil 5.9'dan görebiliriz. Diğer iki haritanın çizelgelerine bakarsak sonuçların çok daha kötü olduğu rahatlıkla anlaşılabilir. Bunun yanı sıra R2 skorlarının 1'den uzak olduğu ve iki boyutlu grafiğinin doğrusal olmaktan uzak olduğunu anlayabiliriz. Sonuç olarak ikinci dereceden bu polinomu modelimize uygulamak pek mantıklı olmayacaktır.

Çizelge 5.5 : Harita 3 için ikinci dereceden polinom regresyonu sonuçları.

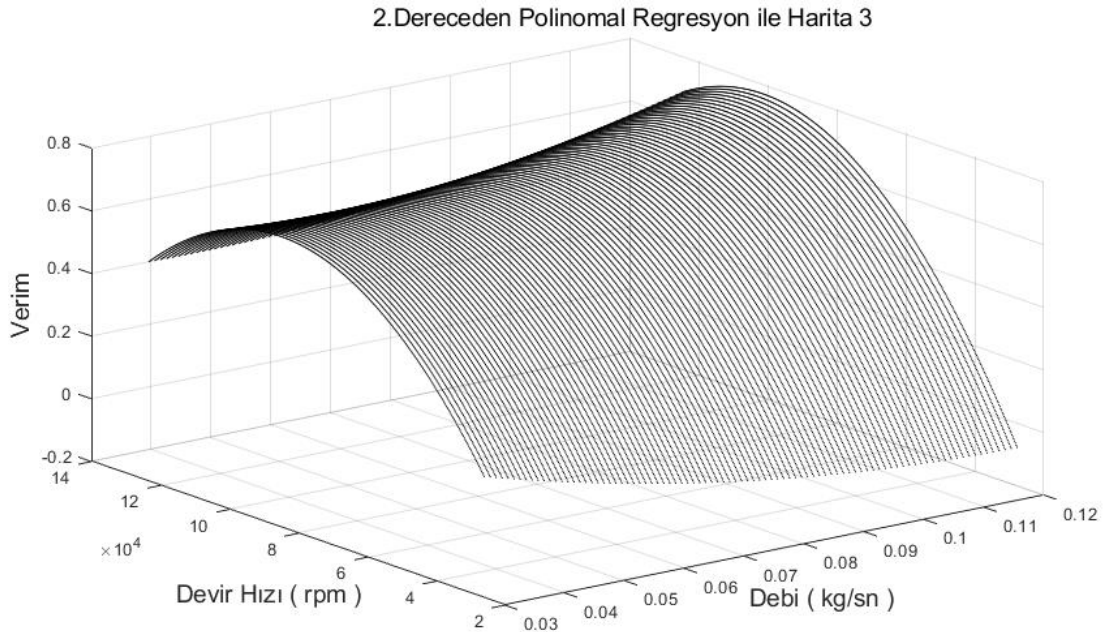
Harita 3	Eğitim Verisi	Test Verisi	Tüm Veriler
İkinci Derece	(26325)	(8775)	(35100)
MAE	0.04270849	0.04306750	0.042798244
MAPE	0.11625081	0.02547518	0.093556907
MSE	0.0029116	0.0029625	0.00292436
RMSE	0.05395966	0.05442891	0.054077362

Eğitim Verileri: R=0.92878071 Test Verileri: R=0.928460971 Tüm Veriler: R=0.9287001



Şekil 5.9 : Harita 3 için ikinci dereceden polinom R2 skorları.

Şekil 5.10’da ise 3 boyutlu görsel çıktısına bakarak sonucun iyi olmadığını görebiliriz.



Şekil 5.10 : Harita 3’ün tüm verileri ile ikinci derecen polinom regresyonun çıktısı.

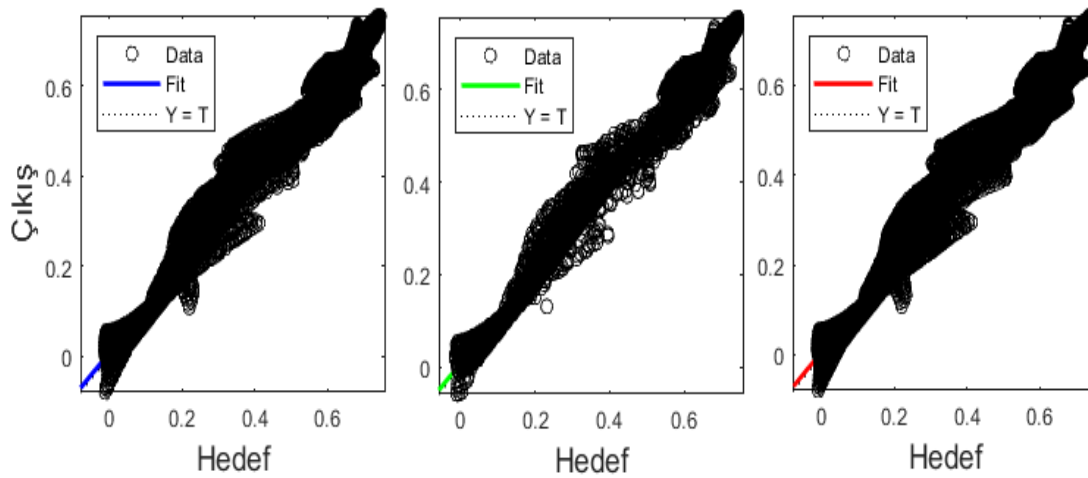
5.2.6 Altıncı dereceden polinom regresyonu ile harita 3 için sonuçlar

Polinom regresyonu ile ilgili olarak son olarak harita 3'ün verileri ile altıncı dereceden bir polinom üretilir. Üretilen bu polinomun sonuçlarını çizelge 5.6 ve şekil 5.11'den görebiliriz. Diğer sonuçlarda olduğu gibi altıncı dereceden polinomun ikinci dereceden polinoma göre ciddi anlamda daha iyi sonuç verdiğini görebiliyoruz.

Çizelge 5.6 : Harita 3 için altıncı dereceden polinom regresyonu sonuçları.

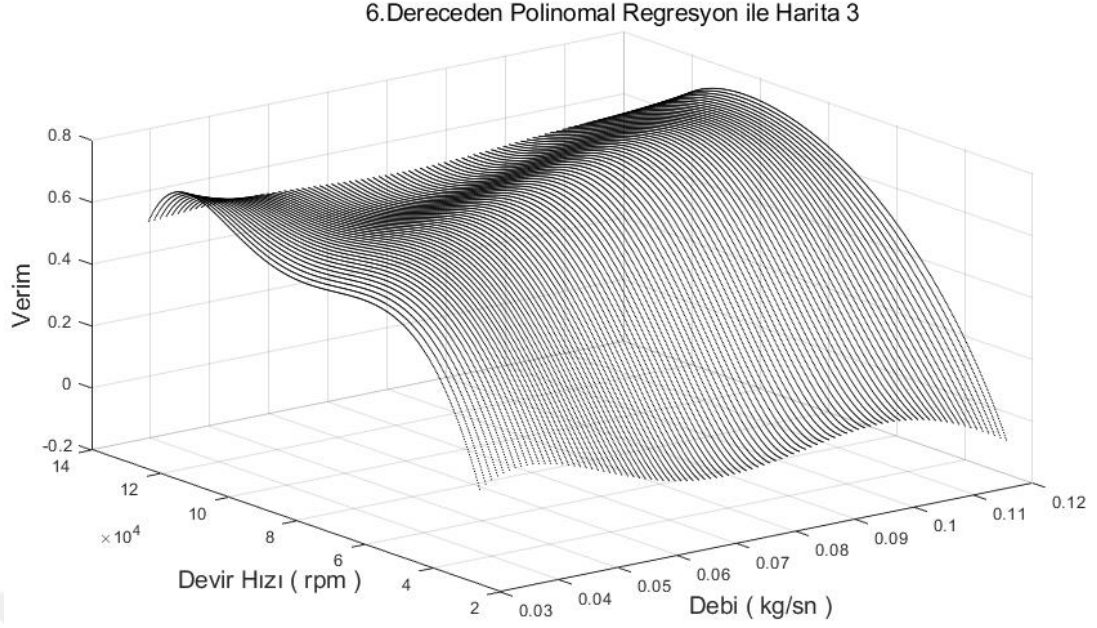
Harita 3	Eğitim Verisi	Test Verisi	Tüm Veriler
Altıncı Derece	(26325)	(8775)	(35100)
MAE	0.016940384	0.017029507	0.016962665
MAPE	0.029531894	-0.051629075	0.009241651
MSE	6.4256×10^{-4}	6.5060×10^{-4}	6.4457×10^{-4}
RMSE	0.025348907	0.025506956	0.025388512

Eğitim Verileri: R=0.9842827 TestVerileri: R=0.9842891 Tüm Veriler: R=0.984284



Şekil 5.11 : Harita 3 için altıncı dereceden polinom R2 skorları.

Çizelge 5.6 ve şekil 5.11'e bakarsak ikinci derece fonksiyona göre iyileşme sağlanmak ile birlikte diğer haritalardaki sonuçların doğruluk değerlerine erişemediğini görürüz. Görsel olarak sonucu görmek için şekil 5.12'ye de bakabiliriz. Harita 3 için polinom regresyonu yöntemini tercih etmek pek iyi olmayacaktır. Bu nedenle diğerlerinden daha karmaşık olan harita 3 için farklı makine öğrenmesi yöntemlerinden elde edilen model ya da derin öğrenme yöntemi ile elde edilen modelin sonuçlarına bakarak en etkili olanı dizel motorun modeline koymak gerekecektir.



Şekil 5.12 : Harita 3’ün tüm verileri ile altıncı derecen polinom regresyonun çıktısı.

Sonuç olarak polinom harita 1 ve 2 için tatmin edici sonuçlar verirken , harita 3’te çok iyi sonuç vermemiştir. Diğer bölümlerde ağırlıklı olarak harita 3 üzerinde çalışmalar yaparak farklı yöntemlerin sonuçları incelenecektir.

5.3 Karar Ağaçları Yönteminin Sonuçları

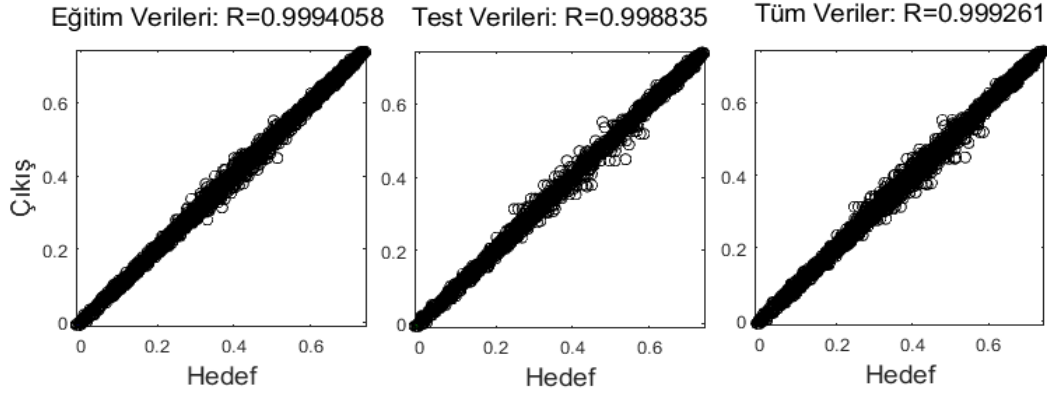
Harita 3 için karar ağaçları (regresyon ağaçları) yöntemi kullanılarak elde edilen değerlendirme metriklerinin sonuçları çizelge 5.7 ‘de yer almaktadır.

Çizelge 5.7 : Harita 3 için regresyon ağaçları modelinin sonuçları.

Harita 3	Eğitim Verisi	Test Verisi	Tüm Veriler
Regresyon ağacı	(26325)	(8775)	(35100)
MAE	0.0034688	0.00451846	0.00373123
MAPE	0.0071939	-0.008784319	0.003199373
MSE	2.428×10^{-5}	4.8222×10^{-5}	3.0272×10^{-5}
RMSE	0.00492844	0.00694423	0.00550206

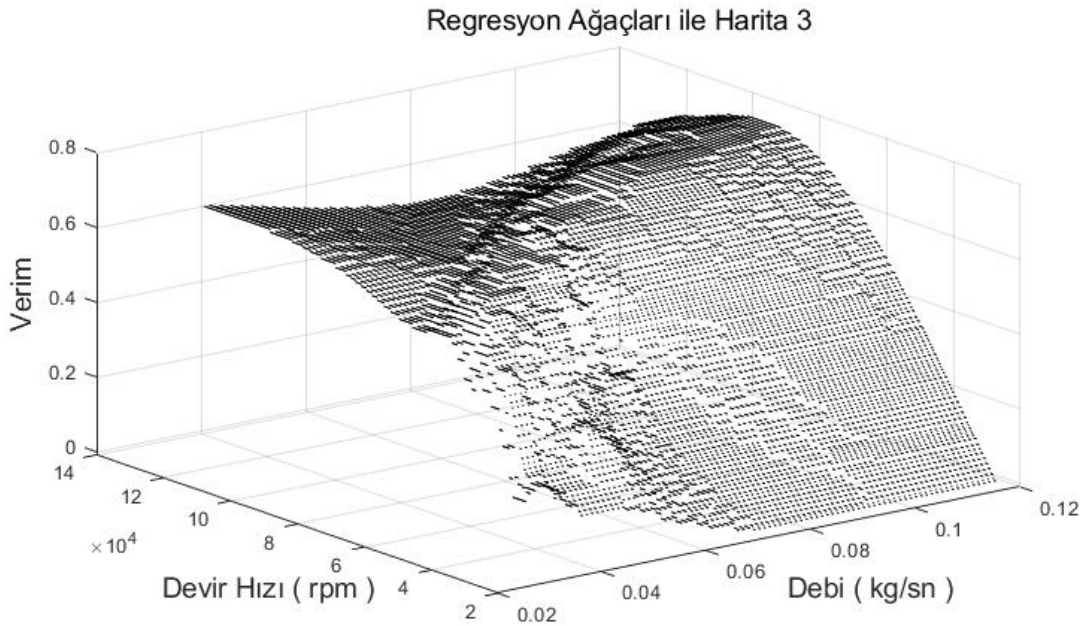
Yukarıdaki değerlendirme metriklerine göre modelimizin harita 3’ü iyi bir şekilde öğrendiği görülüyor. Bu değerlendirmelere ek olarak ve modelin doğruluğunu öğrenmek için R2 skoruna da şekil 5.13’den bakarak daha doğru bir yorum yapabiliriz.

Önceki bölümlerde R2 skoru değerinin 1'e yakın olmasının ve grafiğinin doğrusal olmasının modelin iyi öğrendiği anlamına geldiği belirtilmişti. Dolayısıyla regresyon ağacı yönteminin bu harita için başarılı bir sonuç verdiğini ortaya koyabiliriz.



Şekil 5.13 : Harita 3 için regresyon ağaçları modelinin R2 skorları.

Harita 3 için regresyon ağaçları yönteminin sonucunu görsel olarak şekil 5.14'de görebiliriz. Bu şekil incelenirse karar ağaçları yönteminin bir dezavantajı olan küçük kesik çizgiler ile geçişlerin olduğu fark edilmektedir. Burada yumuşak geçişlerin olmama sebebi ise tamamen regresyon ağaçları yönteminin doğasından kaynaklanır. Modelimiz sonuca giderken devamlı olarak iki değer arasında sorgulama yapar ve birini seçer bu da her zaman yüksek hassasiyetli bir sonuca ulaşmasına engel olur. Modelin derinliği kesik çizgilerin görünümünde azalma sağlayabilir.



Şekil 5.14 : Regresyon ağaçları modelinin harita 3 verileri ile üç boyutlu görseli.

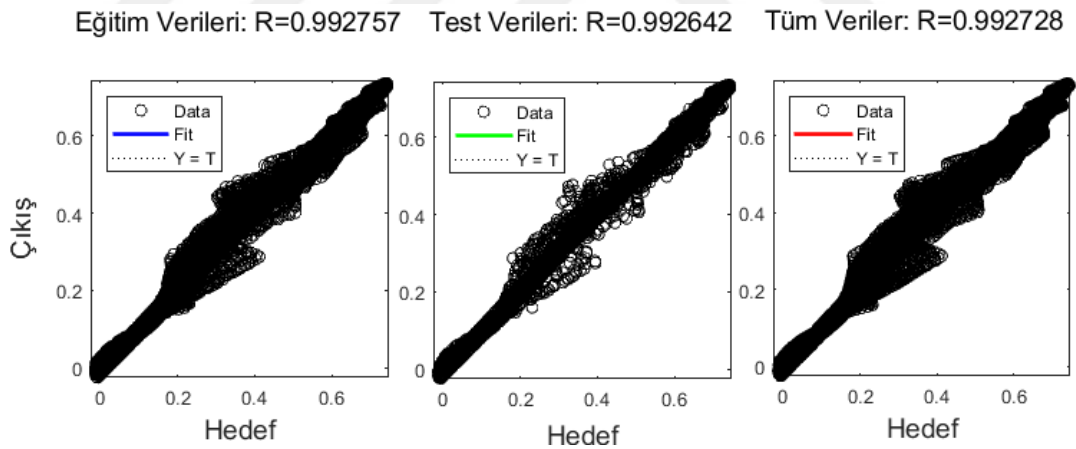
5.4 Destek Vektör Makineleri Yönteminin Sonuçları

Harita 3 için destek vektör makineleri yöntemi kullanılarak elde edilen değerlendirme metriklerinin sonuçları çizelge 5.8 'de yer almaktadır.

Çizelge 5.8 : Harita 3 için destek vektör makineleri yönteminin sonuçları.

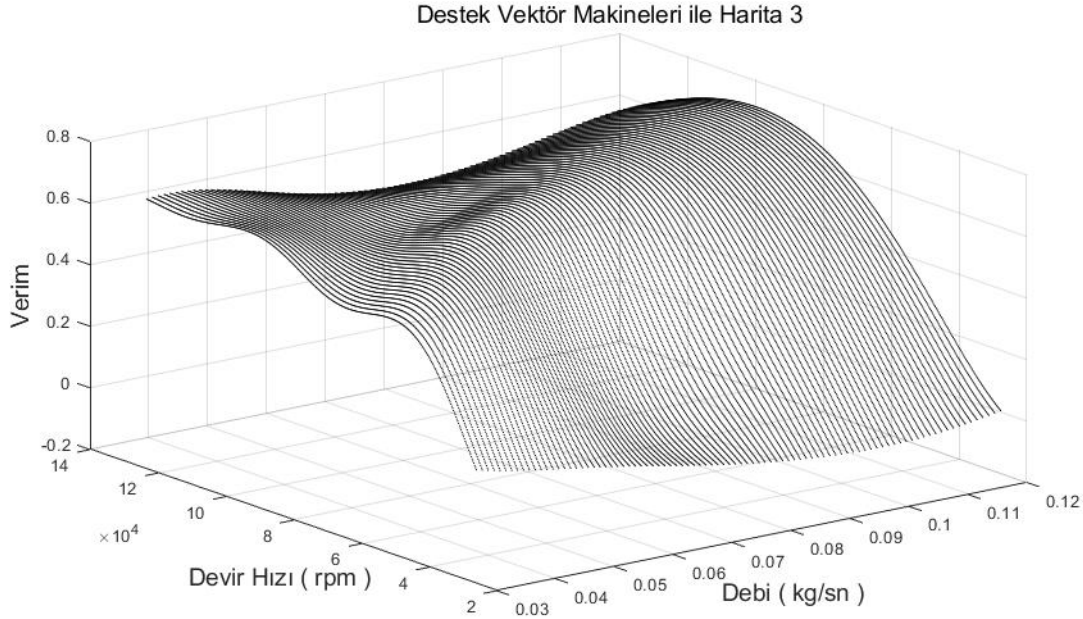
Harita 3	Eğitim Verisi	Test Verisi	Tüm Veriler
SVM	(26325)	(8775)	(35100)
MAE	0.01185943	0.011961839	0.011885036
MAPE	0.02354683	-0.101651892	-0.00775284
MSE	2.9607×10^{-4}	3.047×10^{-4}	2.9823×10^{-4}
RMSE	0.01720692	0.017455681	0.017269449

Yukarıdaki değerlendirme metriklerine göre bu yöntemin harita 3'ü regresyon ağaçları yönteminden daha kötü öğrendiği görülüyor. Ancak bu değerlendirmelere ek olarak ve yöntemin doğruluğunu öğrenmek için sıklıkla kullanılan R2 skoruna da şekil 5.15'den bakarak daha doğru bir yorum yapabiliriz.



Şekil 5.15 : Harita 3 için destek vektör makineleri yönteminin R2 skorları.

Harita 3 için destek vektör makineleri yönteminin sonucunu görsel olarak da şekil 5.16'da görebiliriz. R2 skorları ve diğer 4 performans ölçütü bakımından şimdiye kadar uygulanmış olan polinom regresyonu, karar ağaçları ve destek vektör makineleri yöntemlerini kıyaslayacak olursak en yüksek skoru karar ağaçları daha sonra destek vektör makinelerinin verdiğini görmekteyiz. Ancak her bir yöntemin kendi içinde avantaj ve dezavantajlar barındırdığını göz önünde bulundurmakta fayda var.



Şekil 5.16 : Destek vektör makinelerinin harita 3 verileri ile üç boyutlu görseli.

5.5 Yapay Sinir Ağları Yönteminin Sonuçları

Bu bölümden önce makine öğrenmesi yöntemlerinden olan farklı dereceden polinomal regresyon, destek vektör makineleri ve karar ağaçları yöntemlerinin sonuçları çizelgeler ve şekiller olarak ortaya koyuldu. Tüm bu değerleri ve sonuçları göz önünde bulundurarak aynı veriler ile eğitilen yapay sinir ağlarının sonuçlarını diğer yöntemlerle kıyaslayabiliriz.

5.5.1 Yapay sinir ağları ile harita 1 için sonuçlar

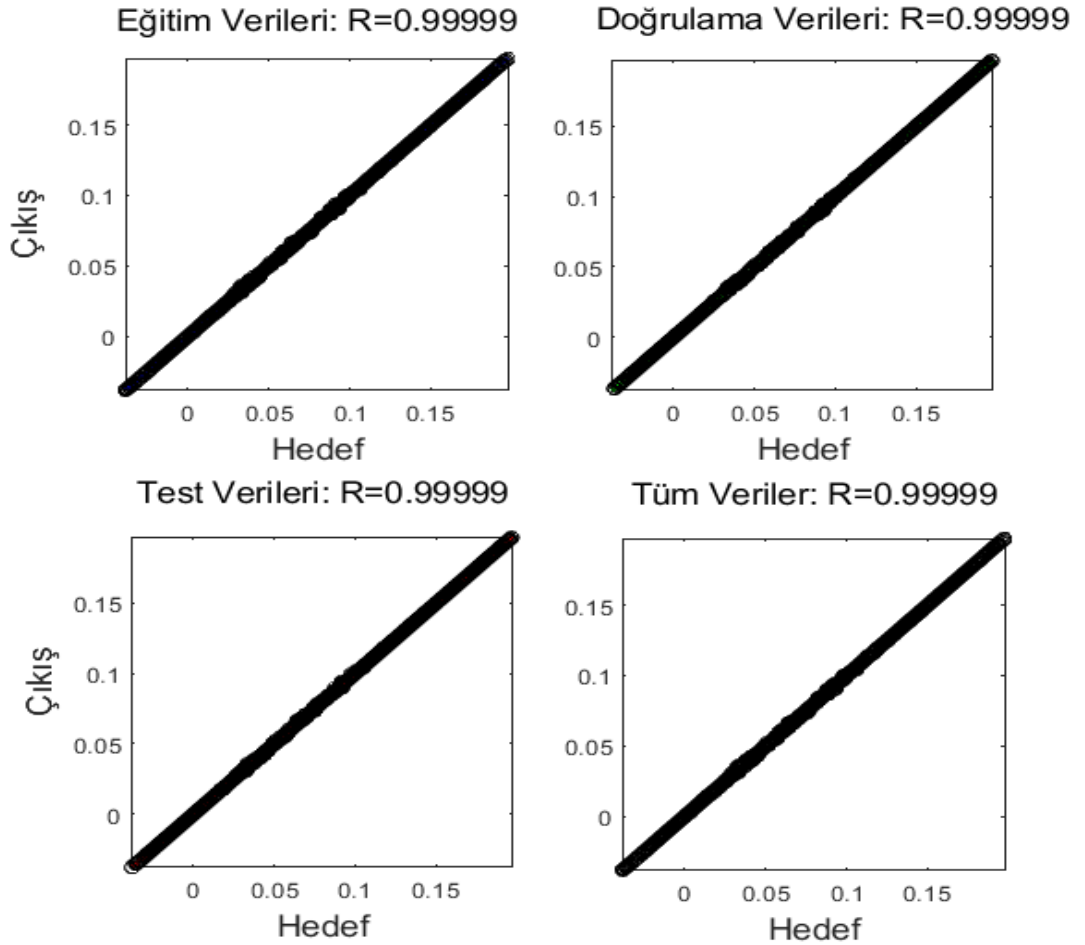
Harita 1 için çizelge 5.9, şekil 5.17 ve şekil 5.18 incelenerek derin öğrenme yönteminin sonuçlarını ikinci ve altıncı dereceden polinomal regresyonun sonuçlarıyla kıyaslayabiliriz.

Çizelge 5.9 : Harita 1 için yapay sinir ağları modelinin sonuçları.

Harita 1	Eğitim Verisi	Doğrulama	Test Verisi	Tüm Veriler
Y.S.A.	(39283)	Verisi (7857)	(14480)	(61620)
MAE	9.80×10^{-5}	9.39×10^{-5}	9.68×10^{-5}	9.79×10^{-5}
MAPE	0.00204854	9.131×10^{-4}	0.00205894	0.001906218
MSE	4.48×10^{-8}	4.69×10^{-8}	4.41×10^{-8}	4.49×10^{-8}
RMSE	2.11×10^{-4}	2.16×10^{-4}	2.10×10^{-4}	2.12×10^{-4}

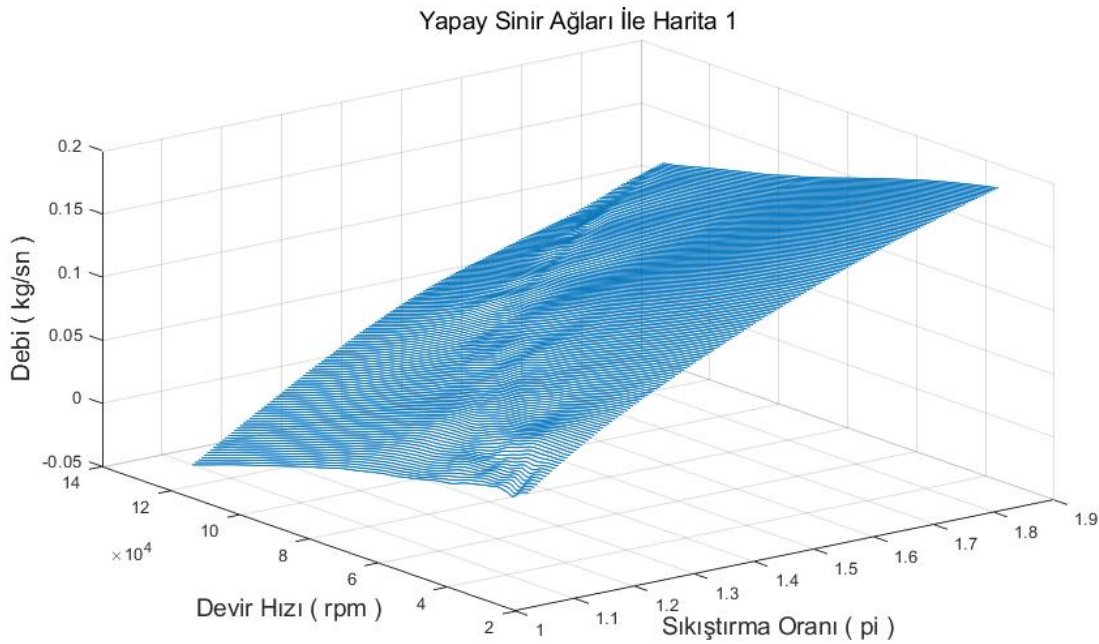
Bu bölümden önceki çizelgelerde dahil olmak üzere tüm çizelgelerde eğitim verisi, test verisi ve tüm veriler kullanarak elde edilen sütunlar yer almaktadır. Yapay sinir ağlarının sonuçlarının yer aldığı çizelgelere bakacak olursak ise fazladan bir sütunun daha yer aldığını görebiliriz. Bilindiği gibi eğitimin ne zaman ezberlemeye geçtiğini veya en iyi sonuca ulaşip ulaşmadığının kontrolü için verilerin bir kısmını validasyon verisi olarak ayrılmaktadır. Eğitim verisi olarak değil de doğrulama verisi olarak kullanılan bu veri kümesi de tekrar eğitim sonucunda ağa verilerek elde edilen sonuçları doğrulama verisi adıyla eklenen bu yeni sütunda görebiliriz.

Çizelge 5.9’da en doğru kıyaslamayı eğitim sırasında hiç kullanılmadığı için test verileri üzerinden yapabiliriz. 4 farklı metrik için test verisi sonuçlarını ele alır ve polinomal regresyon yöntemlerinin çizelgelerinde yer alan sonuçlar ile kıyaslarsak rahatlıkla görebiliriz ki yapay sinir ağlarının ürettiği sonuçlar daha başarılıdır. Ek bilgi vermesi açısından eğitim verileri de kıyaslanabilir fakat eğitim verilerinin sonuçları ezberleme durumunda da iyi sonuç verdiği için güvenilirmez olabilir.



Şekil 5.17 : Harita 1 için yapay sinir ağları ile elde edilen R^2 skorları.

Ağın öğrenmesini en iyi gösteren yöntemlerden birisi de R2 skorları olduğundan bahsedilmişti. Yine çizelgelerde yer alan doğrulama verisi sütunu gibi yapay sinir ağları için R2 skorlarının olduğu tüm şekillerde doğrulama verilerinin sonuçlarını temsil eden fazladan bir grafik daha görmekteyiz. Bilindiği gibi R2 skoru grafiklerinde yatay eksen gerçek sonuçları dikey eksen de ağın sonuçlarını göstermektedir. Grafik ne kadar doğrusal ve R2 skoru 1'e yakın ise sonuç o kadar başarılı olmaktadır. Bu kriterleri göz önüne alarak harita 1 için yapay sinir ağlarının R2 skorları ile ikinci ve altıncı dereceden polinomal regresyonun sonuçlarını kıyaslarsak harita 1 için yapay sinir ağlarının daha başarılı olduğunu görebiliriz.



Şekil 5.18 : Harita 1'in tüm verileri ile Y.S.A modelinin çıktısı.

Gerçek sonuçların kullanılarak oluşturulduğu 3 boyutlu grafik göz önüne alınarak, Harita 1'in tüm verileri kullanılarak elde edilen şekil 5.18 ile harita 1 için ikinci ve altıncı dereceden polinomal regresyonun 3 boyutlu grafikleri kıyaslanırsa yapay sinir ağlarının sonuçlarının gerçek haritaya daha çok benzediği görsel olarak görülmektedir.

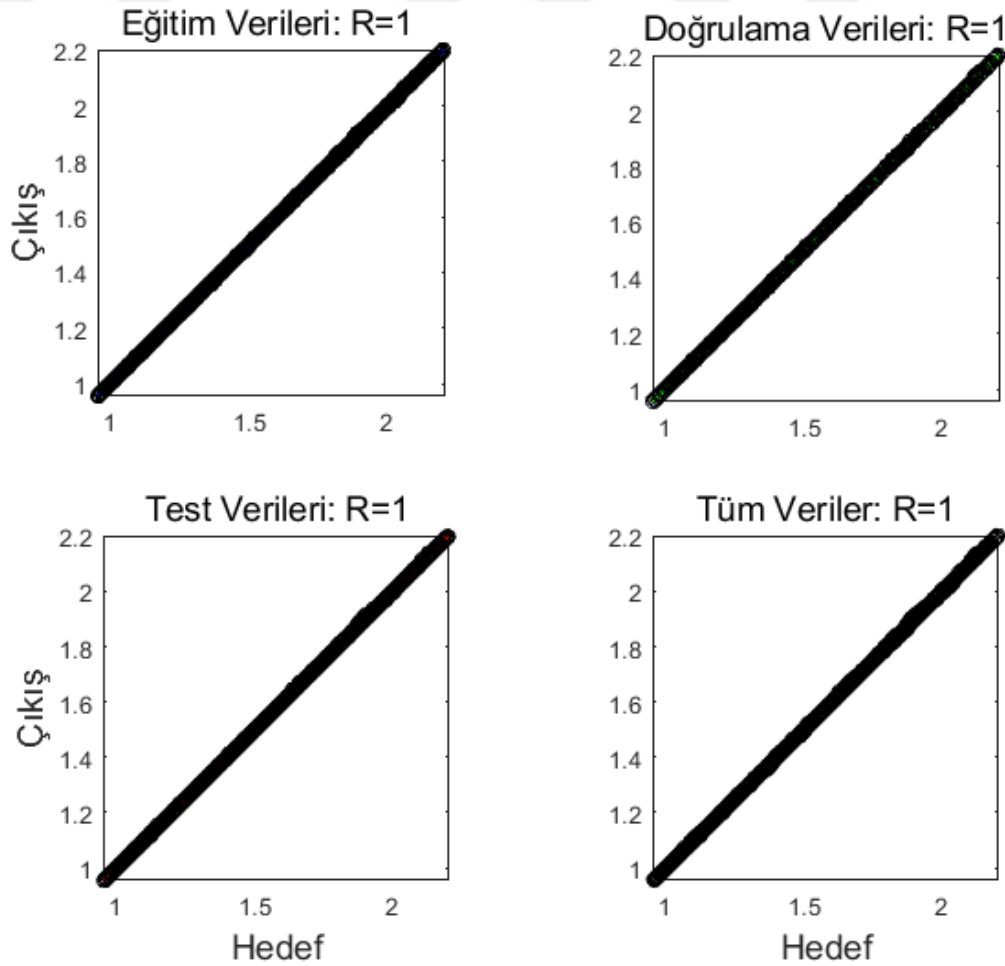
5.5.2 Yapay sinir ağları ile harita 2 için sonuçlar

Harita 2 için çizelge 5.10, şekil 5.19 ve şekil 5.20 incelenerek derin öğrenme yönteminin sonuçlarını ikinci ve altıncı dereceden polinomal regresyonun sonuçlarıyla kıyaslayabiliriz.

Çizelge 5.10 : Harita 2 için yapay sinir ağları modelinin sonuçları.

Harita 2	Eğitim Verisi	Doğrulama	Test Verisi	Tüm Veriler
Y.S.A.	(17901)	Verisi (5968)	(11231)	(35100)
MAE	4.62×10^{-4}	4.54×10^{-4}	4.76×10^{-4}	4.65×10^{-4}
MAPE	3.31×10^{-4}	3.27×10^{-4}	3.39×10^{-4}	3.33×10^{-4}
MSE	6.99×10^{-7}	6.60×10^{-7}	7.69×10^{-7}	7.15×10^{-7}
RMSE	8.36×10^{-4}	8.13×10^{-4}	8.77×10^{-4}	8.45×10^{-4}

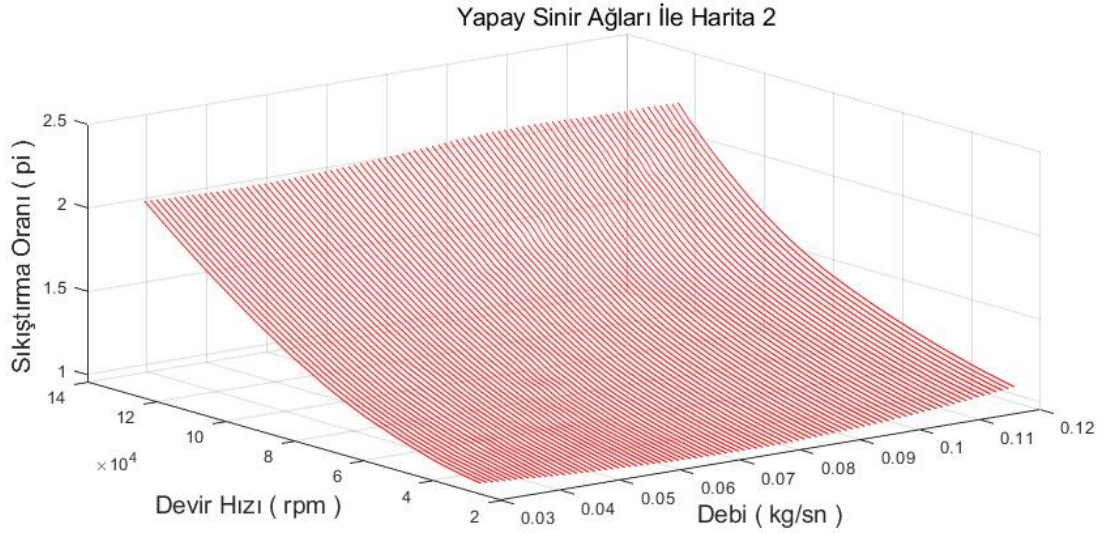
Çizelge 5.10’da yer alan 4 farklı metrik için test verisi sonuçlarını ele alır ve harita 2 için polinomal regresyon yöntemlerinin çizelgelerinde yer alan sonuçlar ile kıyaslırsak yine aynı şekilde görebiliriz ki yapay sinir ağlarının ürettiği sonuçlar daha başarılıdır.



Şekil 5.19 : Harita 2 için yapay sinir ağları ile elde edilen R2 skorları.

Harita 2 için yapay sinir ağlarının R2 skorları ile polinom regresyonu yönteminin sonuçlarını kıyaslırsak yapay sinir ağlarının daha başarılı olduğunu görebiliriz.

Harita 2 giriş çıkış ilişkisi bakımından diğerlerine göre daha basit bir yapıya sahip olduğu için R2 skorunun 1 olduğunu görebiliyoruz. Tabi ki yüzde yüz öğrenme mümkün değil virgülden sonra yedinci basamaktan sonra yuvarlama yapıldığı için tam olarak 1 sonucunu görebiliyoruz. Harita 1 ve harita 2 için kullanılan yapay sinir ağı modelleri aynıydı fakat veri kümesinin öğrenilme durumuna göre sonuçların farklı olduğunu da görebiliyoruz.



Şekil 5.20 : Harita 2'in tüm verileri ile Y.S.A modelinin çıktısı.

Harita 2'nin gerçek sonuçların kullanılarak oluşturulduğu 3 boyutlu grafik göz önüne alınarak, Harita 2'in tüm verileri kullanılarak elde edilen şekil 5.20 ile harita 2 için ikinci ve altıncı dereceden polinomal regresyonun 3 boyutlu grafikleri kıyaslanırsa yapay sinir ağlarının sonuçlarının gerçek haritaya daha çok benzediği görsel olarak görülmektedir. Harita 2 basit bir yapıya sahip olduğu için özellikle başlangıç noktaları, bitiş noktaları ve tepe değerler bize farkı ayırt etmemiz için yol gösterici olmaktadır.

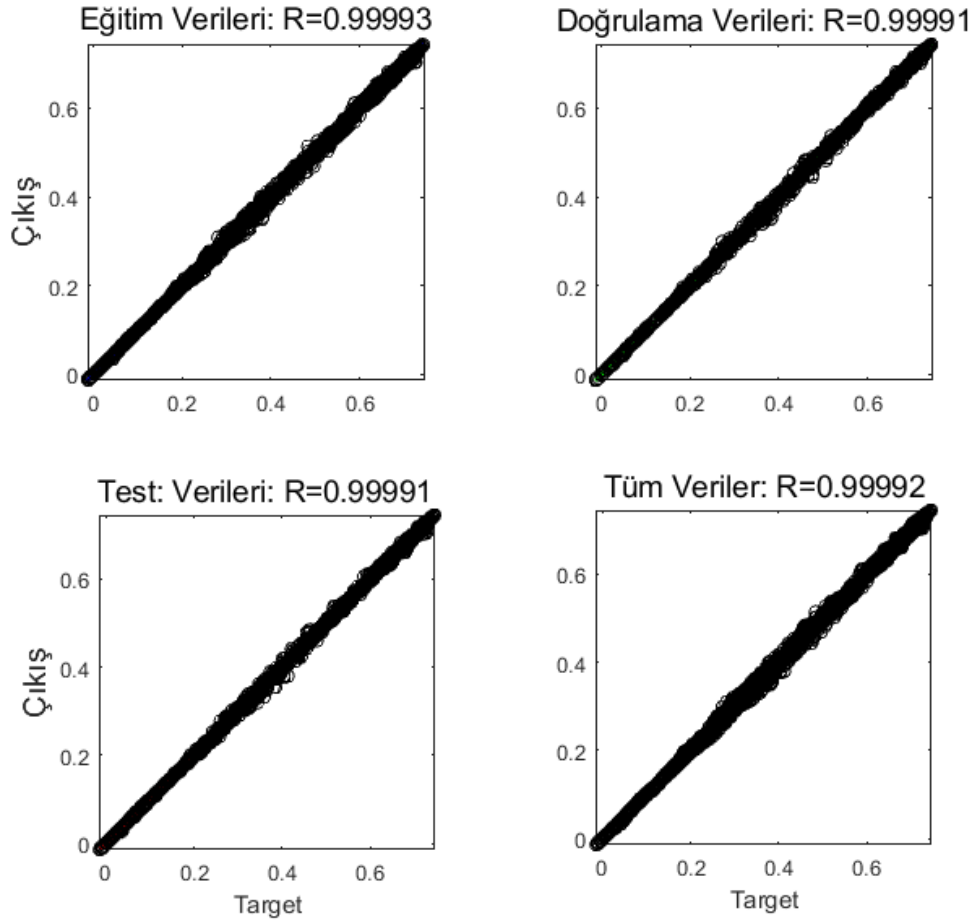
5.5.3 Yapay sinir ağları ile harita 3 için sonuçlar

Harita 3 makine öğrenmesi yöntemlerinden sadece polinomal regresyon değil destek vektör makineleri ve karar ağaçları yöntemleri ile de eğitilip test edilmiştir. Dolayısıyla yapay sinir ağlarının sonuçlarını bu diğer 3 yöntemle kıyaslayarak en iyi sonucu veren yöntem dizel motorun modellerine eklenecektir. Harita 3 için çizelge 5.11, şekil 5.21 ve şekil 5.22 incelenerek derin öğrenme yönteminin sonuçlarını diğer makine öğrenmesi yöntemlerinin sonuçlarıyla kıyaslayabiliriz. Yine odak noktamız test verilerinin sonuçları olmalıdır.

Çizelge 5.11 : Harita 3 için yapay sinir ağı modeli sonuçları.

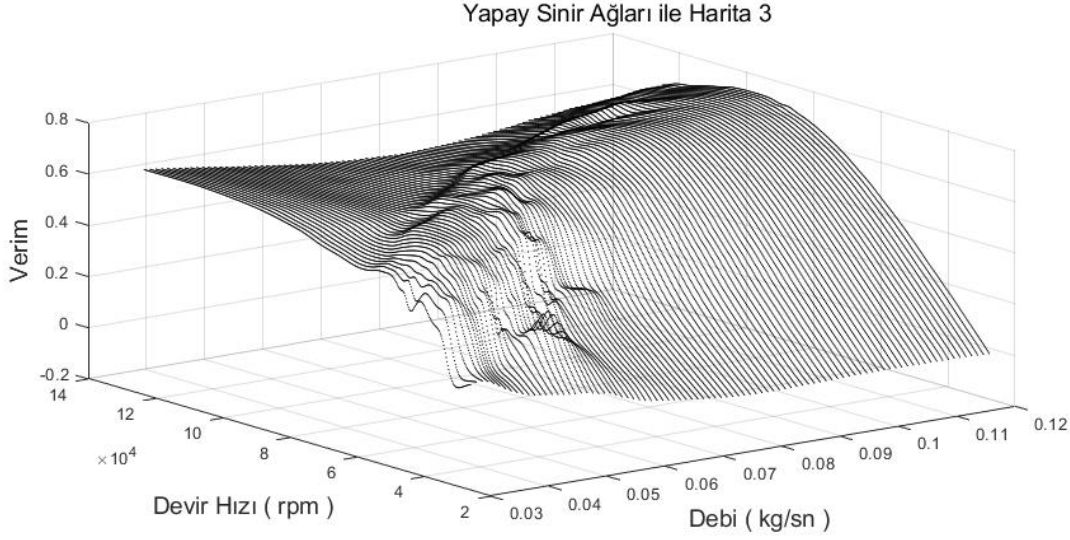
Harita 3	Eğitim Verisi	Doğrulama	Test Verisi	Tüm Veriler
Y.S.A.	(18954)	Verisi (6318)	(9828)	(35100)
MAE	0.00135618	0.00130551	0.00137990	0.00135370
MAPE	0.00238876	0.00314766	0.00108586	0.00216061
MSE	6.53×10^{-6}	6.21×10^{-6}	7.22×10^{-6}	6.67×10^{-6}
RMSE	0.00255578	0.00249333	0.00268851	0.00258266

Çizelge 5.11’de yer alan 4 farklı metrik için test verisi sonuçlarını ele alır ve harita 3 için diğer 3 makine öğrenmesi yöntemlerinin çizelgelerinde yer alan sonuçlar ile kıyaslırsak yine aynı şekilde görebiliriz ki yapay sinir ağlarının ürettiği sonuçlar daha başarılıdır.



Şekil 5.21 : Harita 3 için yapay sinir ağı ile elde edilen R2 skorları.

Harita 3 için yapay sinir ağlarının R2 skorları ile harita 3 için diğer makine öğrenmesi yöntemlerinin sonuçlarını kıyaslarsak yapay sinir ağlarının daha başarılı olduğunu hem grafikten hem de R2 skorlarının bariz farklılıklarından rahatça görebiliriz.



Şekil 5.22 : Harita 3'in tüm verileri ile Y.S.A modelinin çıktısı.

Son olarak harita 3'ün gerçek sonuçların kullanılarak oluşturulduğu 3 boyutlu grafik göz önüne alınarak, Harita 3'in tüm verileri kullanılarak elde edilen şekil 5.22 ile harita 3 için diğer makine öğrenmesi yöntemlerinin ürettiği 3 boyutlu grafikleri kıyaslanırsa yapay sinir ağlarının sonuçlarının gerçek haritaya daha çok benzediği görsel olarak görülmektedir. Özellikle kıvrımlı noktalardan bu fark kolayca fark edilmektedir.

Tüm bu sonuçların özeti olarak açıklama yapılacak olursa dizel motorun turbo-kompresör sisteminin makine öğrenmesi ve derin öğrenme tekniklerini kullanarak modellenmesinde farklı yöntemlerle başarılı sonuçlar alınmıştır. Yapay sinir ağları eğitilerek elde edilen modellerin (yapay sinir ağları için bahsedilen parametrelerin önemi de dikkate alınarak) diğer yöntemlere göre daha başarılı olduğu sonucu ortaya koyulmuştur. Ancak yapay sinir ağları eğitilirken özellikle çok katmanlı yapay sinir ağları için eğitim verilerinin sayısının çok olması gerekmektedir. Eğer eğitim sayısı yeterli düzeyde olmazsa ağın öğrenmesi zorlaşacak ve bu durumlarda az eğitim verisine ihtiyaç duyan diğer makine öğrenmesi yöntemlerini kullanmak daha uygun olacaktır.

KAYNAKLAR

- [1] **Morita S.** (1993) Optimization control for combustion parameters of petrol engines using neural networks: in the case of online control, *Int J Vehicle Des* 14(5/6):552–62.
- [2] **Howlett R. J. Howson P. A. Walters S. D. and Pashley N.** (1996) Determination of air fuel ratio in an automotive ignition system using neural networks, *International symposium on automotive technology and applications*, Florence, Italy.
- [3] **Muller R. Hemberger H. H. and Baier K.** (1997) Engine control using neural networks: a new method in engine management systems, *Mechanica*; 32(5):423–30.
- [4] **Ortmann S. and Glesner M.** (1998) Development and implementation of a neural knock detector using constructive learning methods, *Int J Uncertainty Fuzziness Knowledge-Based Syst*; 6(2):127–37.
- [5] **Heister F. and Froehlich M.** (2001) Non-linear time series analysis of combustion pressure data for neural network training with the concept of mutual information. *Proc Inst Mech Engrs, Part D: J Automobile Engng*; 215(D2):299–304.
- [6] **Yongsheng He, Christopher J. Rutland** (2002) *Artificial Neural Networks*, " SAE Technical Paper 2002-01-2772, 2002.
- [7] **Ka In Wong, Pak Kin Wong & Chun Shun Cheung** (2015) Modelling and Prediction of DieselEngine Performance using Relevance Vector Machine, *International Journal of Green Energy*, 12:3, 265-271
- [8] **N. Acharya and S. Panda** (2017) An Artificial Neural Network Model for a Diesel Engine Fuelled with Mahua Biodiesel, *Computational Intelligence in Data Mining* pp 193-201.
- [9] **Mingxin Yu, X. Tang, Y. Lin and X. Wang** (2018) Diesel engine modeling based on recurrent neural networks for a hardware-in-the-loop simulation system of diesel generator sets, *Neurocomputing* Volume 283 P 9-19.
- [10] **Url-1** <<http://elektromot.com/tag/dort-zamanli-motor/>>, erişim tarihi 12.02.2020
- [11] **Url-3** <<https://docplayer.biz.tr/13950494-Turbosarj-uzmani-holset.html>>, erişim tarihi 12.02.2020.
- [12] **Url-4** <https://dieselnet.com/tech/engine_egr_sys.php>, erişim tarihi 12.02.2020.

[13] **Url-5** <<https://www.conceptsnrec.com/blog/performance-corrections-for-compressor-maps>>, erişim tarihi 12.02.2020



ÖZGEÇMİŞ

Ad-Soyad : Yusuf CAN
Doğum Tarihi ve Yeri : İzmir – 01.03.1992
E-posta : cany17@itu.edu.tr

ÖĞRENİM DURUMU

- **Lisans** : 2016, Marmara Üniversitesi, Teknoloji Fakültesi, Mekatronik Mühendisliği
- **Çift Anadal** : 2018, Marmara Üniversitesi, Teknoloji Fakültesi, Elektrik – Elektronik Mühendisliği

MESLEKİ DENEYİM VE ÖDÜLLER

- 2016 yılı Mekatronik Mühendisliği bölüm üçüncülüğü.
- 2016 yılında 4 ay Aselsan A.Ş.’de uzun dönem staj tecrübesi.
- 2018-2019 yıllarında, Penta Elektronik A.Ş. firmasında gömülü yazılım, bilgisayarlı görü ve arayüz tasarımı konularında ar-ge projelerinde çalıştım.
- 2019 yılından beri ICTERRA Bilgi Teknolojileri şirketinde yazılım mühendisi olarak çalışmaya devam etmekteyim.

YAYINLAR

- **Veysel Gökhan Böcekçi, Yusuf Can, Nur Eser.** New Generation Microcontroller Based Ball Plate System Design. 2018. International Conference on Multidisciplinary Global Issues in Social Sciences & Business and Economics, January 22-23 / Bangkok – Thailand.
- **Halise Türkmen, Yusuf Can, Celal Güzel, Hatice Kevser Yavuz, Volkan Sezer.** 2018. Dikey İniş ve Kalkış Yapan Küçük Boyutlu İHA’lar İçin Kendini Dengeleyen Platform Tasarımı ve Kontrolü, TOK2018 Otomatik Kontrol Ulusal Toplantısı, 12-14 Eylül 2018, Kayseri.