

T.C.
İSTANBUL AYDIN ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ



DERİN ÖĞRENME YÖNTEMLERİNİN TIBBİ TEŞHİS
ALANINDA SORGULANMASI VE BEYİN TÜMÖRÜ
TANISINDA UYGULANMASI

DOKTORA TEZİ

Utku Can AYTAÇ

Bilgisayar Mühendisliği Anabilim Dalı
Bilgisayar Mühendisliği Programı

HAZİRAN, 2022

T.C
İSTANBUL AYDIN ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ



DERİN ÖĞRENME YÖNTEMLERİNİN TIBBİ TEŞHİS
ALANINDA SORGULANMASI VE BEYİN TÜMÖRÜ
TANISINDA UYGULANMASI

DOKTORA TEZİ

Utku Can AYTAÇ
(Y1615.610005)

Bilgisayar Mühendisliği Anabilim Dalı
Bilgisayar Mühendisli Programı

Tez Danışmanı: Prof. Dr. Ali GÜNEŞ

HAZİRAN, 2022

ONAY FORMU



ONUR SÖZÜ

Doktora tezi olarak sunduğum “Derin Öğrenme Yöntemlerinin Tıbbi Teşhis Alanında Sorgulanması ve Beyin Tümörü Tanısında Uygulanması” adlı çalışmanın, tezin proje safhasından sonuçlanmasına kadarki bütün süreçlerde bilimsel ahlak ve geleneklere aykırı düşecek bir yardıma başvurulmaksızın yazıldığını ve yararlandığım eserlerin Kaynakça’da gösterilenlerden oluştuğunu, bunlara atıf yapılarak yararlanılmış olduğunu belirtir ve onurumla beyan ederim. (01/06/2022)

Utku Can AYTAÇ





ÖNSÖZ

Bu çalışma, İstanbul Aydın Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı Doktora Tezi olarak hazırlanan “Derin Öğrenme Yöntemlerinin Tıbbi Teşhis Alanında Sorgulanması ve Beyin Tümörü Tanısında Uygulanması” isimli tezi içermektedir. Çalışmamın her aşamasında bilgi ve deneyimleri ile yardımcı olan danışmanım Sayın Prof. Dr. Ali Güneş'e teşekkür ederim. Çalışmam boyunca destekleri için Prof. Dr. Naim Ajlouni'ye de teşekkür ederim. Son olarak yoğun iş tempom sırasında tez çalışmamı tamamladığım için bana anlayış ve sabır gösteren, yöneticim Sayın Gönenç Seçil Tarakcıoğlu'na ve bana her zaman destek olan canım annem ve babama sonsuz teşekkür ederim.

Haziran, 2022

Utku Can AYTAÇ



DERİN ÖĞRENME YÖNTEMLERİNİN TIBBİ TEŞHİS ALANINDA SORGULANMASI VE BEYİN TÜMÖRÜ TANISINDA UYGULANMASI

ÖZET

Derin öğrenme yöntemlerindeki ilerlemeler, tıbbi görüntüleme ile elde edilen muazzam görüntü verisi yardımıyla insan organlarını kapsamlı bir şekilde araştırmaya ve hastalıkların erken teşhisine yardımcı olmaktadır. Tıbbi görüntü verisi sağlama teknikleri ve donanımları hızla gelişirken, bu veriyi görüntü sınıflandırma ve segmentasyonunda sıkça kullanılan evrişimsel sinir ağları (CNN'ler) ile işleme ve analiz etme, önemli bir araştırma problemi olarak ortaya çıkmaktadır. Bu çalışmada tıbbi teşhis alanındaki derin öğrenme yöntemleri incelenerek, evrişimsel sinir ağları ile tıbbi resimlerdeki hastalıklı yapının teşhis edilmesi ve sınıflandırılması amaçlanmıştır. Çalışmada, evrişimsel sinir ağlarına yönelik özgün parametre optimizasyon yöntemi önerilmiştir. CNN'ler, derin sinir ağlarının eğitimi için geriye yayılım algoritmasında gradyan iniş metodunu kullanırlar. Literatürde son yıllarda, eğitim süresini hızlandırmak ve performansını iyileştirmek için momentum önerilmektedir. Geleneksel momentum yöntemleri genellikle sabit bir değer olup, doğru hiperparametre değerini bulmak hesaplama karmaşıklığı açısından oldukça belirsizdir. Önerilen uyarlanabilir momentum yöntemi, derin sinir ağlarının parametre optimizasyon sürecini hızlandırmak ve daha istikrarlı yakınsamasını amaçlamaktadır. Uyarlanabilir momentum, eğitim sürecindeki (epok) geçmiş hata fonksiyonu değerini bellekte tutarak, hata fonksiyonunun her epokta artmasına veya azalmasına bağlı değişmektedir. Önerilen yöntem, CNN'lerde uygulanmış ve Moleküler Beyin Neoplazisi Veri Havuzu (REMBRANDT) veri seti ile test edilmiştir. Yöntemin geçerliliği ise farklı tıbbi veri setleri ile yapılmıştır. Uyarlanabilir momentum yönteminin performansı, stokastik gradyan iniş (SGD), Adam ve RMSprop gibi diğer optimizasyon yöntemleri ile karşılaştırılmıştır, ayrıca önerilen uyarlanabilir momentuma sahip en başarılı derin CNN mimarileri için performans karşılaştırması yapılmıştır. Sonuçların değerlendirilmesinde doğruluk, kesinlik, duyarlılık ve F1 değerleri kullanılmıştır. Tez, altı bölümden oluşmakta

olup: Giriş, Derin Öğrenme, Materyal ve Yöntem, Bulgular, Sonuç ve Öneriler kısmından meydana gelmiştir. Giriş bölümünde, bu tezin amacı ve bu çalışma hakkında literatür araştırmasına yer verilmiştir. Derin Öğrenme bölümünde, derin sinir ağlarının genel yapısı, çalışma prensibi incelenmiştir. Materyal ve yöntem bölümünde, çalışma alanı ile ilgili bilgi, kullanılan veri seti ve aynı zamanda önerilen yöntem belirtilmiştir. Bulgular bölümünde çalışma sonuçları açıklanmıştır. Sonuç ve Öneriler bölümünde çıktılar değerlendirilmiş, görüşler ve öneriler belirtilmiştir.

Anahtar Kelimeler: Uyarlamalı Momentum Yöntemleri, Konveks Olmayan Optimizasyon, Geri Yayılım Algoritması, Evrimsel Sinir Ağları, Tıbbi Görüntü Sınıflandırması.



INVESTIGATION OF DEEP LEARNING IN MEDICAL IMAGE ANALYSIS AND DETECTION OF BRAIN TUMOR USING NOVEL ADAPTIVE MOMENTUM METHOD

ABSTRACT

Developments of deep learning have combined with large amounts of medical images allows more accurate and rapid diagnosis of disorders. It's helping doctors to diagnose patients more accurately. Deep Learning for medical diagnosis has made a tremendous impact by applying convolutional neural networks (CNNs) to medical image classification and momentum plays an essential role in stochastic gradient optimization algorithms for accelerating or improving training convolutional neural networks. In traditional optimizers in CNNs, the momentum is usually weighted by a constant. However, tuning hyperparameters for momentum can be computationally complex. In this thesis, we investigate deep learning algorithms in image prediction and propose a novel adaptive momentum for fast and stable convergence. Applying adaptive momentum rate proposes to increase or decrease based on error changes in every epoch, and it eliminates the need for momentum hyperparameter optimization. We tested proposed method with Repository of Molecular Brain Neoplasia Data (REMBRANDT) dataset and validated with different medical datasets. The proposed algorithm applies convolutional neural networks (CNNs) to multi-class magnetic resonance (MR) image classification. This method can recognize brain tumor types and identify other diseases. We compared the performance of a novel adaptive momentum optimizer with Stochastic gradient descent (SGD) and other adaptive optimizers such as Adam and RMSprop. We also investigated the performance comparison for the state of the art CNN architectures with adaptive momentum. Results are compared in terms of precision, recall and F1 score. The thesis consists of six parts: Introduction, Deep Learning, Materials and Methods, Findings, Conclusion and Suggestions. In the introduction part, the aim of this thesis and a literature review about this study were made. In the Deep Learning section, the general structure and working principle of deep artificial neural networks are

examined. In the material and method section, information about the study area, the dataset used and also the methods used are specified. In the Results section, the results of the study are explained. In the Conclusion and Suggestions section, the outputs were evaluated and opinions and suggestions were stated.

Keywords: Adaptive Momentum methods, Nonconvex Optimization, Backpropagation Algorithm, Convolutional Neural Networks, Medical Image Classificatio



İÇİNDEKİLER

ONUR SÖZÜ	iii
ÖNSÖZ.....	v
ÖZET.....	vii
ABSTRACT	ix
İÇİNDEKİLER	xi
KISALTMALAR	xiii
ÇİZELGE LİSTESİ.....	xv
ŞEKİL LİSTESİ.....	xvii
I. GİRİŞ	1
A. Tezin Amacı.....	1
B. Literatür Araştırması	2
C. Tezin Literatüre Katkıları.....	13
II. DERİN ÖĞRENME.....	17
A. Tanım	17
B. Basit Algılayıcı Modeli (Perceptron).....	19
C. Çok Katmanlı Derin Ağ Modeli.....	20
D. Derin Sinir Ağlarında Optimizasyon	22
1. Gradyan İnişi.....	23
2. Geri Yayılım Algoritması	27
3. Aktivasyon Fonksiyonları	29
4. Eniyileme (Optimizasyon) Yöntemleri	33
5. Maliyet Fonksiyonları	35
E. Evrişimsel Sinir Ağları (CNN).....	38
1. Ön Bilgi.....	38
2. Tanım	39
3. Evrişim Katmanı	42
4. Doğrusal Olmayan Katman.....	43
5. Havuzlama (Ortaklama) Katmanı	44

6. Tam Bağlantılı (FC) Katman.....	45
7. Seyrekleştirme (Dropout).....	45
8. Yığın Normalleştirme.....	45
9. Evrimsel Derin Sinir Ağı Modelleri	46
III. MATERYAL VE YÖNTEM.....	49
A. Materyal	49
1. Beyin Tümörü Tanısına Genel Bakış	49
2. Veri Seti.....	50
B. Yöntem.....	52
1. Veri Ön İşlemi	52
2. Önerilen Yöntem	54
3. Derin Sinir Ağı Mimarisi ve Eğitimi.....	56
4. Tıbbi Teşhis Performans Metrikleri	58
IV. BULGULAR	61
V. SONUÇ VE ÖNERİLER.....	69
VI. KAYNAKÇA	73
EKLER.....	89
ÖZGEÇMİŞ.....	101

KISALTMALAR

BN	: Batch Normalization (Toplu Normalleştirme)
BOA	: Bayesian Optimization Algorithm (Bayes Optimizasyon Algoritması)
BP	: Back Propagation (Geri Yayılım)
BT	: Bilgisayarlı Tomografi
CNN	: Convolutional Neural Network (Evrişimsel Sinir Ağı)
CPU	: Central Processing Unit (Merkezi İşlem Birimi)
CSAE	: Classification Supervised Autoencoder (Sınıflandırma Denetimli Otomatik Kodlayıcı)
ÇDVM	: Çekirdek Destek Vektör Makinaları
DBN	: Deep Belief Networks (Derin İnanç Ağları)
DICOM	: Digital Imaging and Communications in Medicine (Dijital Görüntüleme ve İletişim)
DL	: Deep Learning (Derin Öğrenme)
DNN	: Deep Neural Network (Derin Sinir Ağı)
DVM	: Destek Vektör Makinaları
DWT	: Discrete Wavelet Transform (Ayrık Dalgacık Dönüşümü)
FC	: Fully Connected (Tam Bağlantılı Katman)
FP-ANN	: Feed Forward Backpropagation Artificial Neural Network (İleri Beslemeli Geri Yayılımlı Yapay Sinir Ağı)
GAN	: Generative Adversarial Network (Çekişmeli Üretici Ağ)
GPU	: Graphics Processing Unit (Grafik İşleme Ünitesi)
GSEM	: Gri Seviye Eş Oluşum Matrisi
K-NN	: K-Nearest Neighbors (K En Yakın Komşu)
LIDC-IDRI	: Lung Image Database Consortium image collection (Akciğer Görüntü Veritabanı Konsorsiyumu Görüntü Koleksiyonu)
LSTM	: Long Short Term Memory (Uzun Kısa Süreli Bellek)
MAE	: Mean Absolute Error (Ortalama Mutlak Hata)
MHSI	: Medical Hyperspectral Image (Tıbbi Hiperspektral Görüntü)

MR	: Magnetic Resonance (Manyetik Rezonans)
MSE	: Mean Squared Error (Ortalama Hataların Karesi)
PACS	: Picture Archiving and Communion Systems (Görüntü Saklama ve İletişim Sistemleri)
PCA	: Principal Component Analysis (Temel Bileşenler Analizi)
PNN	: Probabilistic Neural Network (Olasılıksal Sinir Ağı)
PSO	: Parçacık Sürü Optimizasyonu
RBM	: Restricted Boltzmann Machine (Kısıtlı Boltzmann Makinası)
RNN	: Recurrent Neural Network (Tekrarlayan Sinir Ağı)
ROI	: Region of Interest
SAE	: Sparse Autoencoders (Seyrek Otomatik Kodlayıcılar)
SGD	: Stochastic Gradient Descent (Stokastik Gradyan İniş)
SVM	: Support Vector Machine (Destek Vektör Makinası)
TCIA	: The Cancer Imaging Archive (Kanser Görüntüleme Arşivi)
TPU	: Tensor Processing Unit (Tensör İşleme Birimi)
UED	: Uniform Experimental Design
US	: Ultrasonografi
VAE	: Variational Autoencoder (Varyasyonlu Otomatik Kodlayıcı)
WHO	: World Health Organization (Dünya Sağlık Örgütü)
XOR	: Exclusive OR
YİÖ	: Yerel İkili Örüntü
YSA	: Yapay Sinir Ağları

ÇİZELGE LİSTESİ

Çizelge 1: Veri Seti Rakamları	53
Çizelge 2: Önerilen Derin Sinir Ağı Mimarisi.....	57
Çizelge 3: Performans Metrikleri.....	58
Çizelge 4: Diğer Yöntemler Başarı Oranı Karşılaştırması.....	61
Çizelge 5: Diğer Yöntemlerle Kayıp Değeri Karşılaştırması	62
Çizelge 6: Önerilen Yöntemin Sınıflandırma Sonuçları	62
Çizelge 7: Aktarım Öğrenimi Model Karşılaştırması	63
Çizelge 8: Göğüs X-Ray Veri Seti Başarı Oranı Karşılaştırması	65
Çizelge 9: Covid-19 Veri Seti Başarı Oranı Karşılaştırması	66
Çizelge 10: Diğer Veri Setleriyle Validasyon Sınıflandırma Sonuçları	66
Çizelge 11: Aynı Veri Seti Üzerinde Yapılan Diğer Çalışmalar ile Karşılaştırma....	67
Çizelge 12: Aktarım Öğrenimi Model Karşılaştırması	67



ŞEKİL LİSTESİ

Şekil 1: Biyolojik Sinir Sistemi Yapısının Yapay Sinir Ağlarına Benzerliği (towardsdatascience.com)	18
Şekil 2: Basit Algılayıcı Model (Perceptron).....	19
Şekil 3: Çok Katmanlı Derin Ağ Modeli	21
Şekil 4: Derin Ağ Modelinin Tek Katmanı Örneği.....	21
Şekil 5: Gradyan İnişi Örneği (ibm.com).....	23
Şekil 6: Yerel Minimum ve Eyer Noktası Örneği (ibm.com).....	26
Şekil 7: Sigmoid Aktivasyon Fonksiyonu.....	30
Şekil 8: Tanh Aktivasyon Fonksiyonu.....	31
Şekil 9: Relu Aktivasyon Fonksiyonu.....	31
Şekil 10: Sızıntı Relu Aktivasyon Fonksiyonu	32
Şekil 11: Çok Sınıflı SVM için Maliyet Hesaplanması	36
Şekil 12: Çapraz Entropi Maliyet Hesabı: Softmax Aktivasyon Fonksiyonu Örneği	37
Şekil 13: Bilgisayar Görüntü Örneği	39
Şekil 14: Çapraz Korelasyon İşlemi.....	40
Şekil 15: Evrişim İşlemi.....	40
Şekil 16: Özellik Haritası.....	42
Şekil 17: Doğrusal Olmayan Katman: Relu Örneği.....	44
Şekil 18: Havuzlama Katmanı: Maksimum ve Ortalama	44
Şekil 19: Tam Bağlantılı Katmanlarda Seyrekleştirme Tekniği	45
Şekil 20: XCEPTION Mimarisi: (Westphal & Seitz, 2021).....	46
Şekil 21: RESNET Mimarisi (blog.devgenius.io)	47
Şekil 22: VGG16 Mimarisi (Pandiyen et al., 2019).....	48
Şekil 23: Beyin Tümörü MR Görüntüleri	49
Şekil 24: Veri seti ile ilgili üst veri (metaveri).....	51
Şekil 25: Veri Seti Örneği.....	52
Şekil 26: Ön İşlem Yapılmış Resim Örneği.....	52
Şekil 27: Derin Ağ Mimarisi (CNN).....	57
Şekil 28: Covid-19 Görüntüleri (BT).....	64

Şekil 29: Göğüs Röntgen Görüntüleri (X-Ray).....	65
Şekil 30: Model (İnce Ayarlı) Doğruluk Grafiği	68
Şekil 31: Model (İnce Ayarlı) Hata Grafiği	68



I. GİRİŞ

A. Tezin Amacı

Derin sinir ağıları, büyük miktarda tıbbi görüntü ile birleşerek, hastalıkların daha doğru ve hızlı teşhisine olanak tanımakta, hastalıkların erken teşhisinin zorluğu ise, tıp dünyasında yeni derin öğrenme tekniklerinin kullanımının önemini arttırmaktadır. Son yıllarda grafik işleme birimleri (GPU) veya tensör işleme birimleri (TPU) gibi hızlı hesaplama teknikleri, araştırmacıların daha fazla katmandan oluşan daha karmaşık yapılar tasarlayabilmesine imkân vermesiyle birlikte; çok katmanlı ağ yapısını en iyi ve optimum şekilde tasarlayarak derin öğrenme modellerinin tıbbi görüntüleri analiz etmesine olanak sağladı. Önceki yıllarda, yapay sinir ağlarında öznitelik kümelerinin oluşturulması ve bu öznitelikler arasında temsil kabiliyeti en yüksek olanların seçilmesi üzerine araştırmalar yapılmaktaydı. Derin öğrenme yaklaşımlarının gelişimi ile çok katmanlı yapay sinir ağının nasıl tasarlanacağı; kaç katmandan oluşacağı, ne kadar nöron içereceği, hangi aktivasyon fonksiyonunun ve eniyileme algoritmasının kullanılacağı, hiperparametre optimizasyonunun nasıl yapılacağı problem çözümünde daha önemli hale geldi.

Derin öğrenme, temelde bir optimizasyon problemidir. Doğrusal olmayan problemlerin çözümünde optimum değeri bulmak için öğrenme sürecinde optimizasyon yöntemleri kullanılmaktadır. Derin sinir ağlarını optimize etmek için ise geriye yayılım algoritması kullanılmaktadır. Görüntü işlemede sıklıkla kullanılan ve bir derin öğrenme yöntemi olan evrimsel sinir ağlarında optimizasyon süresini hızlandırmak veya iyileştirmek için momentum kullanımı önemli bir rol oynamaktadır. Geleneksel derin sinir ağlarında, momentum genellikle bir sabit değer olarak belirlenir. Bununla birlikte, momentum için hiperparametreleri ayarlamak, hesaplama açısından karmaşık olabilir. Bu tezde, özgün adaptif momentum algoritması önerilmektedir. Araştırma çıktıları, farklı derin öğrenme mimarileri ve optimizasyon yöntemleri ile karşılaştırılarak önemli sonuçlar elde edilmiştir. Önerilen yöntem, sınıflandırma hatasını azaltarak yakınsama performansını iyileştirmekte ve diğer adaptif optimizasyon yöntemlerine kıyasla daha yüksek başarı

oranı elde etmektedir. Deneysel sonuçlar, önerilen yöntemin tıbbi görüntü sınıflandırmasında önemli performans iyileştirmeleri sağladığını göstermektedir. Yöntemin validasyonu diğer tıbbi veri setleri ile de doğrulandığından dolayı, çalışma çıktıları sadece farklı beyin tümörlerini sınıflandırmak için değil, diğer hastalıklar için de tahminleme analizlerinde doktorlar ve diğer sağlık personelleri tarafından kullanılabilir.

B. Literatür Araştırması

Tıbbi görüntüleme ile elde edilen muazzam görüntü verisi yardımıyla insan organlarını kapsamlı bir şekilde araştırmaya yardımcı olmaktadır. Tıbbi görüntü verisi sağlama teknikleri ve donanımları süratli bir şekilde gelişirken, bu veriyi bilgisayar destekli tanı koyma metodolojileriyle işleme ve analiz etme, önemli bir araştırma problemi olarak ortaya çıkmaktadır.

Tıbbi görüntüleri bir bilgisayarla taramak ve bilgisayarla yüklemek mümkün olduğunda, araştırmacılar otomatik analiz için sistemler kurmuşlardır. İlk olarak, 1970'lerden 1990'lara kadar tıbbi görüntü analizi, düşük seviyeli piksel işleme (kenar ve çizgi dedektörü filtreleri, bölge büyütme) ve matematiksel modellemelerin (uygun çizgiler, daireler ve elipsler) uygulanmasıyla, kural tabanlı bir yapı oluşturmak için çalışmalar yapılmıştır. Aynı dönemde yapay zekâda popüler olan birçok “eğer-ise-değilse” ifadesine sahip uzman sistemlerle bir benzetme vardır. Bu uzman sistemler eski moda yapay zekâ olarak tanımlanmıştır ve genellikle basit, kural tabanlı görüntü işleme sistemlerine benzemektedir. (Haugeland, 1985)

1990'ların sonunda, eğitim verilerinin bir sistem geliştirmek için kullanıldığı denetimli teknikler, tıbbi görüntü analizinde giderek artan bir şekilde popüler hale gelmiştir. Bir sonraki adım, giriş verilerini dönüştüren birçok katmandan oluşan modellerin (ağlar -örnek görüntüler) çıktılarına ait özellikleri temsil eden, örneğin hastalığın mevcut olduğu ya da olmadığı ile ilgili daha yüksek seviye özellikleri öğrenmesidir. Bu kavram, birçok derin öğrenme algoritmasının temelinde yatmaktadır. Genel anlamda, bu sistemler normal görüntüler ile anormal olanları birbirinden ayırmak maksadıyla ön işleme, öznetelik çıkarımı ve sınıflandırma adımlarına sahiptirler. Literatürde, bu problem üzerinde klasik makine öğrenme algoritmalarının uygulanması için çok sayıda yöntem tavsiye edilmiştir (Chaplot at al, 2006; El-Dahshan at al, 2010; Mallat, 1989; Moritz at al, 2000). Yapılan ilk

çalışmalar, bağımsız içerik analizi (ICA), (Moritz et al, 2000), dalgacık dönüşümü (Chaplot et al, 2006; El-Dahshan et al, 2010; Mallat, 1989), destek vektör makinası (SVM) (Chaplot et al, 2006) ve temel bileşen analizi (PCA) (El- Dahshan et al, 2010) tabanlı hibrit çözümler sunmuştur. El-Dahshan ve arkadaşları (2010), görüntü özelliklerini, MR görüntüsünden ayırık dalgacık dönüşümü (DWT) yöntemi ile almışlardır. Ardından PCA, özellik azaltmak için kullanılmış, ayrıca ileri beslemeli geri yayımlı yapay sinir ağı (FP-ANN) ve k-en yakın komşu (k-NN) tabanlı sınıflayıcılar normal ve anormal beyin MR görüntülerini sınıflandırmak için önerilmiştir. Literatürde yapılan bu ilk çalışmalar, tıbbi görüntüleri bilgisayar tabanlı işlemenin önemini ortaya koymaları bakımından çok değerlidirler.

Zamanla ileri makine öğrenimi tekniklerinden faydalanmak açısından daha modern çözümler geliştirilmiştir. Zhang ve arkadaşları, beyin MR görüntüleri için, çekirdek (SVM) destek vektör makinesini parçacık sürü optimizasyonu (PSO) ile bileştirerek bir sınıflayıcı geliştirmişlerdir (Zhang et al., 2013). Sonradan ise, çekirdek destek vektör makinasının (SVM) yerine dalgacık entropisi yöntemini kullanmışlardır (Zhang et al., 2015). Saritha ve arkadaşları, olasılıksal sinir ağında (PNN) dalgacık entropisi yöntemini kullanmıştır (Saritha et al., 2013). Wang ve arkadaşları, parçacık sürü optimizasyonu, yapay arı kolonisi ve ileri beslemeli sinir ağlarını kullanmıştır (Wang et al., 2015a). Zhang ve arkadaşları, Shannon ve Tsallis, ayırık dalgacık dönüşümü (DWT) katsayılarıyla beyin MR görüntüsünü sınıflandırma çalışmaları gerçekleştirmiştir (Zhang et al., 2015). Benzer şekilde, Nayak ve arkadaşları. Özellik çıkarımı için iki boytlı DWT kullanmışlardır. Ayrıca özellik azaltma için olasılıksal temel bileşen analizi ve rassal orman sınıflayıcısı ile AdaBoost algoritmasını kullanmışlardır (Nayak et al., 2016). Araştırmacılar, aynı zamanda modelin doğruluğunu yükseltmek amacıyla destek vektör makinesi (SVM) ile Adaboost optimizasyon yöntemini önermişlerdir. (Nayak et al., 2017). Gudigar ve arkadaşları. Dalgacık, culvelent ve shearlet dönüşümlerini karşılaştırmıştır. Çalışmasında, radyal temelli fonksiyon çekirdekli PSO-SVM sınıflayıcı kullanarak beyin anomalilerini tespit etmişlerdir (Gudigar et al., 2019).

Derin Sinir Ağları (DNN), biyolojik sistemde beyinde bulunan nöronlardan esinlenilerek oluşturulmuş basit işlemcilerden oluşur. Nöronlar, sinyalleri bir nörondan diğer nörona doğru geçiren ağırlıklandırılmış bağlantılar ile birbirine bağlıdır. Sinir hücreleri arasındaki iletişim sinapslar yardımıyla gerçekleşir. Yapay

sinir ağı, sinir hücrelerine gelen bilgileri toplayıp, belirlenen aktivasyon fonksiyonundan geçirerek çıktıyı üretir. Üretilen çıktı bağlantılar üzerinden diğer hücrelere gönderilir. Derin Sinir Ağları (DNN), birçok farklı uygulama alanında yaygın olarak kullanılmaktadır. Literatürde, çok sayıda çalışma beyin hastalığı sınıflandırılmasını ve beyin dokusunun ve anatomik yapıların (örneğin hipokampus) bölümlenmesini ele almaktadır. Diğer önemli uygulama alanları lezyonların tespiti ve segmentasyonu olmuştur (örneğin, tümörler, beyaz cevher lezyonları, mikro kanamalar). Beyin görüntüleri 3D olsa da araştırmaların çoğu 2D'de yapılmakta ve 3D hacimleri parça parça analiz etmektedir. Bu durum makina hesaplama gücü gereksinimleri ile yakından ilgilidir. Daha yeni yayınlarda 3D ağların da kullanıldığı görülmektedir. Bahsedilen yöntemlerin çoğu MR görüntüleri üzerinde yoğunlaşmaktadır. İleriki yıllarda ultrasonografi (US) ve Bilgisayarlı tomografi (BT) gibi diğer görüntüleme yöntemlerinin de derin öğrenme temelli analizlerden yararlanabileceği düşünülmektedir.

Diğer uygulama alanlarına göre yapılan çalışmalara bakıldığında, göz hastalıklarında Oftalmik görüntüleme, son yıllarda hızla gelişmiş olsa da derin öğrenme algoritmaları bu alanda yeni uygulanmaktadır. Anatomik yapıların segmentasyonu, retina anormalliklerinin segmentasyonu ve tespiti, göz hastalıklarının teşhisi ve görüntü kalitesi değerlendirmesi gibi alanlarda yazılan çoğu makale CNN'leri kullanmaktadır (Şahiner et al., 1996). Göğüs hastalıkları tespit ve lokasyon yöntemleri ise en eski DNN uygulama alanlarından biridir. Çoğu göğüs hastalıkları tekniği iki boyutlu olduğundan, doğal görüntülerde başarılı olan yöntemler kolayca aktarılabilir. Literatürde yapılan araştırmalara genellikle göğüs kanserinin tespitini oluşturmakla birlikte kitle benzeri lezyonların tespiti ve sınıflandırılması, mikro kireçlenmelerin tespiti ve sınıflandırılması ve göğüs kanseri risk skorlaması gibi alt başlıklarda incelenmektedir. Göğüs kanseri ile ilgili yapılan çalışmalarda görülen en büyük eksiklik, yeterli veri setinin bulunmamasıdır. Yapılan araştırmada, birçok makalede farklı sonuçlar veren küçük veri setleri kullanılmıştır. Bazı çalışmalar yarı-denetimli öğrenmeyi (Sun et al., 2016), zayıf denetimli öğrenmeyi (Hwang & Kim, 2016) ve transfer öğrenimini yöntemini önermiştir (Kooi et al., 2017; Samala et al., 2016).

Literatürde kalp hastalıkları analizinde de birçok derin öğrenme uygulamaları gerçekleştirilmiştir. Sol kalp karıncığı segmentasyonu en yaygın araştırma alanı

olduğu görülmüştür. Diğer uygulama alanları segmentasyon, sınıflandırma, görüntü kalitesi değerlendirmesi, otomatik kalsiyum skorlaması ve koroner merkez çizgisi takibidir. Çoğu makalede iki boyutlu CNN'ler kullanılsa da (Wolterink et al., 2016) üç boyutlu CNN'lerin de kullanıldığı bazı makaleler vardır. (Poudel et al., 2016) U-net mimarisi içinde sol ventrikül dilimini dilimle bölümlere ayırmak ve bir sonraki dilimi ayırırken önceki dilimlerden hangi bilgileri hatırlayacağını öğrenmek için CNN'leri RNN'lerle birleştirerek tekrarlayan bir bağlantı kurmuştur. (Kong et al., 2016), zamansal regresyon analizi yaparak kalp ritmini tanımlamak için 2D CNN ve LSTM içeren bir mimari kullanmıştır. Bu alanda yapılan birçok makale halka açık verileri kullanmaktadır.

Literatürde karaciğer, böbrek, mesane ve pankreas gibi organları lokalize etmeyi amaçlayan da çok çeşitli çalışmalar mevcuttur. Genellikle organlar için BT resimleri kullanılsa da prostat gibi bazı organların analizi için MRG kullanılmaktadır. Yapılan araştırmalarda CNN kullanıldığı görülse de U-net'e benzer bir 3D fCNN (Yu et al., 2017) kullanılan makaleler de vardır. Bu çalışmada, U-net'te kullanılan birleştirme işlemi yerine bir toplama işleminin kullanıldığı, ResNet ile U-net mimarisi arasında hibrid bir yaklaşım vardır. Bu model iç organ segmentasyonunda en başarılı model olmuştur.

Tıbbi görüntülerde segmentasyon, organların ve diğer yapıların hacim ve şekil ile ilgili klinik parametrelerin analizine olanak sağlamaktadır. Segmentasyonun görevi tipik olarak, görüntü çizgilerini (konturu) tanımlamaktır. Segmentasyon, tıbbi görüntülemeye derin öğrenmeyi uygulayan en yaygın çalışma konularından birisidir ve bu nedenle, CNN tabanlı segmentasyon mimarilerinin geliştirilmesi ve daha geniş RNN uygulamaları dâhil olmak üzere, bu alandaki en geniş makale ve tez çeşitliliğine sahiptir. CNN mimarilerinin tıbbi görüntü analizinde en bilinenleri, (Ronneberger et al., 2015) tarafından yayınlanmış U-net'tir. U-net'teki iki ana mimari yenilik, eşit miktarda üst örnekleme ve alt örnekleme katmanlarının birleşimidir. Her ne kadar öğrenilmiş üst örnekleme katmanları önerilmiş olsa da U-net, karşıt evrişim ve dekonvolüsyon katmanları arasındaki geçiş bağlantılarını birleştirir. Verinin eğitimi kısmından bakıldığında ise tüm görüntülerin U-net tarafından işlenebileceği ve doğrudan segmentasyon yapılabileceği anlamına gelmektedir. Ayrıca, (Çiçek vd., 2016) tarafından U-net'in aynı hacimden birkaç 2D resim ile beslenmesiyle tam bir 3D segmentasyonunun elde edilebileceği gösterilmiştir. RNN'ler de tıbbi görüntü

segmentasyon için popüler hale gelmiştir. Farklı RNN mimarilerini CNN mimarileri ile birleştirerek başarılı sonuçlar elde edilmiş çalışmalar da mevcuttur (Xie et al., 2016b; Stollenga et al., 2015; Andermatt et al., 2016; Chen et al., 2016; Poudel et al., 2016).

Görüntü sınıflandırması, derin öğrenmenin tıbbi görüntü analizine büyük katkı sağladığı ilk alanlardan birisidir. Girdi olarak bir veya birden fazla görüntü, çıktı olarak tek bir teşhis değişkenine sahiptir (örneğin, hastalık mevcut veya değil). İlk yıllarda denetimsiz ön eğitim, SAE'ler ve RBM'ler gibi ağ mimarilerine odaklanıldığı görülmektedir. Bu teknikleri uygulayan ilk makale 2013 yılında yayınlanmıştır (Brosch & Tam, 2013). (Plis et al., 2014; Suk ve Shen, 2013 ve Suk et al., 2014) hastaları Manyetik Rezonans Görüntüleme (MRG) temelli Alzheimer hastalığını sınıflandırmak için derin inanç ağları (DBN) ve SAE'leri uygulamıştır. Son yıllarda ise bu alanda da CNN'lere doğru bir eğilim gözlenebilmektedir. Bu yöntemlerin uygulama alanları, beyin MRG'den retinal görüntülemeye ve dijital patolojiye, bilgisayarlı akciğer tomografisine (BT) kadar çok çeşitlidir. Tıbbi görüntü tahminlemesinde veri kümesi boyutları milyonlaca örnek resmin eğitildiği bilgisayarlı görüntüye kıyasla daha küçüktür. Bu nedenle, transfer öğrenimi de yaygın olarak kullanılmaktadır (Antony et al., 2016; Kim et al., 2016; Esteva et al., 2017; Gulshan et al., 2016). Derin sinir ağları modellerinin performans problemleriyle başa çıkmanın diğer bir yolu, transfer öğrenmeyi kullanmaktır. İşlenmiş verinin MR görüntülerinde olduğu gibi çok büyük olması durumunda, iki transfer öğrenme stratejisi uygulanabilir:

- (i) Öznelik çıkarımı için önceden eğitilmiş bir ağ kullanmak,
- (ii) Önceden eğitilmiş bir ağa verilerle ince ayar yapmak (Litjens et al., 2017).

Önceden eğitilmiş ağ kullanmak, derin öğrenme alanında en etkili yaklaşımlardan birisidir. Daha önceden geniş ölçekte bir veriyle eğitilmiş modelden gelen bilgi, yeni bir modelde kullanılabilir. Bu şu anlama gelir; transfer öğrenme daha önceki modelden öğrenilmiş temsilleri kullanır ve bu bilgiyi başka bir alana uygular. Bu durum, küçük veri setlerine sahip çalışmalar için büyük bir önem arz eder. Veri sayısı nispeten küçük olduğu zaman, model bir süre sonra “ezberleme”ye (overfit) başlar. Eğer önceki veri yeterince geniş ve genel ise öğrenilmiş öznelikler, orijinal

veri setinde bulunmayan farklı sınıfları sınıflandırmak için kullanılabilir. Transfer öğrenmenin bir diğer avantajı, çok büyük hesaplama gücüne ihtiyaç duymuyor oluşudur. Model, ağ katmanlarının önceden eğitilmiş modelden gelen değerlerini kullanır ve sadece tam bağlantılı son katmanı eğitir.

CNN kullanan yazarların makalelerinde, önceden eğitilmiş ağırları kullanmak yerine kendi ağ mimarilerini sıfırdan eğittikleri görülmektedir. (Menegola et al., 2016), önceden eğitilmiş ağırlar ile sıfırdan oluşturulmuş kendi ağ mimarilerini karşılaştıran deneyler gerçekleştirmiş ve önceden eğitilmiş ağırlara sadece ince ayar yaparak yaklaşık 1000 deri lezyonu görüntüsünden oluşan küçük bir veri setine göre oldukça iyi çalıştığını gözlemlemişlerdir. Ancak bu deneyler, genel sonuçlara varmak için çok küçük ölçeklerdir. Özetle, görüntü ve nesne üzerinde önceden eğitilmiş CNN'lerin, sınıflandırmada çok iyi sonuçlar verdiği söylenebilir.

Görüntü sınıflandırması, tıbbi görüntülerin küçük bir bölümünün iki veya daha fazla kategoriye sınıflandırılmasına odaklanmaktadır. Örneğin, göğüs BT'sinde nodül sınıflandırmasına odaklanma gibi. Sınıflandırma işleminde lezyonun olup olmadığının ve lezyonun yeri hakkındaki genel bağlamsal bilgiler doğru sınıflandırma tahmini için gereklidir. Bu iki bilgiyi öğrenmek, genel derin öğrenme mimarilerinde mümkün değildir. Birkaç yazar bu problemi çözmek için çoklu akış mimarilerini kullanmıştır. Örneğin, (Shen et al., 2015), her biri girdi olarak farklı bir ölçekte nodül bilgisi alan üç CNN mimarisi kullanmıştır. CNN'lerin ortaya çıkan özellik çıktıları daha sonra son özellik vektörünü oluşturmak için birleştirilmiştir. Farklı çözünürlüklerki cilt lezyonlarını sınıflandırmak için çok akışlı bir CNN kullanan yazar tarafından da benzer bir yaklaşım izlendiği görülmüştür (Kawahara & Hamarneh, 2016). (Gao et al., 2015), nükleer katarakt derecelendirmesinde, CNN filtrelerinin önceden eğitildiği CNN ve RNN kombinasyonlarının kullanılmasını önermiştir. Bu birleşim, görüntü boyutuna bakmaksızın tüm bağlamsal bilgilerin işlenmesini sağlamakta olup, 3D bilgilerinin ağırlıklı olarak dâhil edilmesinin de tıbbi görüntü sınıflandırma tahminlerinde iyi performans verdiği görülmektedir. Bilgisayar görüntülerinin 2 boyutlu görüntüler olmasından dolayı, bu durumlarda geliştirilen ağırlar doğrudan 3D bilgisini kullanmazlar. Literatür incelendiğinde birçok yazar, 3D'yi özel mimarilerle etkili bir şekilde bütünleştirmek için farklı yaklaşımlar kullanmışlardır. Setio ve arkadaşları, göğüs BT'de ilgilenilen noktaları nodül veya nodül olmayan olarak sınıflandırmak için çok akışlı bir CNN kullanmış, dokuz farklı

görüntü verisi çeşidini sınıflandırma çıktısını bağlı katmanlarda birleştirilmiştir (Setio et al., 2016), Neredeyse tüm çalışmalar, uçtan uca eğitilmiş CNN'lerin kullanımını tercih etse de Kısıtlanmış Boltzmann makinesi (RBM) (van Tulder & Bruijne, 2016; Zhang et al., 2016), seyrek otomatik kodlayıcılar (SAE) (Cheng et al., 2016) ve evrişimli seyrek otomatik kodlayıcılar (CSAE) (Kallenberg et al., 2016) gibi diğer mimariler ve yaklaşımlar da kullanılmaktadır. CSAE ile CNN arasındaki en büyük fark, seyrek otomatik kodlayıcılarla denetimsiz ön eğitim kullanımıdır.

Beyin tümörü tıbbi teşhis ve tanısında Yağub ve arkadaşları, çalışmasında MR görüntülerinin tümörlü ve tümörsüz olarak sınıflandırılmasında farklı yapılarda Rastgele Orman sınıflandırıcısı kullanmıştır (Yağub et al., 2014). Bu yapılardan Ağırlıklı Hızlı Yerel Rasgele Ormanlar ile %90, Ağırlıklı Lokal Rastgele Ormanlar ile %87 ve Yerel Rasgele Ormanlar ile %88 başarımlar elde etmişlerdir. (Eman et al., 2015) ise MR görüntülerinden tümör tespitinde, K-Ortalama Kümeleme ve Bulanık C-Ortalama Kümeleme tekniklerinin performansını üç farklı veri setinde karşılaştırmıştır. K- Ortalama Kümeleme tekniği ile ortalama %92,27 doğruluk başarımları, Bulanık C-Ortalama Kümeleme tekniği ile %96,66 doğruluk elde etmiştir. MR görüntülerinde tümör olabilecek ilgi alanlarına ait öznelik çıkarımında Gabor dalgacık dönüşüm özellikleri ve istatistiksel özellikler kullanılarak farklı sınıflandırıcıların başarımlarını kıyaslaması yapılmıştır (Nabizadeh & Kubat, 2015). Mousavi ve arkadaşlarının çalışmasında MR beyin görüntülerinden tümör tespit için ROI metodu kullanılmıştır. Sunulan metod morfolojik işlemler, alçak geçirgen filtreleme ve eşikleme gibi farklı görüntü işleme tekniklerinden oluşmaktadır. On farklı veri setinden ortalama %98,48 başarımlar ile tümör bölütleme işlemi gerçekleştirilmiştir (Mousavi et al., 2015). Bir diğer çalışmada, (Ahmadvand & Kabiri, 2016) tarafından dalgacık dönüşüm özellikleri kullanılmıştır. Tümörlü ve tümörsüz MR görüntülerinin sınıflandırılması için Rassal Orman sınıflandırıcısı kullanılan bu çalışmada sınıflandırma başarımlarını ortalama %90 olarak belirtilmiştir. T2 ağırlıklı MR görüntülerinden tümör tespiti için önerilen metodlar; eşikleme, hızlı fourier dönüşümü ve ileri seviye morfolojik işlemlerden oluşmaktadır. Yazarlar çalışmalarında tümör tespit başarımlarını %89,20 olarak belirtmişlerdir (Sudharani et al., 2016). (Kadam & Dhole, 2017) tarafından tümör olabilecek yapılara ait öznelikler Gri Seviye Eş oluşum Matrisi (GSEM) ile elde edilmiştir. Çekirdek Destek Vektör Makinaları (ÇDVM) sınıflandırıcısı ile de tümör olan ve tümör

olmayan yapılar sınıflandırılmıştır. Yazarlar sınıflandırma başarımının %97,50 olduğunu belirtmişlerdir. Bazı çalışmalarda üç boyutlu görüntülerden beyin tümörü tespiti otomatik olarak yapılmıştır. Histogram eşitleme yöntemi ile görüntüler ön işleminden geçirilmiş, tümör olabilecek yapılar belirlenmiştir. (Abbasi et al., 2017) çalışmasında üç farklı düzleme ait MR görüntülerinden Yerel İkili Örüntü (YİÖ) operatörü ile öznelikler çıkartılmıştır. Kullanılan veri setinde Dice Benzerlik indeksi 0.93, Jaccard Benzerlik indeksi ise 0.87'dir. Kullanılan diğer veri setinde ise Dice Benzerlik indeksi 0.90, Jaccard Benzerlik indeksi ise 0.82'dir. (Amin et al., 2017) tarafından tümörlü ve tümörsüz MR görüntülerini otomatik olarak ayırt edebilen bir metot sunulmuştur. Tümör olabilecek her doku için şekil, yapı ve yoğunluk özellikleri hesaplanmıştır. Farklı geçerlilik testleri belirlenmiş ve sınıflandırıcı olarak Destek Vektör Makinesi (DVM) kullanılmıştır. Yazarlar, çalışmasında %97,10 doğruluk, %91,90 duyarlılık ve %98 özgüllük elde etmişlerdir. (Pereira et al., 2016), CNN mimarisini kullanarak tam otomatik beyin tümörü tespiti yapan yöntem geliştirmiştir. Bu yöntemde BRATS 2013 veri tabanı kullanılmıştır. (Dong et al., 2017) derin öğrenme algoritmalarını kullanarak beyin tümörlerinin bölütlemesini gerçekleştirmişlerdir. Burada U-Net mimarisi kullanılarak veriler eğitilmiş, mevcut başarı oranları daha da yükseltilmiştir (Dong et al., 2017).

Literatür araştırmasında tıbbi görüntü analizi için en başarılı model türünün CNN olduğu görülmüştür. Yetmişli yılların sonlarından beri CNN'ler üzerinde çalışmalar yapılmakta (Fukushima, 1980) ve ilk olarak Yann LeCun ve arkadaşları tarafından LeNet el yazısı karakter tanıma sisteminde uygulanan CNN'lerin kullanımı, derin ağları verimli bir şekilde eğitmek için çeşitli yeni teknikler geliştirilene kadar ivme kazanmaktadır (LeCun et al., 1998). (Zhang et al., 2020), röntgen görüntülerinden oluşan veri seti üzerinde çoklu etiketli sınıflandırma yapmak için ortalama havuzlama yerine adaptif bir yöntem kullanarak CNN modelini geliştirmişlerdir. Röntgen görüntüleri özünde çoklu etiketli sınıflandırma problemidir. Bu çalışmada yazarlar, çoklu etiketli sınıflandırma problemini çoklu ikili problemlere bölerek uygulamış ve başarılı sonuçlar elde etmişlerdir. Li ve arkadaşlarının yaptıkları bir diğer çalışmada, ölçekten bağımsız öznelikleri yakalamak için tıbbi görüntülerden çıkarılan özellikleri farklı boyutlara ölçeklendirerek küçük veri kümesinin, zayıf ek açıklamaların ve değişen ilgi ölçeklerinin zorluğuna değinilmiştir (Li et al., 2019). Bu çalışmada, paylaşılan çekirdekler farklı ölçeklerin özelliklerinden değişmeyen

kalıpları tespit etmek için kullanılmıştır. Daha sonra da her bir evrişim kanalındaki özellik haritasından en yüksek aktivasyonları çıkarmak için top-k havuzu uygulanmıştır. (Wei et al., 2019), Medikal Hiperspektral Görüntü (Medical Hyperspectral Image-MHSI) verilerinin sınıflandırılması için etiketlenmemiş verilerden küresel temsili özelliklerin çıkarılması için uçtan uca CNN ve lokal detay bilgilerini korumak amacıyla temel CNN olmak üzere iki kanallı CNN önerisinde bulunmuşlardır. (Agrawal et al., 2019), transfer öğrenimi uygulanması için uygun CNN mimarisini test etmek üzere bir ölçüm tasarlamışlardır. Gastro-intestinal sistem görüntülerinin sınıflandırmasını yapmış ve beş farklı CNN mimarisinin performansını karşılaştırmışlardır. Yine başka bir çalışmada, doğruluğu artırmak ve test uygulamalarını azaltmak amacıyla CNN mimarisini otomatik olarak ayarlamak için bir yaklaşım sunulmuştur (Ramos et al., 2019). CNN ayarlama yaklaşımlarında GA (Genetik Algoritma), BOA (Bayes Optimizasyon Algoritması) ve NM (Nelder-Mead) olmak üzere üç optimizasyon algoritması kullanılmıştır. Son olarak da optimizasyon algoritmalarını kullanarak ince ayarlanmış bir CNN mimarisi oluşturma süreci altı farklı sınıfa ait beş bin biyomedikal vaka görüntüsü üzerinden değerlendirilmiştir. Sonuçlar her üç optimizasyon algoritmasında da anlamlı bulunmuştur. CNN'nin farklı uygulama yaklaşımlarına ek olarak, yakın tarihli bir çalışmada (Dutta et al., 2020), tıbbi görüntüler için CNN ve transfer öğrenimini değerlendirilmiş ve bunlar tanımlanmış etiketler altında sınıflandırılmıştır. Bazı araştırmacılar (Priyanka et al., 2020), tıbbi görüntü sınıflandırması için yeni modeller tasarlamak yerine CNN'nin performansını artırmaya yönelik bir Meta sezgisel optimizasyon algoritması uygulamıştır. Lin ve Jeng'in yaptıkları bir çalışmada, CNN modeli histopatolojik görüntü sınıflandırması için optimize edilmiş ve sonuçta da modelin performansının başarılı olduğu belirtilmiştir (Lin & Jeng, 2020). Araştırmacılar tek tip deneysel tasarım (uniform experimental design-UED) kullanmış ve meme kanseri histopatolojik görüntülerinin parametre optimizasyonunu gerçekleştirmişlerdir. (Papandrianos et al., 2020) kemik sintigrafisi görüntülerini sınıflandırmak için CNN kullanmış ve aynı zamanda da görüntü sınıflandırması için iyi bilinen GoogleNet, VGG16 ve ResNet50 CNN mimarilerini karşılaştırmıştır.

Tıbbi görüntü analizinde derin öğrenme uygulamaları son yıllarda oldukça artış göstermiştir (Balasooriya ve Nawarathna, 2017; Kharrat et al., 2015; Quratul et al., 2014; Cheng et al., 2015; Gurusamy ve Subramaniam, 2017; Usman ve Rajpoot,

2017; Jones et al., 2015; Sajjad et al., 2019; Afshary et al., 2018; Swati et al., 2019). Örneğin; bazı araştırmacılar (Balasooriya ve Nawarathna, 2017), CNN'leri kullanarak beyin tümörü sınıflandırmasını gerçekleştirirken, başka araştırmacılar destek vektör makinelerini (Kharrat et al., 2015), dalgacık dönüşümlerini (Gurusamy ve Subramaniam, 2017; Usman ve Rajpoot, 2017) kullanmışlardır. Tıbbi görüntü sınıflandırması için CNN modellemesinde birçok araştırmacı (Balasooriya ve Nawarathna, 2017; Gurusamy ve Subramaniam, 2017; Sajjad et al., 2019) tıbbi görüntüleri kullanarak sınıflandırma yapmıştır. Öte yandan (Sajjad et al., 2019), yaptıkları bir çalışmada CNN parametreler eğitilerek görüntüler üç beyin tümörü tipi ile ilgili beş kategoride sınıflandırılmıştır. Bir grup araştırmacı beyin tümörü sınıflandırmasında yeni bir makine öğrenme mimarisi (CapsNet) kullanmış olsa da yavaş öğrenme süreci nedeniyle tercih edilmemiştir (Afshary et al., 2018). Başka bir çalışmada da araştırmacılar, transfer öğrenimine dayalı ince ayar yaklaşımını uygulayarak 2D-CE-MR görüntülerini üç tip beyin tümörü olarak sınıflandırmıştır (Swati et al., 2019). Ayrıca Covid-19 sebebiyle günümüz dünyası her geçen gün kötüleşirken bazı araştırmacılar derin öğrenme modellerini kullanarak hastalıkların otomatik tanısına odaklanmaya devam etmektedir. Örneğin; Haque ve arkadaşları tarafından, 165 Covid-19 hastasının röntgen görüntüsü üzerinde bir CNN modeli kullanarak sınıflandırma yapılmıştır (Haque et al., 2020). Model hem pozitif Covid-19 hastalarının hem de negatif vakaların röntgen görüntüleri ile eğitilmiştir. Sonuçlar, pandemiye olumlu katkıda bulunabilecek modeli kullanarak hastalığı tespit etme konusunda umut verici olmuştur.

Yapılan literatür araştırmasında derin öğrenme algoritmalarının tıbbi görüntü analizinde uygulanmasının çeşitli zorlukları olduğu görülmüştür. Büyük eğitim veri setlerinin olmayışı genellikle bir engel olarak belirtilmektedir. Bununla birlikte, Görüntü Saklama ve İletişim Sistemlerinin (PACS) radyolojide kullanımı bu alanda yapılan araştırmalar için görüntü sayısını arttırmaktadır. Tıbbi alanda kişisel verilerin korunma zorunluluğu olsa da verilerin anonimleştirilerek daha geniş çapta kamuya açık veri setlerinin kullanıma sunulduğu da bir gerçektir: örneğin bir çalışmada 18 farklı açık kaynak veri seti ve 10^5 'den fazla görüntü eğitim verisi olarak kullanılmıştır (Esteva et al., 2017). Genellikle araştırmacılar, görüntü verileri ile ilgili derin öğrenme analizleri yapabilmek için için alanında uzmanlardan (örneğin radyolog, patologlar) bilgi talep etmektedirler. Yeterince büyük bir veri kümesini

etiketlemenin ise önemli miktarda zaman aldığı görülmektedir. Örneğin, radyolojide bölümlendirmeye yönelik derin öğrenme sistemlerini genellikle 3D olarak eğitmek için, 2 boyutlu dilimlenmiş resimlerin ek açıklamaları yapılması gerekmektedir ve bu çok zaman almaktadır. Bu nedenle, sınırlı veriden verimli bir şekilde öğrenme, tıbbi görüntü analizinde önemli bir araştırma alanı olduğu görülmüştür. (Çiçek vd., 2016) sadece 2D segmentasyonları kullanarak 3D segmentasyon için derin öğrenme sistemini eğiterek veriden anlamlı bilgiler edinmiştir. Çok örnekli veya aktif öğrenme yaklaşımlarının da bazı durumlarda fayda sağlayabildiği görülmüştür (Yan et al., 2016). Diğer yandan verilerin etiketlenmesi uzmanlar tarafından yapılsa bile, etiket gürültüsü algoritmaların geliştirilmesinde önemli bir sınırlayıcı faktör olabilmektedir, oysa bilgisayar görüntüsünde görüntülerin etiketlenmesindeki gürültü nispeten düşük olmalıdır. Bir örnek vermek gerekirse, akciğer BT'sindeki nodülleri tespit etmek için görüntü analizi algoritmalarını değerlendirmek için yaygın olarak kullanılan LIDC-IDRI veri setidir (Armato et al., 2011). Bu veri setinde pulmoner nodüller dört radyolog tarafından bağımsız olarak belirtilmiştir. Daha sonra, radyologlar sonucu birlikte değerlendirmiş ancak fikir birliği sağlanamamıştır. Bu tür veriler hakkında derin öğrenme sistemlerinin eğitimi, referans standardında gürültü ve belirsizlikle nasıl başa çıkılacağına dikkatlice değerlendirilmesini gerektirmektedir. Belirsizlik seçeneğini doğrudan kayıp fonksiyonuna dâhil etmek gibi çözümler düşünülebilir. Denetimsiz yöntemler, etiketlenmemiş veriler ile ağ eğitimine izin verdikleri için kullanışlıdır. Büyük miktarda etiketlenmemiş tıbbi veri elde etmek genellikle etiketli veri elde etmekten çok daha kolaydır ve VAE'ler ve GAN'lar gibi denetimsiz yöntemler bu öğrenme metodundan en iyi şekilde faydalanabilir.

Veri ile ilgili diğer bir zorluk, sınıf dengesizliğidir. Tıbbi görüntülemelerde, eldeki göreve bağlı olarak, anormal sınıf görüntüleri bulmak zor olabilir. Örnek olarak, meme kanseri tarama programlarının uygulanması, dünya genelinde birçok yerde kurulmuş olan geniş veritabanlarının oluşmasına yol açmıştır. Bu görüntülerin çoğu sağlıklıdır ve şüpheli lezyon içermemektedir. Bununla birlikte, mamogram görüntüsü şüpheli bir lezyon içerdiğinde, genellikle kanserli değildir ve çoğu kanserli lezyon bile hastanın ölümüne yol açmamaktadır. Bu sınıf dengesizliğini ele alma konusunda başarılı derin öğrenme sistemleri tasarlamak da önemli bir araştırma alanıdır. Mevcut literatürde kullanılan tipik bir strateji, spesifik veri artırma algoritmalarının sadece yeterince veri bulunmayan sınıfa uygulanmasıdır, örneğin yeni lezyonlar oluşturmak

için ölçekleme ve döndürme dönüşümlerinin uygulanması. (Pereira et al., 2016), beyin lezyonu segmentasyonunda sınıf dengesizliği ile baş edebilmek için veri artırma stratejilerinin kapsamlı bir değerlendirmesini yapmıştır. Tıbbi görüntü analizinde faydalı bilgiler sadece görüntülerin içinde yer almamaktadır. Doktorlar genellikle daha iyi kararlar almak için hasta geçmişi, yaşı, demografisi ve diğerleri hakkındaki farklı bilgilerden yararlanmaktadır. Bazı yazarlar bu bilgiyi derin öğrenme ağlarında basit bir şekilde birleştirmeyi araştırmışlardır (Kooi et al., 2017). Ancak, bu yazarların belirttiği gibi, elde edilen iyileştirmeler beklendiği kadar başarılı olmamıştır. Zorluklardan biri, derin sinir ağındaki binlerce görüntü özelliğinin sayısını az sayıdaki klinik bilgi ile dengelenmesidir. Doktorlar sıklıkla doğru bir tanı koymak için anatomik bilgileri kullanmaya ihtiyaç duymaktadırlar. Bununla birlikte, tıbbi görüntülemedeki birçok derin öğrenme sistemi hala anatomik bilginin genellikle ağ tarafından bilinmediği sınıflandırma sistemine dayanmaktadır. Ayrıca, tam görüntülerin ağı beslenmesi, bellek kısıtlamaları gibi problemlere yol açmaktadır. Bazı durumlarda bu, GPU veya TPU teknolojisindeki gelişmeler nedeniyle çözülebilir ancak gigapiksel boyutlu görüntülere sahip dijital patoloji için başka stratejiler geliştirilmelidir.

C. Tezin Literatüre Katkıları

Derin sinir ağlarında ortaya çıkan gelişmeler pek çok tıbbi görüntüyle birleşerek son yıllarda hastalıklara daha doğru ve hızlı teşhis konulmasına olanak sağlamaktadır. Bu durum, doktorların hastalara daha doğru teşhis koymasına ve etkili tedaviler önermesine yardımcı olmaktadır. Hastalıklarda erken tanının zor olması tıp biliminde yeni derin öğrenme tekniklerinin önemini artırmıştır. Evrişimsel sinir ağları (CNN) bilgisayarlı görüde en popüler derin öğrenme algoritmasıdır. Geleneksel ağlarla kıyaslandığında CNN'nin temel avantajı önemli özellikleri otomatik olarak algılaması ve ağ mimarisinin CNN'ye görüntülerden karmaşık özellikleri öğrenme yetisi vermesidir (Alzubaidi et al., 2021). Yeni yöntemler aynı zamanda CNN'lerin etkisini ve doğruluk payını da artırmaktadır.

Stokastik gradyan inişi (SGD), CNN'leri optimize etmek için en verimli algoritmalarından biri olarak kabul edilir (Leung & Haykin, 1991). Bu optimizasyon yöntemi, modelin parametrelerini kontrol ederek kayıp fonksiyonunu minimize eder. Optimizasyon, parametrelerin farklı parametrelere kıyasla amaç fonksiyonun

gradyanının ters yönünde güncellenmesiyle elde edilir (Yu et al., 2014). Yerel minimum, algoritmanın öğrenme katsayısı tarafından gereken adım sayısı ve boyutu ile temsil edilir. CNN optimizasyonu için gradyan iniş yönteminin yararı, geleneksel istatistiksel yöntemlerin çözemediği çok yönlü karmaşık problemleri çözme yeteneğidir. Verimli bir algoritma olmasına rağmen, yüksek sabit hata oranı ve yavaş yakınsama gibi bazı sınırlamalar performansını düşürür. Bu sınırlamalar, momentum tekniği olarak bilinen bir yöntem kullanılarak azaltılmaktadır. Bu teknik yakınsama oranını ve yakınsama hızını artırmaya yardımcı olur. Momentum tekniğinin dezavantajı, sabit momentum negatif gradyan yönünde olduğunda, ağırlığı eğimden aşağıya değil, hata yüzeyine göre düzenlemesidir (Wang et al., 2015b; Istook & Martinez, 2002; Qiu et al., 1992). Geri yayılım (BP) tekniği, SGD'nin kullandığı bu gradyanı hesaplar ve BP, sabit momentumu uyarlanabilir momentuma genişletmeye yardımcı olarak bu sorunu çözebilir. Bu, optimal yakınsama hızı elde edilirken yinelemelerle doğru momentum adaptasyonu ile elde edilebilir. Yineleme işlemi sırasında, adaptif momentum adım adım kendini güncelleyecektir. Bu uyarlanabilir davranışın bağlı olduğu tek değişken, her yinelemede bir çıktı değeri hatasının tahminidir.

Bu tezde, derin öğrenme yöntemlerinde genel hatayı azaltmak, en yüksek tahminleme doğruluğunu elde etmek ve yakınsamayı hızlandırmak için özgün adaptif momentum yöntemi sunulmuştur. Performans karşılaştırma analizinden (karışıklık matrisi) alınan sonuçlar, önerilen adaptif momentumun, geleneksel stokastik gradyan inişi ve diğer optimizasyon algoritmalarına kıyasla daha iyi sonuç verdiğini göstermektedir. Tezin literatüre katkıları şu şekilde özetlenebilir:

- Basit ikili sınıflandırma ile karşılaştırıldığında daha karmaşık ve zorlayıcı olan çok sınıflı beyin tümörü sınıflandırması için momentum ile güçlendirilmiş geri yayılım yöntemi önerilmiştir. Önerilen yöntem REMBRANDT veri seti (Clark et al., 2013) üzerinde test edilmiş ve bu veri seti ile yapılan çalışmalarda en yüksek sonucu elde etmiştir.
- Adaptif momentum optimizasyon algoritması diğer optimizasyon algoritmalarıyla karşılaştırılmış ve başarılı sonuçlar vermiştir. Aynı zamanda Adaptif momentumun, tıbbi teşhis sürecinde CNN'lerde SGD'nin performansını ve yakınsama hızını iyileştirdiği gösterilmiştir.

- Sadece derin sinir ağı mimarilerine karşı değil, aynı zamanda tıbbi görüntüleme verileri kullanılarak sıfırdan eğitilmiş derin sinir ağlarına karşı da adaptif momentum performansı araştırılmış ve literatüre katkı sunulmuştur.
- Çalışma alanı beyin tümörü ile sınırlandırılrsa da yöntemin geçerliliği Covid-19 ve diğer göğüs hastalıkları veri setleri ile de sağlanmıştır.





II. DERİN ÖĞRENME

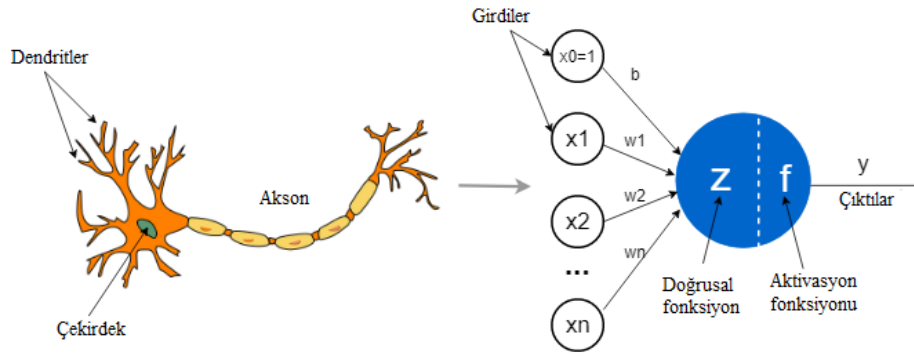
Bu bölümün amacı, bu tezde incelenen ve tıbbi görüntü analiz çalışmalarında bulunan derin öğrenme kavramlarına dair terminoloji, teknik ve mimarilerinin tanımını sağlamaktır.

A. Tanım

Bilgisayarlar, yüzyıllar boyunca insan beyninin nasıl çalıştığının merak edilmesi sonucunda ortaya çıkan ilkel hesap makinelerinin gelişmiş versiyonlarıdır. Teknoloji geliştikçe insanlar bilgisayarların işlemleri hızlı yapmak gibi bazı özellikleri bu temel düşüncenin üstünü örtmüş ve insanlar teknolojinin geliştirilmesini teknolojiden elde edecekleri yarara bağlamışlardır. Bazı araştırmacılar ise ısrarla çalışmalarını insanın davranışlarının modellenmesi yolunda devam ettirmişler ve yapay zekâ bilimi başta olmak üzere oldukça önemli gelişmeleri ortaya çıkartmışlardır. 1980’li yılların başlarına gelindiğinde aslında bilgisayar teknolojisinin günümüz ile kıyaslanamaz bir durumda olduğunu görüyoruz. O zamanlarda, insan beyninin bir benzeri yapılabilir mi? İnsan gibi davranabilen bir robot oluşturulabilir mi? Bilgisayarlar düşünebilirler mi? Olayları öğrenebilirler mi? vb. gibi soruların cevapları henüz bulunmamıştı. Aslında bu tür sorular yıllardır sürekli sorulmaktaydı. Sorulmaya da devam edilmiş ve sonuç olarak günümüzde bu soruların bazıları kısmen de olsa cevaplandırılmıştır. Hepsinin cevabı bulunamamıştır ama artık bilgisayarlar öğrenebilmektedir.

Yapay sinir ağları yapay zekâ biliminin altında araştırmacıların çok yoğun ilgi gösterdikleri bir araştırma alanıdır. Özünde, bilgisayarların öğrenmesine yönelik çalışmaları kapsamaktadır. Yapay sinir ağları, olayların örneklerine bakmakta onlardan ilgili olay hakkında genellemeler yapmakta, bilgiler toplamakta ve daha sonra hiç görmediği örnekler ile karışılışınca öğrendiği bilgileri kullanarak o örnekler hakkında karar verebilmektedir. Bundan 15-20 yıl önce böyle bir şeyin olacağını düşünmek bile mümkün değildi. O zaman bilgisayarlar öğrenebilecek denilse idi bu hayal ürünü olarak görülürdü.

Yapay sinir ağı temelde insan biyolojik sinir sisteminden etkilenecek geliştirilmiştir. İnsan beyni sinir hücrelerinden oluşmaktadır. Beyin bilginin öğrenilmesi ve saklanması hücreler arasında iletişim ile gerçekleşmektedir. Yetişkin bir insan beyinde ortalama 1 trilyon hücre olduğu tahmin edilmektedir. Sinir hücresine gelen bilgiler sinir uçlarında (dendrit) toplanır. Biyolojik sinir hücreleri sinir uçlarında (dendrit) gelen bilgiyi akson boyunca ilerleyerek diğer sinir hücrelerinin dendritlerine göndermektedirler. Bu ağlar birbirine bağlı proses elemanlarından (yapay sinir hücrelerinden) oluşur. Her bağlantının bir ağırlık değeri vardır. Şekil 1’de görüldüğü üzere, biyolojik sinir hücreleri birbirleri ile sinapsler vasıtası ile iletişim kurarlar ve sinir hücresi işlediği bilgileri aksonlar yolu ile diğer hücelere gönderir. Benzer şekilde yapay sinir hücreleri dışarıdan gelen bilgileri bir toplama fonksiyonu ile toplar ve aktivasyon fonksiyonundan geçirerek çıktıyı üretilip ağırlık bağlantılarının üzerinden diğer hücelere (proses elemanlarına) gönderir.



Şekil 1 Biyolojik Sinir Sisteminin Yapay Sinir Ağlarına Benzerliği
(towardsdatascience.com)

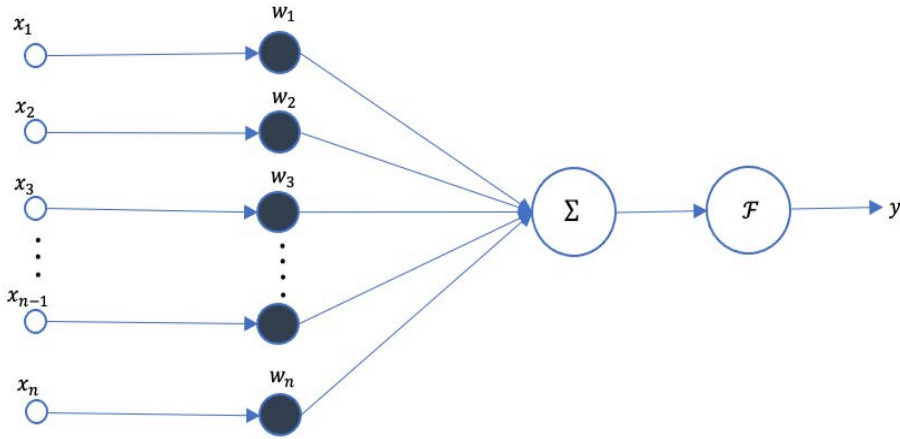
Nasıl ki sinir hücreleri, giriş sinyallerini diğer sinir hücrelerinden dendritler yoluyla alırsa; benzer şekilde, bir yapay sinir hücresi de verilerini diğer sinir hücrelerinden alır. Dendritler ve biyolojik nöronlar arasındaki bağlantı noktalarına sinaps denir. Aynı şekilde girdiler ve algılayıcılar arasındaki bağlantılara da ağırlıklar denir. Ağırlıklar, her girdinin önem düzeyini ölçerler. Biyolojik bir nöronda çekirdek, dendritler tarafından sağlanan sinyallere dayalı olarak bir çıkış sinyali üretir. Aynı şekilde, bir yapay sinir ağındaki çekirdek (mavi renkli), girdi değerlerine dayalı olarak bazı hesaplamalar yapar ve bir çıktı üretir. Biyolojik bir nörona, çıkış sinyali akson tarafından taşınır. Aynı şekilde, yapay sinir ağındaki akson, sonraki sinir ağı algılayıcıları için girdi olacak olan çıktı değerleri oluşturur (Kadiroğlu, 2019).

Yapay Sinir ağları insan beyninin en temel özelliği olan öğrenme fonksiyonunu

gerçekleştiren bilgisayar sistemleridir. Nasıl ki yeni doğan bir bebek zamanla sıcak bir cisme dokunmamayı yaşadığı örnek olaylar ile öğreniyorsa, yapay sinir ağları da öğrenme işlemini insan beyni gibi edindiği örnekler yardımı ile gerçekleştirirler. Yapay sinir ağının edindiği bilgi, ağırlık değerlerinde saklı olup ağa yayılmıştır.

B. Basit Algılayıcı Modeli (Perceptron)

1958 yılında F.Rosenblatt tarafından tanıtılan Perceptron, insan beyninin öğrenme sürecinin matematiksel modellenmesi şeklinde ifade edilebilir. Perceptron, tek bir yapay sinir hücresinden oluşan bir ağ modelidir ve bir nöronun matematiksel olarak ifade edilmesidir. Birden fazla girdinin yalnızca iki sonuç (1 veya 0) üretmesine dayanmaktadır. Şekil 2’de basit algılayıcı model örneği verilmiştir.



Şekil 2 Basit Algılayıcı Model (Perceptron)

Tek perseptron giriş, hesaplama ve çıkış katmanı olmak üzere üç parçadan oluşmaktadır. Perseptron bir veya daha fazla girdi alabilir ve hesaplama katmanındaki sonuca göre bir çıktı üretebilir. Giriş katmanında, Rosenblatt her katmanı $x_1, x_2 \dots x_i$ olarak ve her ağırlığı $w_1, w_2 \dots w_i$ olarak tanımlamıştır. Bu değerler birer reel sayı olarak tanımlanmıştır. Ağırlık, perseptron modeline belirli bir hedefe yönelik çıktı üretme yeteneği sağladığı için çok önemlidir. Perseptron, sıfır ve birlerden oluşan bir değer üretmektedir. Girdi katmanındaki değerler ile ilgili ağırlık değerlerinin çarpımının toplamına yanlılık (bias) değeri eklendiğinde, sonuç eşik değerinden daha büyük ise perseptron 1, değilse 0 olarak çıktı üretecektir. Örneğin, perseptron girdisinin toplam değerinin 0.42 ve eşik değerinin 0 olduğunu varsayılırsa, toplam değer eşik değerinden büyük olduğu için bu nöron ateşlenecek

ve çok katmanlı bir ağ yapısında kendinden sonra gelen diğer perseptronlara bilgiyi aktaracaktır. Perseptronun çıkış üreten fonksiyonu Denklem 1’de gösterilmiştir.

$$x = \sum_i w_i x_i + b$$
$$y = \begin{cases} 1, & \text{eğer } x \geq 0 \\ 0, & \text{diğer} \end{cases}$$

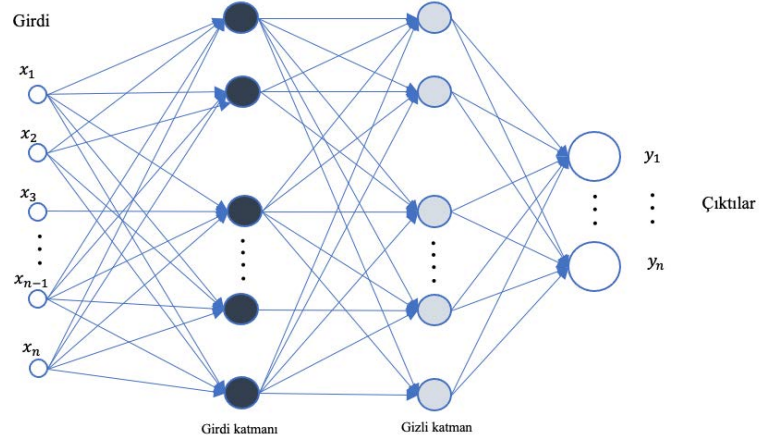
Denklem 1

Denklem 1’de çıktı y, girdi katmanı (x_i), ağırlıklar (w_i), kutuplama değeri (b) olarak gösterilmiştir.

C. Çok Katmanlı Derin Ağ Modeli

Tek katmanlı perceptron modeli XOR gibi kompleks problemlerin çözümünde yetersiz kalmaktadır. Bu nedenle çok katmanlı perceptron yapısı ya da derin sinir ağları oluşturulmuştur. Çok katmanlı ağ modeli en az bir ara katmana sahiptir. Temel olarak yapay sinir ağları 3 katmandan meydana gelir. Bu katmanlar sıralı bir yapay sinir ağı modelini oluşturur. Bunlar;

- Girdi Katmanı: Yapay sinir ağında gelen bilgiler girdi katmanında temsil edilir. Veri setine göre değişen özelliklerin her biri farklı bir düğüm olarak girdi katmanında temsil edilir. Her bir girdinin (özellik) ağırlık(weight) değeri vardır. Bu girdiler gizli katmandaki düğümler ile bu ağırlıklar aracılığı ile bağlıdır. Girdiler için belirlenen ağırlık değerleri yapay sinir ağında o özelliğin önemini, yani ağırlığını belirtmektedir.
- Gizli, Ara Katmanlar: Gizli katmanlar, girdi katmanından gelen bilgilerin işlenip bir çıktıya dönüştürüldüğü katmanlardır. Çıktıya dönüştürme işlemi yapay sinir ağının ağırlık değerleri kullanılarak gerçekleştirilir. Gizli katman sayısı problemin zorluğuna göre çeşitlilik gösterebilmektedir
- Çıktı Katmanı: Gizli katmandan gelen sonuç bilgisi hesaplanan çıktı değeri olarak belirlenir. Çıktı değeri veya değerlerinin tutulduğu katmandır. Verilen girdi değerine ait çıktıyı barındırır (Sertkaya, 2018).



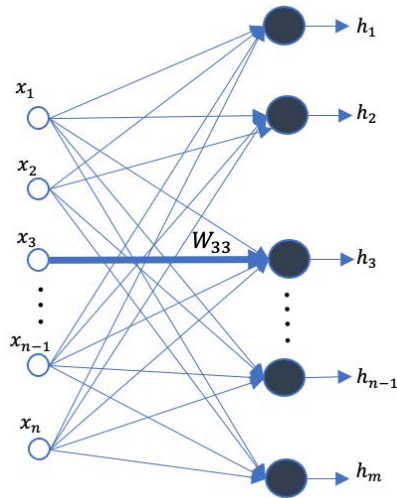
Şekil 3 Çok Katmanlı Derin Ağ Modeli

Şekil 3'teki çok katmanlı ağ modelinin her bir katmanı için girdi ve çıktı bilgileri örneği Denklem 2'de gösterilmiştir. Diğer katmanlar içinde aynı işlem tekrarlanmaktadır.

$$h_n = f\left(\sum_{i=1}^n x_i w_i + b\right)$$

$$y = f(W^T U)$$

Denklem 2



Şekil 4 Derin Ağ Modelinin Tek Katmanı Örneği

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} \quad w = \begin{bmatrix} w_{01} & \cdots & w_{0n} \\ \vdots & \ddots & \vdots \\ w_{n1} & \cdots & w_{3m} \end{bmatrix} \quad h = \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ \vdots \\ h_m \end{bmatrix}$$

Derin ağ modelinin matriks olarak gösterimi yukarıdaki gibidir. Burada x giriş

vektörü, w ağırlık matrisi ve h ise çıkış vektörüdür.

D. Derin Sinir Ağlarında Optimizasyon

Yapay sinir ağlarının başarısını etkileyen veri sayısı, katman sayısı, öğrenme oranı, gibi birçok parametre bulunmaktadır. Bu parametrelerin optimize edilmesi yapay sinir ağının başarısını olumlu bir şekilde arttıracaktır. Ancak hata hesaplama ve hata optimizasyonu fonksiyonlarının doğru seçilmesi gibi parametreler yapay sinir ağının başarısını mutlak şekilde doğrudan etkilemektedir.

Yapay sinir ağlarında optimizasyon, ağırlıkların güncellenmesi bakımından önemlidir. Hesaplanan çıktı değeri ile beklenen çıktı değeri belirlenen hata fonksiyonuna göre hesaplanır. Hesaplanan hata değeri beklenen sonuçtan ne kadar uzak olduğumuzu belirler. Hesaplanan bu değere göre seçilen “optimizasyon fonksiyonu” kullanılarak ağırlıklar güncellenir. Bu işlem aslında ağın öğrenmesi demektir. Yapay sinir ağı ağırlıklarını güncelleyerek öğrenmeye başlar. Hata optimize edilmeye başlanır. Beklenen sonuçları veren ağırlıkların bulunması sistemin eğitilmesi anlamına gelir (Kadiroğlu, 2019).

Yapay sinir ağlarında optimizasyon ağırlıkların güncellenmesi ve maliyet fonksiyonunun en aza indirgenmesi yoluyla gerçekleşir. Çok katmanlı bir ağ modeli olan Denklem 2’yi ele alırsak ağın girdisi x , çıktısının y' ve olması gereken çıktı değerinin y olduğunu varsayarsak, yapay sinir ağının parametreleri öyle bir güncellenmelidir ki $y = y'$ olsun. Böyle bir durumda bu da zaten bize hata fonksiyonunun değerini 0 verir. İleriki bölümlerde hata fonksiyonlarından daha detaylı bahsedilecektir fakat bir hata fonksiyonu örneğini ele alalım. Regresyon problemlerinde sıkça kullanılan ortalama kare hatası (MSE) oranı için N ’in örneklem sayısı olduğunu varsayarsak Denklem 3 şu şekilde ifade edilebilir:

$$MSE = \frac{1}{N} \times \sum_{j=1}^N (y_j - y'_j)^2$$

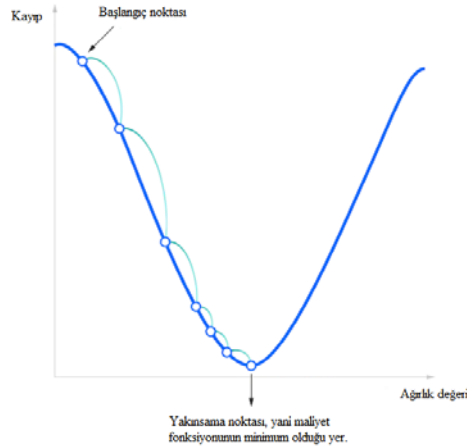
Denklem 3

Denklem 3’teki gibi bir maliyet fonksiyonunu minimize etmek için çeşitli yöntemler vardır. Newton yöntemi, fonksiyonun köklerini ikinci dereceden türev olarak bulmak için kullanılsa da birinci dereceden türevleri kullanan gradyan inişi gibi yöntemlerle

karşılaştırıldığında, ikinci derece türev almanın hesaplama karmaşıklığı daha fazladır. Bu nedenle çoğu sinir ağının eğitimi için gradyan inişi tercih edilmektedir.

1. Gradyan İnişi

Bir fonksiyonun türevi fonksiyonun arttığı yönü, negatif ise fonksiyonun azaldığı yönü göstermektedir. Derin sinir ağı modelini eğitmek maliyet fonksiyonunu azaltmaktır ve bu fonksiyonu minimize etmek için türevin negatif yönünde ilerlemek gerekmektedir. Gradyan inişi kademe boyunca aşağı gitme yani modelin parametrelerini kayıp fonksiyonundan aşağı gidecek şekilde ayarlama işlemidir. Türevi hesaplamak için son katmandan girdi verilerini alan ilk katmana zincir kuralı uygulanmaktadır. Gradyan inişte hesaplanan türevler kullanılarak parametrelere göre kayıp fonksiyonunun minimumuna ulaşılmaya çalışılır. Bunun içinde minimuma ulaşma yönünde ne kadar hareket etmek istendiğini düzenleyen bir öğrenme oranı ile çarpılan ağırlıkların türevinin, giriş ağırlıklarından çıkarılarak parametrelerin güncellenmesi gerekmektedir. Gradyan iniş algoritması Şekil 5'teki gibi konveks fonksiyon tabanında olsa da konveks olmayan fonksiyonların da çözümünde etkilidir. Gerçek dünyadaki problemlerin çözümünün çoğunun konveks değildir ve gradyan iniş algoritması bu fonksiyonlarda global minimuma ulaşmaya çalışır.



Şekil 5 Gradyan İnişi Örneği (ibm.com)

Başlangıç noktası performans değerlendirmesi için yalnızca rastgele bir noktadır. Bu başlangıç noktasından türevi (veya eğimi) bulunur ve oradan eğimin dikliğini görmek için teğet bir çizgi kullanılabilir. Eğim parametrelerdeki güncelleme bilgilerini verecektir. Başlangıç noktasında eğim daha dik olacaktır. Parametreler

güncellendikçe diklik yakınsaklık noktası olarak bilinen eğri üzerindeki en düşük noktaya ulaşınca kadar kademeli olarak azalmalıdır.

Doğrusal regresyonda en iyi uyum doğrusunu bulmaya benzer şekilde gradyan inişinin amacı, maliyet fonksiyonunu veya beklenen ile gerçek değer arasındaki hatayı minimize etmektir. Bunu yapmak için yön ve öğrenme oranı gibi iki veri noktası gereklidir. Bu faktörler yerel veya global minimum değerine (yani yakınsama noktasına) kademeli olarak giderek bir sonraki yinelemelerin kısmi türev hesaplarını belirler.

Öğrenme hızı minimuma ulaşmak için atılan adımların boyutudur. Bu tipik olarak küçük bir değerdir ve maliyet fonksiyonunun gidişatına göre değerlendirilir ve güncellenir. Yüksek öğrenme oranları daha büyük adımlarla sonuçlanır. Ama bunların global veya lokal minimuma ulaşamama riski vardır. Bunun aksine düşük bir öğrenme oranı küçük adım boylarına sahiptir. Daha fazla hassasiyet avantajına sahip olsa da minimuma ulaşmak için daha fazla zaman ve hesaplama gerektirdiğinden eğitim süresi genel verimliliği olumsuz yönde katkıda bulunmaktadır.

Maliyet (veya kayıp) fonksiyonu mevcut konumunda gerçek değer ile beklenen değer arasındaki farkı veya hatayı ölçer. Bu modele geri bildirim sağlayarak makine öğrenimi modelinin verimliliğini artırır. Böylece hatayı en aza indirmek ve yerel veya global minimumu bulmak için parametreleri ayarlayabilir. Maliyet fonksiyonu sıfıra yakın olana veya sıfıra kadar en dik iniş (veya negatif gradyan) yönünde hareket ederek sürekli olarak yinelenir. Tam bu noktada model öğrenmeyi durduracaktır. Bunlara ek olarak maliyet fonksiyonu ve kayıp fonksiyonu terimleri eş anlamlı kabul edilse de aralarında fark vardır. Kayıp fonksiyonunun bir eğitim örneğindeki hatayı ifade ederken, öte yandan maliyet fonksiyonu tüm eğitim setindeki ortalama hatayı hesaplamaktadır.

Literatürde üç tür gradyan iniş öğrenme algoritması vardır. Bunlar: toplu gradyan inişi, stokastik gradyan inişi ve mini toplu gradyan inişi (Baydar, 2018).

Toplu gradyan inişi: Modeli sadece tüm eğitim örnekleri değerlendirildikten sonra güncelleyerek bir eğitim kümesindeki her nokta için hatayı toplar. Bu süreç modelin eğitimidir. Bu toplu işlemin hesaplamada verimlilik sağlamasının yanı sıra yine de

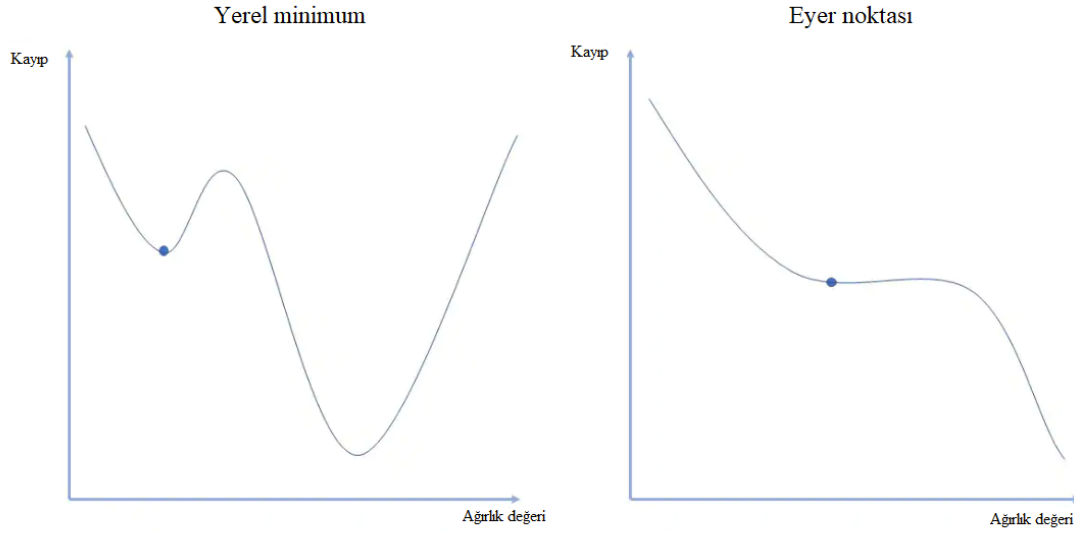
işlemin tüm verileri belleğe kaydetmesi gerektiğinden, büyük eğitim veri setleri için uzun bir işlem süresine mâl olmaktadır.

Stokastik gradyan iniş: Stokastik gradyan iniş (SGD) veri kümesi içindeki her örnek için bir eğitim süreci (epok) ve her eğitim örneğinin parametreleri için birer birer güncellemeler gerçekleştirir. Yalnızca bir eğitim örneği gerektiğinden bunları bellekte saklamak daha kolaydır. Bu sık güncellemeler daha fazla ayrıntı ve hız avantajı sunsa da toplu gradyan inişiyle karşılaştırıldığında hesaplama verimliliğinde kayıplar olabilir. Ayrıca güncellemeler gürültülü gradyanlara neden olabilmektedir. Ancak bu aynı zamanda yerel minimumdan kaçmada ve global minimumu bulmada yardımcı da olabilmektedir.

Mini toplu gradyan inişi: Mini toplu gradyan inişi, toplu gradyan inişi ve stokastik gradyan inişi kavramlarını birleştirmektedir. Eğitim veri setini küçük yığın boyutlarına böler ve bu yığınların her birinde güncelleştirmeler yapar. Bu yaklaşım toplu gradyan inişinin hesaplama verimliliği ile stokastik gradyan inişinin hızı arasında denge kurmaktadır.

Gradyan iniş optimizasyonunda bazı sorunlar bulunmaktadır.

Yerel minimum ve eyer noktaları: Dışbükey (konveks) problemlerde gradyan inişi global minimumu kolaylıkla bulabilir. Ama konveks olmayan problemler ortaya çıktıkça gradyan inişi modelin en iyi sonuçları elde ettiği global minimumu bulmakta zorlanabilir, çünkü maliyet fonksiyonunun eğimi sıfır veya sıfıra yakın olduğunda modelin öğrenmesini durdurur (ibm.com). Global minimumun ötesinde birkaç senaryo da bu eğimi verebilir. Bunlar yerel minimumlar ve eyer noktalarıdır. Yerel minimum maliyet fonksiyonunun eğiminin mevcut noktanın her iki tarafında da arttığı global minimum şeklini taklit etmektedir. Öte yandan eyer noktalarında negatif gradyan noktanın yalnızca bir tarafında bulunurken bir tarafta yerel bir maksimuma ve diğer tarafta yerel bir minimuma ulaşır. Bu isim atın eyerinden esinlenilerek belirlenmiştir. Şekil 6'da yerel minimum ve eyer noktası örneği gösterilmiştir.



Şekil 6 Yerel Minimum ve Eyer Noktası Örneği (ibm.com)

Daha derin sinir ağlarında modeller gradyan inişi ve geri yayılım algoritması ile eğitildiğinde iki problemle daha karşılaşılabilir:

Kaybolan gradyanlar: Gradyan değeri çok küçük olduğunda meydana gelir. Geri yayılım sırasında geriye hareket ettikçe gradyan küçülmeye devam eder ve ağıdaki önceki katmanların sonraki katmanlardan daha yavaş öğrenmesine neden olur. Bu olduğunda, ağırlık parametreleri önemsiz hale gelene kadar güncellenir; yani 0 olduğunda artık karşımızda öğrenmeyen bir algoritma vardır.

Patlayan gradyanlar: Gradyan çok büyük olduğunda gerçekleşir ve dengesiz bir model oluşturur. Bu durumda da model ağırlıkları çok büyüyecek ve sonunda tanımlanamayan değer olarak temsil edilecektir. Bu sorun için bir çözüm yolu olarak model içindeki karmaşıklığı en aza indirmeye yardımcı olabilecek bir boyutluluk indirgeme tekniği kullanmaktır.

Sonuç olarak gradyan inişinin derin sinir ağlarını optimize etmek için en iyi algoritmalarından biri olarak kabul edildiği söylenebilir. Bu optimizasyon yöntemi modelin parametresini kontrol ederek amaç fonksiyonunu minimuma indirir. Parametreleri farklı parametreler karşısında amaç fonksiyonunun gradyanının ters yönünde güncelleyerek kontrol sağlanır. Yerel minimum algoritmasının öğrenme hızının gerektirdiği adım sayısı ve boyu ile temsil edilir. Gradyan iniş yönteminin

CNN'e faydası, evrimsel istatistiksel yöntemlerin çözemediği çok yönlü karmaşık problemleri çözebilmesidir. Etkin bir algoritma olmasına rağmen yüksek sabit hata oranı ve yavaş yakınsama gibi bazı sınırlamalar performansını düşürür. Bu sınırlamalar momentum tekniği olarak bilinen bir yöntem kullanılarak azaltılabilir. Bu teknik en dik iniş hatasını azaltır. Ayrıca yakınsama oranını artırmaya da yardımcı olur. Momentum tekniğinin dezavantajı sabit momentum negatif gradyan yönünde olduğunda, ağırlığı eğimden aşağıya değil, hata yüzeyine göre ayarlayacak olmasıdır. Geri yayılım (BP) tekniği SGD'nin kullandığı bu gradyanı hesaplar (Zhang et al., 2020).

2. Geri Yayılım Algoritması

Araştırmacılar geniş bir Boole fonksiyon yelpazesini hesaplamak için çok katmanlı ağırları kullanırken (Leung & Haykin, 1991) hesaplama çabasında orantılı bir artış olduğunu göstermiştir. Çok sayıda parametreye sahip karmaşık topolojiler için geri yayılım algoritması tercih edilir. Geleneksel BP algoritması D. E. Rumelhart ve arkadaşları tarafından ortaya atılmıştır (Rumelhart et al., 1986). Bu çalışma sinir ağırları alanına önemli bir katkı olarak görülmektedir. Makaleye göre algoritma, gradyan iniş sürecini kullanarak ağırlık alanındaki hata fonksiyonunun minimumunu arar. Öğrenme problemleri çözümü sonuçta hata fonksiyonunun en aza indiği ağırlıkların birleştirilmesiyle elde edilir.

BP algoritması bilinen bir girdi-hedef çıktı örnekleri veri seti ile uygulanan, tahmin ve sınıflandırma uygulamalarında kullanılmakta olan denetimli bir öğrenmedir. Bir giriş katmanı, bir veya daha fazla saklı katman ve bir çıkış katmanından oluşur. Katmanlar giriş katmanından başlayarak saklı katmanlardan çıkış katmanına sırayla bağlıdır. Katmanlar arasındaki bağlantıların olduğu yerler ağırlık içerir ve her katman bir veya daha fazla nöron içerir. Bu durumda BP ardındaki temel fikir öğrenme süreci sırasında genel çıktı hatasını kademeli olarak en aza indirmektir. Eğitim setleri doğru çıktıyı kestirmek için giriş katmanı aracılığıyla yinelemeli olarak tahmin edilir. BP süreci iki aşamaya ayrılır: ileriye yayılım ve geriye yayılım. İleri yayılım sürecinde, verilen ağırlık ve yanlılık (bias) değerleriyle sinir ağının hangi değeri tahmin ettiği belirlenir. Örneğin Şekil 3 gibi bir derin sinir ağı için 2 katmanlı aktivasyon fonksiyonlu (sigmoid) ileri yayılım sırasıyla şu şekilde hesaplanır:

1. $h_1 = x_1 * w_1 + b_1$
2. $a_1 = f(h_1) = \frac{1}{1+e^{-h_1}}$
3. $h_2 = a_1 * w_2 + b_2$
4. $a_2 = f(h_2) = \frac{1}{1+e^{-h_2}}$
5. $E = \frac{1}{2}(T - a_2)^2$

Burada x_1 girdi, h_1 ve h_1 çıkış değerleri, a_1 ve a_2 aktivasyon fonksiyonu ile etkileşim haline girmiş çıktı, w_1 ve w_2 ağırlıklardır. E değeri hata fonksiyonu T değeri ise hedef istediğimiz sonuçtur. Geri yayılım sürecinde hedef, ağdaki parametreleri, gerçek çıktının hedef çıktıya yakınsaması için güncellenmesidir. Böylece hatayı en aza indirmek mümkün olacaktır. Hata fonksiyonu $E(w)$ 'yi minimize ederek global bir minimuma ulaşmak ana hedeftir. Bu durum aşağıdaki formüller ile açıklanabilir.

$$w \rightarrow -w + \eta \nabla E(w)$$

Denklem 4

$$\nabla E(w) = \left(\frac{\partial E}{\partial w_1}, \frac{\partial E}{\partial w_2}, \dots, \frac{\partial E}{\partial w_n} \right)$$

Denklem 5

E değerinin w değerine göre gradyanını bulabilmek için zincir kuralı uygulanır. w_2 örneği için;

$$\frac{\partial E}{\partial w_2} = \frac{\partial E}{\partial a_2} \cdot \frac{\partial a_2}{\partial h_2} \cdot \frac{\partial h_2}{\partial w_2}$$

- $E = \frac{1}{2}(T - a_2)^2$

$$\frac{\partial E}{\partial a_2} = 2 \cdot \frac{1}{2}(T - a_2) \cdot (-1) = -(T - a_2)$$

- $a_2 = \frac{1}{1+e^{-h_2}} = (1 + e^{-h_2})^{-1}$

$$\frac{\partial a_2}{\partial h_2} = -1 \cdot (1 + e^{-h_2})^{-2} \cdot e^{-h_2} \cdot (-1)$$

$$= \frac{e^{-h_2}}{(1 + e^{-h_2})^{-2}} = (a_2)^2 \frac{1 - a_2}{a_2} = a_2(1 - a_2)$$

$$\bullet \quad z_2 = a_1 \cdot w_2 + b_2$$

$$\frac{\partial z_2}{\partial w_2} = a_1$$

$$\frac{\partial E}{\partial w_2} = -(T - a_2) \cdot a_2(1 - a_2) \cdot a_1$$

$$\text{yeni ağırlık } (w_2) = \text{eski ağırlık } (w_2) - lr * \left(\frac{\partial E}{\partial w_2} \right)$$

Burada $lr > 0$ eğimin negatif yönü boyunca ne kadar gideceğini gösteren öğrenme katsayısıdır. Gradyan inişinde yakınsama yavaştır ve durum derin katmanlı ağlarda daha da belirgindir. Bu yöntemin ikinci zorluğu ise doğru öğrenme katsayısı seçimidir. Bu durum geleneksel geriye yayılım süreçlerinde momentum gibi farklı parametrelerin kullanılmasına yol açmıştır. Momentum terimi fonksiyonun aşağıdaki gibi güncellenmesi olarak yorumlanabilir:

$$W_{n+1} = W_n - \eta E_W(W_n) + \alpha(W_n - W_{n-1})$$

Denklem 6

Bu durumda $\Delta W_{n-1} = (W_n - W_{n-1})$ olacaktır. Yukarıdaki denklem şu şekilde yazılabilir;

$$\Delta W_n = -\eta E_W(W_n) + \alpha \Delta W_{n-1} \quad n = 0, 1, \dots \quad (9)$$

Denklem 7

α , $0 < \alpha < 1$ aralığında değer alan momentum katsayısıdır.

3. Aktivasyon Fonksiyonları

Bir sinir ağındaki aktivasyon fonksiyonu girdinin ağırlıklı toplamının ağırlık katmanındaki nod veya nodlardan bir çıktıya nasıl dönüştürüleceğini belirler. Bazen aktivasyon fonksiyonuna "transfer fonksiyonu" da denir. Birçok aktivasyon fonksiyonu doğrusal değildir ve katmanda veya ağ tasarımında "doğrusalsızlık" olarak adlandırılabilir. Aktivasyon fonksiyonunun seçilmesi sinir ağının kapasitesi ve performansı üzerinde büyük bir etkiye sahiptir ve modelin farklı bölümlerinde farklı aktivasyon fonksiyonları kullanılabilir (Wang, 2018). Ağlar bir katmandaki bütün nodlar için aynı aktivasyon fonksiyonunu kullanmak üzere tasarlanmış olsa da

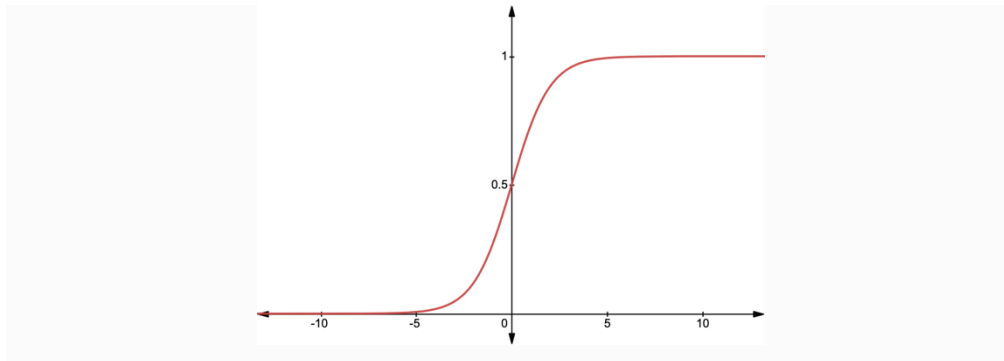
aktivasyon fonksiyonu teknik olarak ağıdaki her nodun dâhili işlenmesinde veya sonrasında kullanılır. Tüm gizli katmanlar tipik olarak aynı aktivasyon fonksiyonunu kullanır. Çıkış katmanı tipik olarak gizli katmanlardan farklı bir aktivasyon fonksiyonu kullanır ve modelin gerektirdiği tahmin türüne bağlıdır. Aktivasyon fonksiyonlarının türevi alınabilir. Yani verilen bir girdi değeri için birinci dereceden türev hesaplanabilir. Sinir ağlarının tipik olarak modelin ağırlıklarını güncellemek için tahmin hatasının türevini gerektiren hata algoritmasının geri yayılımı kullanılarak eğitildiği göz önüne alındığında bu gereklidir. Gizli katman ve çıkış katmanları için pratikte kullanılan çok az sayıda fonksiyon olsa da sinir ağlarında kullanılan birçok farklı aktivasyon fonksiyonu vardır (Baydar, 2018).

Sigmoid Aktivasyon Fonksiyonu: Girdi olarak reel sayı alır ve 0 ile 1 arasında bir değer çıkış aralığına sahiptir. Şekil 7’de görüldüğü üzere doğrusal olmayan bir fonksiyondur. Fonksiyonun matematiksel olarak ifadesi ise şu şekildedir:

$$S(z) = \frac{1}{1 + e^{-z}}$$

Denklem 8

Türevi kolay alınabildiğinden tercih edilen bir fonksiyon olmasına rağmen kaybolan gradyan problemine sahiptir.



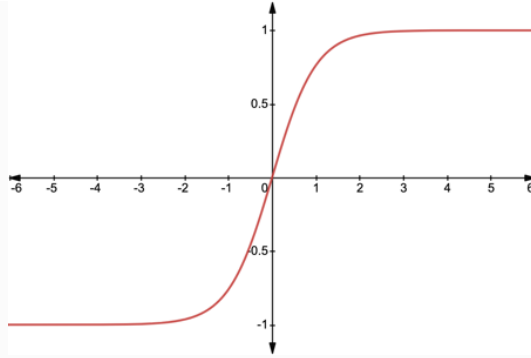
Şekil 7 Sigmoid Aktivasyon Fonksiyonu

Tanh Aktivasyon Fonksiyonu: -1 ile 1 arasında çıktı veren ve doğrusal olmayan bir fonksiyondur. Denklemin matematiksel gösterimi aşağıdaki gibidir.

$$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

Denklem 9

Şekil 8’de görüldüğü üzere, Sigmoid fonksiyonuna çok benzer fakat Tanh fonksiyonu sıfır merkezli olduğundan Sigmoid’e göre daha hızlı yakınsama özelliğine sahiptir.

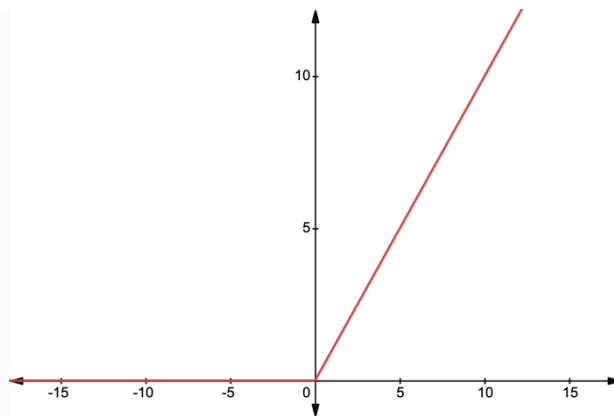


Şekil 8 Tanh Aktivasyon Fonksiyonu

Relu Aktivasyon Fonksiyonu: Kaybolan gradyan problemine karşı geliştirilmiştir. Hesaplama yükünün Sigmoid fonksiyonuna göre daha az olması derin sinir ağlarında daha çok tercih edilmesine sebep olmuştur. Fonksiyonun denklemi aşağıdaki gibidir:

$$f(z) = \begin{cases} 0, & z < 0 \\ z, & z \geq 0 \end{cases}$$

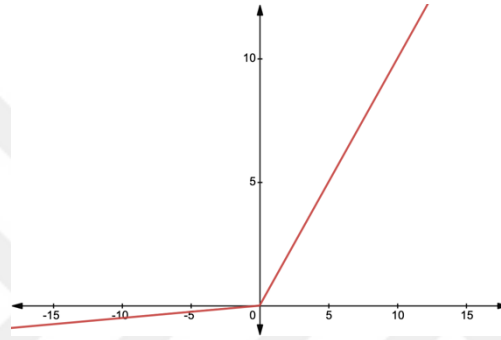
Denklem 10



Şekil 9 Relu Aktivasyon Fonksiyonu

Relu genel olarak yapay sinir ağlarında ara katmanlarda kullanılır. Her ne kadar türevi sifıra yakınsamadığı için kaybolan gradyan probleminden etkilenmese de negatif değerlerde türevin sıfır olmasından dolayı geri yayılım esnasında parametreler güncellenemez. Bu durumun üstesinden gelmek için fonksiyonun farklı varyasyonları geliştirilmiştir (Baydar, 2018).

Sızıntı Relu Aktivasyon Fonksiyonu: Fonksiyonun Relu'dan farkı negatif değerlerin sifıra çok yakın olması fakat sıfır olmamasından dolayı negatif tarafta da türevinin alınabilir olmasıdır. Böylece geri yayılım esnasında öğrenme negatif tarafta da gerçekleşmektedir.



Şekil 10 Sızıntı Relu Aktivasyon Fonksiyonu

Fonksiyonun matematiksel olarak ifadesi aşağıdaki gibidir. Burada a değeri 0'dan farklı sabit bir sayıdır.

$$lr(z) = \max(ax, x)$$

Denklem 11

Softmax Aktivasyon Fonksiyonu: Softmax aktivasyon fonksiyonu derin sinir ağlarında genellikle çıkış katmanında kullanılır. Fonksiyon, sayıları olasılık dağılımına dönüştürmektedir.

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}$$

Denklem 12

Matematiksel formülü Denklem 12'de gösterilen softmax, olasılığın dağılımını 0 ile 1 arasına sıkıştırır.

4. Eniyileme (Optimizasyon) Yöntemleri

Derin sinir ağlarında, öğrenme işleminin gerçekleşebilmesi için hata fonksiyonunun minimum değerinin bulunması gerekmektedir. Optimizasyon algoritmaları ile bu işlem gerçekleştirilmektedir. Optimizasyon, ağın ürettiği çıkış değeri ile gerçek değer arasındaki farkı yani hata fonksiyonunu minimize etmeyi amaçlar. Yapay sinir ağlarının optimizasyonu için en çok kullanılan yöntemlerden biri gradyan inişidir.

Model performansını arttırmak için kullanılan bir diğer önemli araç ise öğrenme ve momentumdur. Öğrenme katsayısı, model parametrelerinin gradyana göre güncelleme miktarlarına karar veren bir hiper-parametredir. Öğrenme katsayısı çok küçük olarak ayarlanmışsa, optimizasyon çok zaman alır ve modelin ağırlığında küçük değişikliklere neden olur. Öğrenme hızı çok yüksekse, optimize edici minimum seviyeyi aşabilir ve salınım yaparak daha da kötüleşebilir. Bu, modelin performansını ciddi seviyede etkiler. Geri yayılım algoritmasında yakınsamayı hızlandırmak ve ağın performansını arttıracak tekniklerden biri olan sezgisel yaklaşım teknikleri momentum katsayısı kullanılarak yapılır. Momentum katsayısı YSA'nın daha hızlı yakınsamasına yardım eder. Temel olarak daha önceki değişimin bir kısmını işlem gören değişime eklemeye dayanır. Momentum katsayısı ağın yerel minimumu aşmasını sağladığı gibi aynı zamanda hatanın azalmasına da yardımcı olur. Optimizasyon algoritmalarında öğrenme katsayısının ve momentumun ayarlanması modelin eğitiminde kritik bir rol oynamaktadır. Ancak, her optimizasyon algoritması ile modeldeki katsayıları tam olarak ayarlamak mümkün değildir. Bu problemi çözmek için gradyan gradyan inişi yöntemini esas alan çeşitli adaptif algoritmalar önerilmiştir.

Momentum SGD: Gradyan inişine momentum katsayısını ekleyen bir stokastik optimizasyon yöntemidir. Momentumlu gradyan inişi, her yinelemede Δw değerini bellekte saklar ve bir sonraki güncelleme değerini gradyan ve önceki güncel değerin doğrusal bir kombinasyonu olarak belirler. Formül şu şekilde ifade edilebilir:

$$V_t = \beta V_{t-1} + (1 - \beta) \nabla_w L(W, X, y) \\ W = W - \alpha V_t$$

Denklem 13

Denklem 13'e göre L kayıp fonksiyonu, α ise öğrenme katsayısıdır.

RMSPProp: Sinir ağlarının eğitiminde kullanılan gradyan tabanlı bir optimizasyon

teknikidir. Her parametre için gradyanların azalan hareketli ortalamasını kullanır. Sinir ağırları gibi çok karmaşık fonksiyonların gradyanları, veri fonksiyon boyunca yayılırken kaybolan gradyanlar problemi yaşar. RMSprop, gradyanı normalleştirmek için kare gradyanların hareketli bir ortalamasını kullanarak bu sorunu çözer. Bu normalleştirme, adım boyutunu (momentum) dengeler. RMSprop, öğrenme oranını bir hiper parametre olarak ele almak yerine uyarlanabilir bir öğrenme hızı kullanır. Bu da öğrenme hızının adaptif olduğu ve zamanla değiştiği anlamına gelir. RMSprop formülü şu şekilde ifade edilebilir:

$$\begin{aligned}v_{dw} &= \beta \cdot v_{dw} + (1 - \beta) \cdot dw^2 \\v_{db} &= \beta \cdot v_{db} + (1 - \beta) \cdot db^2 \\W &= W - \alpha \cdot \frac{dw}{\sqrt{v_{dw} + \epsilon}} \\b &= b - \alpha \cdot \frac{db}{\sqrt{v_{db} + \epsilon}}\end{aligned}$$

Denklem 14

Adam: Momentumlu gradyan iniş algoritması ve RMSProp algoritmasının bir kombinasyonudur. Bu algoritma, gradyanların 'üssel ağırlıklı ortalamasını' dikkate alarak gradyan iniş algoritmasını hızlandırmak için kullanılır. Hareketli ortalamaları kullanmak, algoritmanın minimuma daha hızlı yaklaşmasını sağlar. Derin sinir ağının her ağırlığı için öğrenme oranını uyarlamak için birinci ve ikinci gradyan anlarının tahminlerini kullanmaktadır. Formülü şu şekildedir:

$$\begin{aligned}v_t &= \beta_1 \cdot v_{t-1} + (1 - \beta_1) \cdot g_t \\s_t &= \beta_2 \cdot s_{t-1} + (1 - \beta_2) \cdot g_t^2 \\\Delta w_t &= -\eta \frac{v_t}{\sqrt{s_t + \epsilon}} \cdot g_t \\w_{t+1} &= w_t + \Delta w_t\end{aligned}$$

Denklem 15

Denklem 15'e göre parametreler ise şu şekilde açıklanabilir:

η : Öğrenme katsayısı

g_t : t zamanında w^j boyunca gradyanı

v_t : Gradyanların w_j boyunca Üstel Ortalaması

s_t : Gradyanların w_j boyunca karelerin üstel ortalaması

β_1, β_2 : Hiper-parametreler (0,9 ve 0.99)

5. Maliyet Fonksiyonları

Denklem 3'te görüldüğü üzere maliyet fonksiyonları optimizasyon işlemlerinde sıkça kullanılmaktadır. Çeşitli çalışmalarda hata fonksiyonu olarak da geçen maliyet fonksiyonlarının seçimi problemin tanımına göre değişmektedir. Regresyon problemlerinde ortalama kare hatası (MSE), ortalama mutlak hata (MAE) vb. kullanılırken, sınıflandırma işlemlerinde çok sınıflı SVM için maliyet hesaplanması veya çapraz entropi maliyeti (Cross Entropy Loss) gibi yöntemler kullanılmaktadır. Derin sinir ağlarında optimizasyon bölümünde MSE açıklanmıştır. Diğer bazı maliyet fonksiyonları şu şekildedir:

MAE: MSE'ye çok benzer. İki sürekli değişken arasındaki farkın ölçüsüdür. Matematiksel olarak ifadesi aşağıdaki gibidir:

$$MAE = \frac{1}{N} \times \sum_{j=1}^N |y_j - y'_j|$$

Denklem 16

Ortalama mutlak hatada mutlak değer alınmasının sebebi negatif sonuçlarında maliyet fonksiyonuna katılımının sağlanmasıdır. Örneğin; ölçümleme yapmak istediğimiz değer 90 olsun ve bulduğumuz değer de 85 olsun. Burada hata 5'dir. Birden fazla bulmak istediğimiz değer olduğunda ise durum değişmektedir. Bir diğer değer 50 olsun ve bulduğumuz değer de 55 olsun. Burada hata -5 olacaktır ve tüm hataların toplamı $-5+5=0$ olmaktadır, fakat bu yanlıştır. Burada hata aslında $|-5| + |5| = 10$ olmalıdır. MAE fonksiyonunu MSE'den ayıran temel fark MSE'nin kare alma işleminden dolayı aykırı değer olarak nitelendirebileceğimiz değerleri daha fazla dikkate alarak hesaplama fonksiyonuna katmasıdır. MSE ve MAE genel olarak regresyon problemlerinde kullanılmaktadır. Eğer çözmek istediğimiz problemde aykırı değerler bizim için önemliyse MSE kullanmak daha doğru olacaktır.

Çok Sınıflı SVM için Maliyet Hesaplanması (MultiClass SVM Loss): Görüntü sınıflandırma işlemlerinde hata oranı bulunurken görüntünün skoru ve gerçek görüntüye ait olan sınıfın skoru arasındaki fark hesaplanarak maliyet fonksiyonu bulunur ve diğer sınıflara ait elde edilen skorlar maliyet hesabı için kullanılmaktadır. Örneğin farklı tümör çeşitlerine sahip tahmin edilmesi gereken 3 sınıfımız (t1, t2, t3) olduğunu ve kaydedilen çıktı skor değerlerinin W olduğunu varsayarsak $f(x, W) =$

Wx:

	t1	t2	t3
t1	3.2	1.3	2.2
t2	5.1	4.9	2.5
t3	-1.7	2.0	-3.1

→

	t1	t2	t3
t1	3.2	1.3	2.2
t2	5.1	4.9	2.5
t3	-1.7	2.0	-3.1
Kayıp	2.9	0	10.9

Şekil 11 Çok Sınıflı SVM için Maliyet Hesaplanması

Verilen Örnekte x değerinin sınıfı ve y_i değerlerinin sonuç skorlarını, s sınıf sonuç skorlarının vektör olarak çıktısını temsil ettiğini düşünürsek, Çok Sınıflı SVM için Maliyet Denklem 17'deki gibi hesaplanmaktadır.

$$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$

Denklem 17

Şekil 11'de görüldüğü üzere t1 için maliyet:

$$\begin{aligned}
 &= \max(0, 5.1 - 3.2 + 1) + \max(0, -1.7 - 3.2 + 1) \\
 &= \max(0, 2.9) + \max(0, -3.9) \\
 &= 2.9
 \end{aligned}$$

t2 için maliyet:

$$\begin{aligned}
 &= \max(0, 1.3 - 4.9 + 1) + \max(0, 2 - 4.9 + 1) \\
 &= \max(0, -2.6) + \max(0, -1.9) \\
 &= 0
 \end{aligned}$$

t3 için maliyet:

$$\begin{aligned}
 &= \max(0, 2.2 + 3.1 + 1) + \max(0, 2.5 + 3.1 + 1) \\
 &= \max(0, 5.3) + \max(0, 5.6) \\
 &= 10.9
 \end{aligned}$$

Çok Sınıflı SVM için tüm maliyet hesaplanması:

$$L = \frac{1}{N} \sum_{i=1}^N L_i$$

Denklem 18

$$L = \frac{2.9 + 0 + 10.9}{3}$$

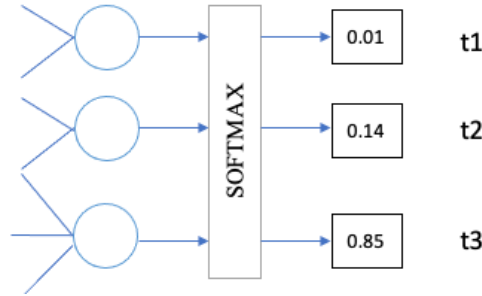
$L = 4.6$ olarak hesaplanır.

Çapraz Entropi Maliyet Hesabı (Cross Entropy Loss): İkili ve çoklu Sınıflandırma işlemlerinde sıkça kullanılmaktadır.

$$CE = - \sum_{i=1}^C y_i * \log(y'_i)$$

Denklem 19

Denklem 19'da C sınıf sayısını temsil etmektedir. Çoklu sınıflandırma probleminden bir örnekle açıklamak gerekirse, t1, t2 ve t3 tümör sınıflandırma problemini tekrar ele alalım. Yapay sinir ağlarının son katmanında Softmax aktivasyon fonksiyonunun Şekil 12'deki gibi bize şu olasılık dağılımını verdiğini düşünelim:



Şekil 12 Çapraz Entropi Maliyet Hesabı: Softmax Aktivasyon Fonksiyonu Örneği

Softmax aktivasyon fonksiyonunun 0 ile 1 dağılımında olasılık dağılımını verdiğini söylemiştik. Sınıf tahminleri sırayla 0.01, 0.14 ve 0,85 olarak simüle edilmiştir. Tahminleri (y') ve gerçekleşmesi beklenen çıktıları (y) vektör olarak ifade edersek:

$$y' = \begin{bmatrix} 0.01 \\ 0.14 \\ 0.85 \end{bmatrix} \quad y = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

Denklem 20

Maliyet hesabı $CE = - 0 * \log(0.01) - 1 * \log(0.14) - 0 * \log(0.85) = 0,85$ olarak

hesaplanır.

E. Evrişimsel Sinir Ağları (CNN)

Bu bölümde ilk olarak bilgisayarlı görü ile ilgili ön bilgi verildikten sonra evrişimsel sinir ağının çalışma prensibi anlatılmıştır. Analiz ve uygulama bölümünde evrişimsel sinir ağının tıbbi görüntü işlemede kilit noktalarını anlamak için, CNN mimarisi içindeki her bir katmanın işlevselliği açıklanmıştır.

1. Ön Bilgi

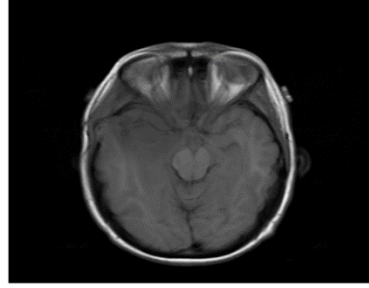
Bilgisayarlı görü, dijital görüntülerin bilgisayar tarafından nasıl anlamlandırılacağını anlatan disiplinler arası bilimlerden biridir. Görüntü işleme, mobil kameralar, televizyonlar, farklı ekran türleri, tarayıcılar, yazıcılar ve diğer birçok uygulama etrafımızda görülen görsel efektleri bize sunmak için sayısal görüntü işlemeyi kullanır. Bilgisayarlı görü, görüntünün özelliklerini almak ve değiştirmek veya özelliklerini iyileştirmek için görüntünün matematiksel gösterimi ve işlenmesidir. Sayısal görüntüler genel olarak iki boyutlu matris gösterimi şeklinde ele alınır. Bu iki boyutlu matrisler fonksiyon olarak gösterilir. İki bağımsız değişken olan x ve y için bir g fonksiyonu düşünecek olursak g(x,y) fonksiyonu görüntü fonksiyonu olarak kabul edilebilir. g fonksiyonu yoğunluk veya görüntü sinyali olarak da adlandırılabilir. Bağımsız değişkenler x ve y, sırasıyla bir görüntüdeki piksellerin satırını ve sütununu gösteren değişkendir. Bu formülde g(x,y) fonksiyonu bir görüntüdeki x satırı ve y sütunu tarafından tanımlanan konumdaki piksel için ışığın yoğunluğudur (Gonzalez & Woods, 2001). Şekil 13 dijital bilgisayarlarda sayısal görüntülerin en çok kullanılan temsili olan yoğunluk kuralını göstermektedir. Bu gösterimden yola çıkarak herhangi bir sayısal görüntü aşağıdaki gibi matris biçiminde gösterilebilir:

$$g(x,y) = \begin{bmatrix} g(1,1) & g(1,2) & \cdots & g(1,M) \\ g(2,1) & \cdots & \cdots & \vdots \\ \vdots & & & \\ g(2,M) & & & g(N,M) \end{bmatrix}$$

Denklem 21

Aşağıda açıklanan şekilde görüntü M çizgileri ve N sütunlarından oluşmaktadır. M ve N sayıları görüntünün piksel cinsinden boyutuna karşılık gelir. Görüntü boyutu santimetre gibi mesafe yerine genel olarak piksel cinsinden tanımlanır. 1024*1024 piksel

boyutundaki görüntü, içinde 1024*1024 farklı yoğunluk değerleri bulunan bir görüntüdür.



0	0	0	0	0	0	0	0
0	0	0	5	9	7	0	0
0	0	8	11	12	7	0	0
0	0	8	8	8	8	8	0
0	0	13	12	1	12	14	0
0	0	2	6	4	2	0	0

Şekil 13 Bilgisayar Görüntü Örneği

Sayısal görüntülerde yoğunluk değerleri görüntünün gösterimine bağlıdır. Bilgisayarlarda görüntü yoğunluğu, Şekil 13'teki gibi minimum ve maksimum değerlerin var olduğu kapalı bir aralıktır. Uygulamada ise bir görüntünün yoğunluk aralığı görüntü işlemenin amacına göre seçilir. Yoğunluk değeri yükseldikçe genellikle bir görüntünün daha parlak piksellerini ifade eder. Beyaz veya parlak renkler için daha yüksek yoğunluk değerleri verilirken, koyu piksellere daha düşük yoğunluk değerleri verilir. Teoride bir görüntüdeki piksellerin parlaklığını tanımlamak için yoğunluk değerlerinin gerçek pozitif değerler olması yeterlidir. Fakat ikili sayılar (0, 1) kullanan sayısal sistemlerin yapı ve donanım sınırlamaları bir görüntüdeki yoğunluk değerlerinin sayısının 2'nin tamsayı gücü olduğunu işaret eder. 8 bitlik görüntüler 28 farklı yoğunluk değerine sahipken, öte yandan 16 bit olan görüntüler 216 farklı yoğunluk değerine sahiptir. 8 bitlik bir görüntüdeki değerler aralığı ise 0 ile 255 arasındadır.

2. Tanım

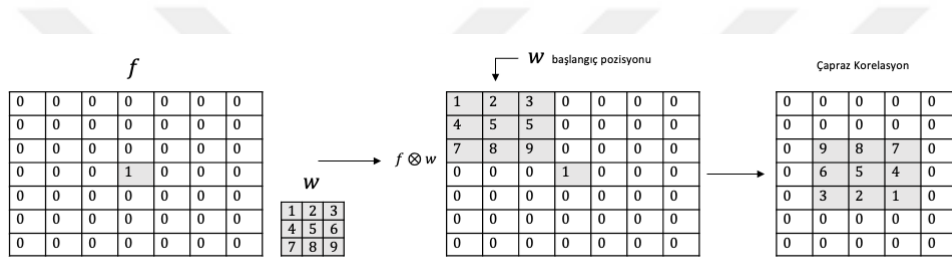
Evrişimsel sinir ağları, otonom arabalar, yüz tanıma sistemleri gibi bilgisayarlı görünün sıkça kullanıldığı uygulamaların derin tahmin omurgasını oluşturmaktadır. Temel matris çarpım operasyonu yerine konvolüsyon (evrişim) operasyonu gerçekleştirilmektedir. Bu işlemi anlamak için öncelikle matrislerde çapraz korelasyon (cross-correlation) işlemini anlamak gerekir. Görüntü işlemede korelasyon işlemi, genellikle çekirdek olarak adlandırılan bir filtre maskesini görüntü üzerinde hareket ettirme ve her bir konumdaki rengi temsil eden sayıların filtre ile

çarpımlarının toplamını hesaplama işlemidir. Korelasyon, filtrenin hareket etme işlemidir. Başka bir deyişle, korelasyonun ilk değeri filtrenin pozisyonuna karşılık gelir, ikinci değer filtrenin bir birim yer değiştirmesine karşılık gelir. 2 boyutlu bir korelasyon işleminin matematiksel olarak formülü şu şekilde ifade edilebilir:

$$f \otimes w(x,y) = \sum_{j=0}^N \sum_{i=0}^N f(i,j) * w(x+i,y+j)$$

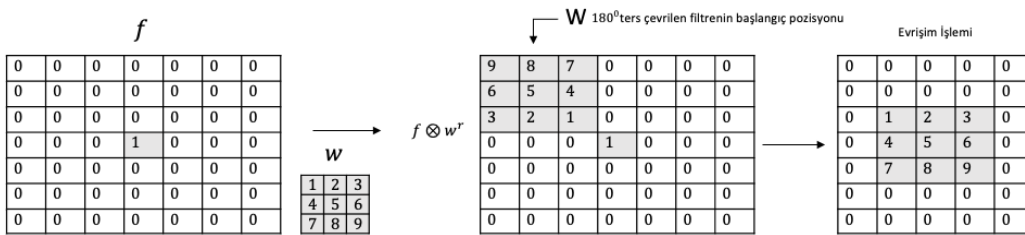
Denklem 22

Filtre başlangıç pozisyonu Şekil 14'deki gibi (0,0) noktasıdır ve Denklem 22'ye göre korelasyon işlemi gerçekleşir.



Şekil 14 Çapraz Korelasyon İşlemi

Evrişim işlemi de çapraz korelasyon ile benzerdir fakat burada öncelikle filtre 180 derece çevrilir, sonra görüntüye uygulanır. Şekil 15'de görüldüğü üzere çapraz korelasyona göre filtrenin öncelikle çevrilmesinin sebebi filtrenin resim üzerinde tam olarak işlenmesini sağlamaktadır.



Şekil 15 Evrişim İşlemi

Evrişim işleminin matematiksel formülü aşağıdaki gibidir.

$$f \otimes w(x, y) = \sum_{j=0}^N \sum_{i=0}^N f(i, j) * w(x - i, y - j)$$

Denklem 23

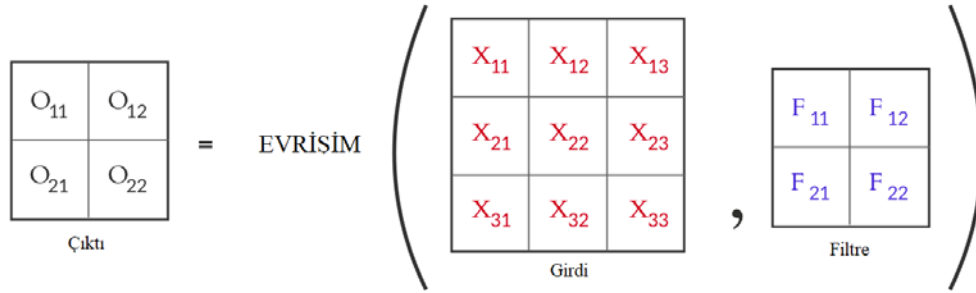
Evrişimsel Sinir Ağları ilk olarak 1980 yılında Fukushima tarafından Neocognitron adıyla gündeme getirilmiştir. Hubel ve Weisel tarafından bahsedilen sinir sisteminin hiyerarşik modelinden ilham alınarak oluşturulmuştur. Ancak model karmaşık algoritması nedeniyle popülerlik kazanamamıştır. 1989'da Yann LeCun ABD Posta Servisi tarafından el yazısı formatında Posta Kodu Tanıma için kullanılan LeNet adlı bir mimari sunmak için Neocognitron kavramlarıyla birlikte geri yayılımı kullanmıştır. Yann LeCun bu proje üzerinde daha fazla çalışmış ve sonuç olarak 1998'de bugün CNN'de hala kullanılan bazı temel kavramları tanıtan ilk modern konvnet olan LeNet-5'i kullanıma sunmuştur (LeCun et al., 1998). Buna ek olarak makine öğreniminde belki de en ünlü kıyaslama veri seti olan MNIST el yazısı formatında veri seti yayınlamıştır. 1990'larda bilgisayarlı görü alanı odağını değiştirmiştir ve birçok araştırmacı CNN mimarileri üzerinde çalışmayı bırakmıştır. 2015'te evrişimsel sinir ağları tabanlı mimari ResNet yapay sinir ağları problemlerinde başarılı sonuçlar elde ederek adından sıkça söz etmiştir. Evrişimsel sinir ağı da girdiler tarafından içe aktarılan ve doğrusal olmayan bir dönüşüm gerçekleştiren kendi kendine optimize edilmiş nöronlardan oluştuğundan yapay sinir ağına benzerdir (Bar et al., 2015). Yapay sinir ağlarıyla karşılaştırıldığında evrişim sinir ağı ağ mimarisine görüntüye özgü özellikleri kodlayarak ağı görüntü tabanlı özellik öğrenimi için daha uygun hale getirdiğinden görüntülerde tanımda yaygın olarak kullanılır (Ahn et al., 2016). Evrişimsel sinir ağı içerisinde beş temel unsur bulunmaktadır. Bunlar: giriş katmanı, evrişim katmanı, doğrusal olmayan katman, havuzlama katmanı ve tam bağlantı katmanı. Bir evrişimsel sinir ağının temel fonksiyonelliği şu beş temel noktada toplanabilir:

1. Giriş katmanı görüntünün piksel değerlerini tutar.
2. Evrişim katmanı girdilerin evrişim hesabını içerir.
3. Doğrusal olmayan katman önceki katmanın ürettiği girdilerin doğrusal olmayan dönüşümünü uygulamak için kullanılır.
4. Havuzlama katmanı aktivasyondaki parametrelerin sayısını azaltarak verilen girdilerin örnekleme yapar.

5. Tam bağlantı katmanı aktivasyonlardan elde edilen sınıf puanlarını gösterir ve sınıflandırma için kullanılır.

3. Evrişim Katmanı

Evrişimsel sinir ağlarının ana hesaplamaları yapan katmanı evrişim katmanıdır. Evrişim katmanı burada görüntüye farklı filtreler uygulayarak görüntünün kenar, çizgi köşe gibi farklı özelliklerini ortaya çıkarır. Evrişim katmanında filtreler geri yayılım algoritması sayesinde öğrenebilir filtre ağırlıklarına sahiptir. Filtreler ile evrişime giren girdi sayesinde aktivasyon haritaları üretilir. Şekil 16’da bir evrişim işlemi görselleştirilmiştir. X ve F fonksiyonu arasındaki evrişim işlemi O aktivasyon çıktısını vermektedir.



Şekil 16 Özellik Haritası

Şekil 16’da 3x3’lük matrise 2x2’lik bir filtre uygulandığında 2x2’lik özellik haritası oluştuğu görülmektedir. Filtrenin başlangıç pozisyonundan itibaren birer sağa kaydırılması ile çıkabilecek matris görölse de bunun hesaplanması şu şekildedir:

$$O_{i,j} = (\text{girdi boyutu} - \text{filtre boyutu} + 1)$$

Denklem 24

Filtre başlangıç noktasından 32x32 ‘lik bir resme 5x5’lik filtre uygulandığında 28x28’lik bir matris elde edilecektir. Bu matris özellik haritası veya aktivasyon haritası olarak bilinmektedir. Evrişim katmanında dikkate alınması gereken üç önemli hiperparametre bulunmaktadır: Derinlik, adım (stride) ve dolgu ayarı (padding) (Wang, 2018). Çıkış hacminin derinliği evrişim için kullanılan filtre sayısına karşılık gelir ve bu filtrelerin her biri görüntü girdisinin kendi özellik haritasını oluşturur. Adım, filtrenin matris üzerinde nasıl hareket edileceğini belirler. Adım 1 olarak belirlenirse, filtreler her seferinde 1 piksel sağa hareket eder. Adım 3

olarak belirlendiğinde, filtreler her seferinde 3 piksel atlayarak soldan sağa ve yukarıdan aşağı hareket eder. Adım çok büyük seçilirse aktivasyon çıktılarının küçüleceği açıktır. Dolgu ayarı (padding) genellikle sınırdaki kalan piksellerin filtreler tarafından daha fazla işlenmesi için matrisin etrafında aynı boyutta sıfır eklenmesinden ibarettir. Burada unutulmaması gereken nokta, filtre sayısı, adım sayısı ve dolgu ayarı birer hiperparametre olup, evrimsel sinir ağlarında değiştirilebilmektedir. Filtre sayısı ve dolgu genişliği gibi de düşünüldüğünde öznitelik haritasının çıktı matris boyutu şu şekildedir:

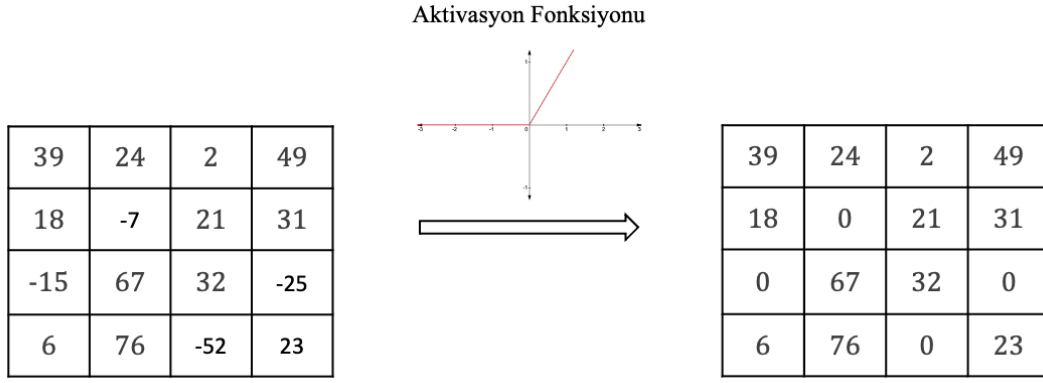
$$O_{i,j} = \frac{\text{girdi boyutu} + 2 * \text{padding} - \text{filtre boyutu}}{\text{stride}} + 1$$

Denklem 25

Evrşim katmanında parameter sayısını azaltmak için parametre paylaşımı işlemi yapılır. Parametre paylaşımı, bir bölgenin özelliğinin belirli bir uzamsal bölgede hesaplanması faydalıysa, o zaman başka bir bölgede de faydalı olacağı varsayımına dayanmaktadır. Evrşim katmanı içinde, çıktı hacmi içindeki her aktivasyon haritası aynı ağırlıklar ve sapma değerine bağlı olarak ayarlanır, bu nedenle evrşim katmanı tarafından üretilen parametrelerin sayısında azalma meydana gelir. Bu teoriye dayanarak, geri yayılım aşamasında, çıktıdaki her nöron genel gradyanı temsil eder, böylece yalnızca küçük bir ağırlık grubunun güncellenmesi yapılır (Baydar, 2018).

4. Doğrusal Olmayan Katman

Evrşim katmanında, girdi üzerinde doğrusal olmayan transformasyon uygulanır. Derin sinir ağlarında doğrusal olmayan dönüşüm fonksiyonu olarak genellikle sigmoid veya hiperbolik tanjant kullanılmaktadır. Ancak görüntü işleme için görüyü seyrelttiğinden ve veri seti eğitiminin hızlı olmasından dolayı Relu gibi fonksiyonlar tercih edilmektedir.

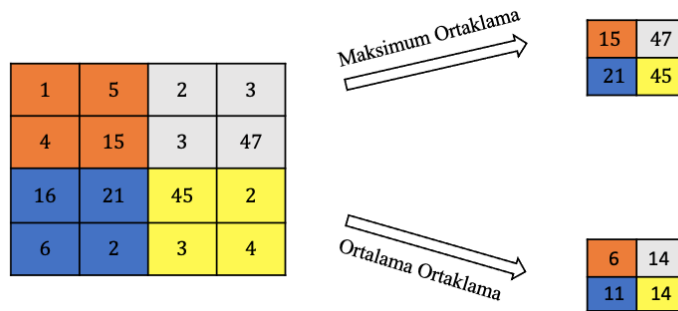


Şekil 17 Doğrusal Olmayan Katman: Relu Örneği

Şekil 17’de Relu uygulanmış bir matris görülmektedir. Denklem 10 uyarınca negatif değerler 0’a, diğer değerler rakamın kendisine eşittir. Relu’da çıkış ile giriş aynı boyuttur.

5. Havuzlama (Ortaklama) Katmanı

Evrişim katmanından sonra veriler havuzlama katmanına gelir. Havuzlama katmanının temel amacı boyutsallığı, parametre sayısını ve hesaplama karmaşıklığını azaltmaktır. Havuzlama katmanı, girdinin her bir özellik haritası üzerinde çalışır ve tanımlanan havuzlama fonksiyonunu kullanarak boyutsallığını ölçeklendirir. Literatürde, maksimum havuzlama ve ortalama havuzlama, en sık kullanılan havuzlama fonksiyonlarıdır. Şekil 18, her iki havuzlama yönteminin işlemlerini göstermektedir.



Şekil 18 Havuzlama Katmanı: Maksimum ve Ortalama

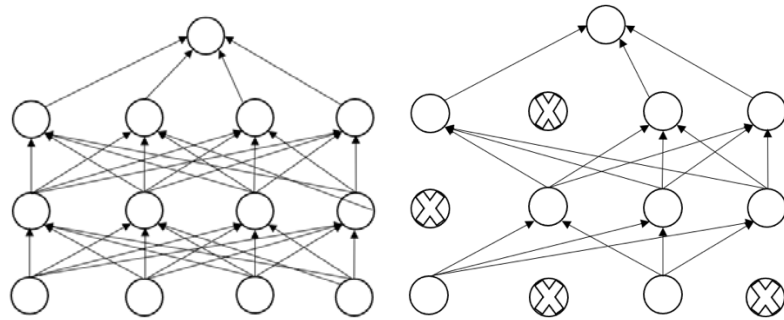
Şekil 18 detaylı incelendiğinde, 2 x 2 filtrenin 2 adım ile 4 x 4’lük aktivasyon matrisine işlendiğini görürüz. Maksimum ortaklama filtrenin uygulandığı yerin maksimum pikseli, ortalama ortaklama filtresinin uygulandığı yerde ise o bölgenin piksel ortalamasının alındığı ise kolayca görülmektedir.

6. Tam Bağlantılı (FC) Katman

Tam bağlantılı katmanlar çok katmanlı derin sinir ağı yapısına sahiptir. Çıkış sayısı burada belirlenir ve aktivasyon fonksiyonları burada da kullanılır. Bağlantılı katmanlar evrişim katmanında çıkarılan öz niteliklere göre verinin sınıflandırmasını yapar. Bu katmanlarda aşırı öğrenme veya eksik öğrenme gibi durumlarda seyrekleştirme veya nöron sayısını arttırmak gibi çeşitli işlemler gerçekleştirilebilir.

7. Seyrekleştirme (Dropout)

Temel olarak derin sinir ağlarında model eğitiminde iki problem meydana gelir: Aşırı öğrenme (overfitting) ve eksik öğrenme (underfitting). Eğitim veri setinde yüksek başarı sağlanmasına rağmen test veri setinde çok düşük başarı sağlanıyorsa model aşırı öğrenme gerçekleşmiş, verileri ezberlemiş ve test veri setinde de bu durumu probleme yansıtmıştır. Veri biliminde bu durum yüksek varyans düşük yanlılık olarak ifade edilir. Düzenleme (regularization) aşırı öğrenme problemine karşı kullanılan bir tekniktir. Dropout, evrişimsel sinir ağlarında tam bağlantılı katmanda her bir epokta bazı yapay sinir hücrelerinin işleme dahil edilmemesidir. Böylece genelleştirilme sağlanarak aşırı öğrenmenin önüne geçilmiş olunur. Şekil 19'da dropout uygulanmış ve uygulanmamış tam bağlantılı katman örneği gösterilmiştir.



Şekil 19 Tam Bağlantılı Katmanlarda Seyrekleştirme Tekniği

8. Yığın Normalleştirme

Eğitim sırasında her katman hatasını düzeltmeye çalışır. Ancak, katmanlar ayrı ayrı hareket eder. Örneğin 2. katmanın çıktısı ile 3. katman beslenir. Bir katmandaki parametre değişikliği sonraki katmanlarda da değişime yol açar. Giriş katmanlarındaki bu kaymalar sebebiyle hesaplamalar için zaman gittikçe artar. Çünkü, 3. katman'daki

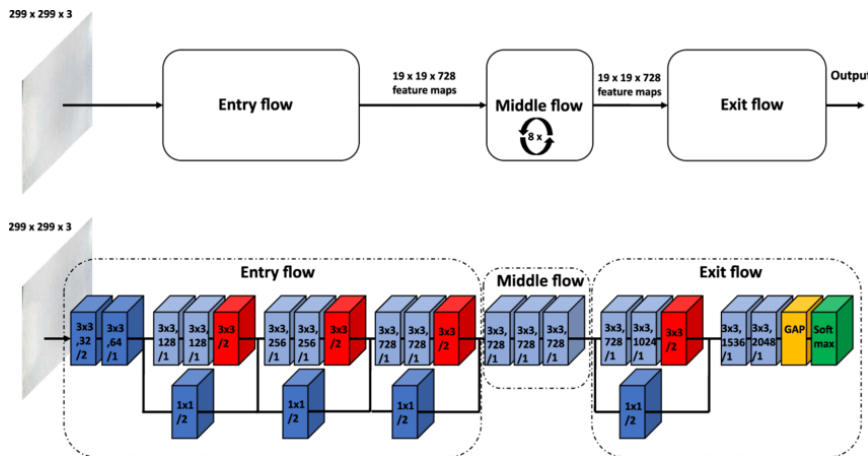
öğrenmeye başlamak için 2. katmanın öğrenmesinin bitmesi gerekmektedir. Bu kaymaları azaltmak için yığın normalleştirme uygulanır.

Derin öğrenmede kullanılan veri çok büyük olduğundan, ağıın girdisi önemli ölçüde değişir. Normalization işlemi; tüm girdilerin dağılımının ortalamasını sıfır, standart sapmasını 1 olacak hale getirmektir. Dolayısıyla değerler -1 ila 1 arasına sıkıştırılmış olur. Bu normalization işlemini girişe uygulayabiliriz. Fakat bu normalleştirmeden 2. katman faydalanamaz hale gelir. Bu yüzden de katmanlar arasında normalizasyon yapılır. Buna yığın normalleştirme denmektedir. Yığın normalleştirme sayesinde ağıdaki katmanlar, önceki katmanın öğrenmesini beklemek zorunda kalmaz. Eş zamanlı olarak öğrenime olanak sağlar. Yığın normalleştirme eğitimin hızlanmasını sağlar (coimer.medium.com).

9. Evrişimsel Derin Sinir Ağı Modelleri

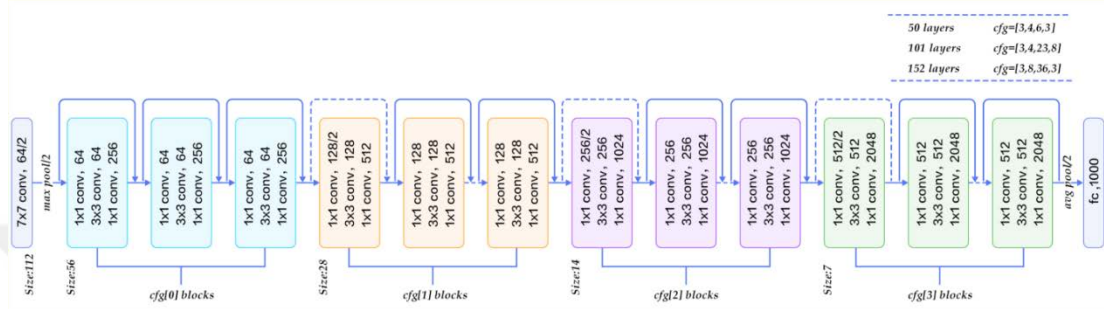
CNN mimarilerinde daha verimli öğrenme işleminin gerçekleştirilmesi için literatüre sunulan bazı derin sinir ağı mimarileri bulunmaktadır. Bu mimarilerin neredeyse tamamı, ağı boyunca kullanılan tekrarlanabilir bir birime dayanmaktadır. Bu mimariler, görüntü sınıflandırma, nesne algılama, görüntü bölütleme ve diğer birçok makine öğrenimi işlemi için kullanılabilen zengin öznitelik çıkarıcılar olarak kullanılmaktadır. En başarılı modern mimarilerden örnekler şu şekildedir:

XCEPTION: 71 adet katmandan oluşan bir derin sinir ağı mimarisidir. Imagenet veritabanından bir milyondan fazla görüntü üzerinde eğitilmiştir. 1000'e yakın kategoriye ayırabilen bu derin sinir ağı, giriş boyutu olarak 299 x 299 piksel almaktadır. Mimarinin gösterimi Şekil 20'deki gibidir. Şekil 20



Şekil 20 XCEPTION Mimarisi: (Westphal & Seitz, 2021)

RESNET: ImageNet 2012 yarışmasını kazanan ilk CNN tabanlı mimariden (AlexNet) sonra, sonraki her mimari, hata oranını azaltmak için derin bir sinir ağına daha fazla katman kullanmıştır. Ancak katman sayısını artırdığımızda, derin öğrenmede kaybolan gradyan problemi (vanishing gradient problem) olarak adlandırılan yaygın bir sorun oluşmaktadır. Bu durum, gradyanın 0 veya çok büyük olmasına neden olmaktadır. Böylece katman sayısı artırıldığında eğitim ve test hata oranı da armaktadır.

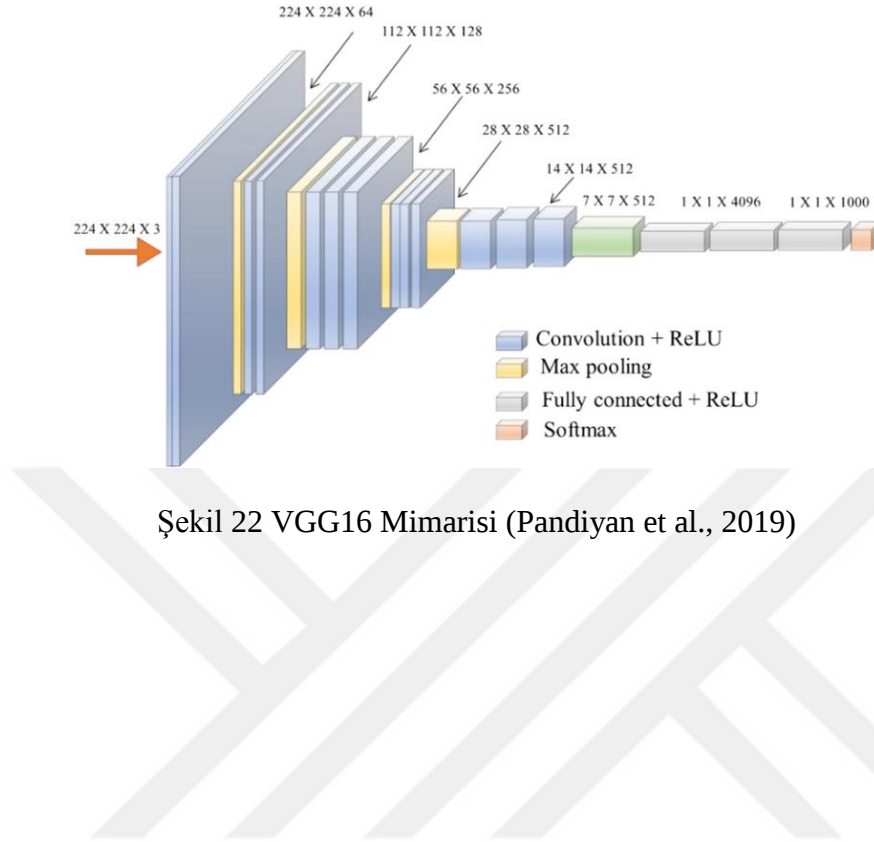


Şekil 21 RESNET Mimarisi (blog.devgenius.io)

2015 yılında Microsoft araştırmacıları tarafından önerilen ResNet, Rezidüel Ağ adı verilen yeni bir mimariyi tanıtmıştır. Kaybolan gradyan problemini çözmek için önerilen bu mimari, Rezidüel Ağ adı verilen kavramı da hayatımıza sokmuştur. Bu ağda bağlantı atlama adı verilen bir teknik kullanılmaktadır. Atlama bağlantısı, eğitimi birkaç katmandan atlatır ve doğrudan çıktıya bağlanır. Bu ağın arkasındaki yaklaşım, katmanların temel haritalamayı öğrenmesi yerine, ağın artık haritalamaya uymasına izin vermektir. Bu tür bir atlama bağlantısı eklemenin avantajı, herhangi bir katmanın mimarinin performansına zarar vermesi durumunda, düzenleme tarafından atlanmasıdır. Böylece, kaybolan/patlayan gradyanın neden olduğu problemler olmadan çok derin bir sinir ağının eğitilmesiyle sonuçlanır. Bu ağ, VGG-19'dan ilham alan 34 katmanlı düz ağ mimarisini kullanmaktadır (He et al., 2015).

VGG16: 2014 yılında Imagenet yarışmasını kazanan derin sinir ağı mimarisidir. VGG16, çok sayıda hiperparametreye sahip olmak yerine, adım 1 ile 3x3 filtrenin evrişim katmanlarına odaklanmaları ve her zaman aynı dolguyu ve adım 2'nin 2x2 filtresinin maksimum havuzlama katmanını kullanmalarıdır. Tüm mimari boyunca tutarlı bir şekilde evrişim ve maksimum havuz katmanlarını kullanmaktadır. Sonunda 2 FC'ye (tamamen bağlı katmanlar) ve ardından çıktı için bir softmax'a sahiptir.

VGG16'daki 16, ağırlıkları olan 16 katmana sahip olduğunu ifade eder. VGG16 yaklaşık 138 milyon eğitilebilir parametreye sahiptir (Pandiyen et al., 2019).



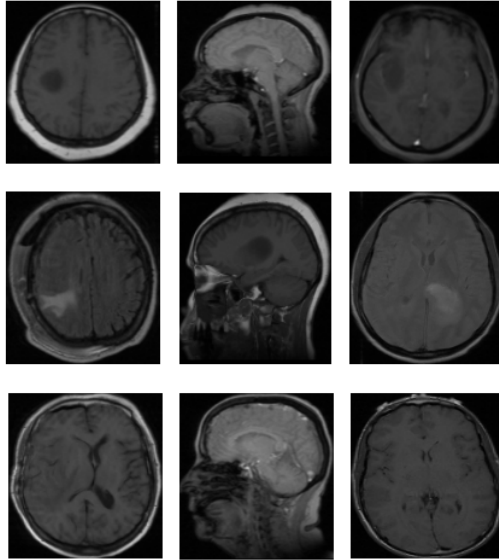
Şekil 22 VGG16 Mimarisi (Pandiyen et al., 2019)

III. MATERYAL VE YÖNTEM

A. Materyal

1. Beyin Tümörü Tanısına Genel Bakış

İnsan vücudundaki organların hücrelerinin anormal büyümesine tümör denir. Tümör tipleri iyi veya kötü huylu olmak üzere ikiye ayrılır. İyi huylu tümörlerde kanser tehlikesi olmamakla birlikte diğer organlara yayılmadan mutlak suretle tedavi edilmesi gerekmektedir. Kötü huylu tümörler kanserojendir, tedavi edilse bile yayılma riski vardır (Council et al., 2014). Beyin insan vücudunda istemli ve istemsiz hareketlerin kontrol merkezidir. Beynin çok daha fazla görevleri vardır fakat burada değinilmeyecektir. Kısaca sağlıklı ve uzun yaşamak için beyin sağlığı hayati derecede önemlidir. Çevresel faktörler ve genetik faktörler sebebiyle beyinde tümörler gelişebilmektedir (abta.org).



Şekil 23 Beyin Tümörü MR Görüntüleri

Beyinde oluşan tümör, işlevselliği bozar ve sağlıklı beyin dokularına zarar verir. Şekil 23’de farklı tipte beyin tümörüne sahip beyin MR görüntü örnekleri gösterilmiştir. Bu tümörler beyinde oluşum yerlerine göre farklı şekilde sınıflandırılırlar (Rana & Singh, 2015).

- Tip ve sınıf
- Birincil veya ikincil tümör
- İyi huylu veya kötü huylu tümör
- Tümörün bulunduğu yer

Birincil beyin tümörleri ilk başta beynin bölgesinde meydana gelir. İkincil beyin tümörleri, önce vücudun başka yerinde oluşur, sonrasında beyne yayılmaktadır. Bunlara metastatik beyin tümörleri de denmektedir. Dünya Sağlık Örgütü (WHO) 2016 yılı raporuna göre merkezi sinir sistemindeki tümörlerin sınıflandırılmasında histolojik ve moleküler genetik özellikler kullanılmaktadır. Bu yeni kategorizasyonda, beyindeki destekleyici dokuyu oluşturan glial hücrelerdeki (gliomlar) büyüme paterni ve genetik mutasyonlara göre gruplandırılmıştır (Louis et al., 2016). Bu araştırma, WHO sınıflandırmasındaki en sık görülen üç glioma tipine odaklanmıştır. Astrositom, Glioblastoma (Glioblastoma Multiforme) ve Oligodendroglioma, araştırma için seçilen tümör tipleridir. Astrositom tümörleri, beynin destek dokusunu oluşturan hücrelerden kaynaklanır. Glioblastomlar, derece 4 astrositom tümörleridir. Bunlar beynin serebral yarım küresinde bulunur ve yaşa bağlı olarak hızla büyüyebilir. Glioblastomlar tedavi edilmesi en zor tümör tipi olarak kabul edilirler. Oligodendroglioma tümörleri, oligodendrosit hücrelerinden meydana gelir. Bu tümörler ayrıca serebral hemisferde daha sık bulunur. Birincil beyin tümörlerinin yaklaşık %7'si Astrositomlar, %17'si Glioblastomlar ve %2'si Oligodendrogliomlardan oluşmaktadır.

2. Veri Seti

Tıbbi görüntü elde etmek amacıyla çok çeşitli teknikler kullanılmaktadır ve manyetik rezonans (MR) bunlardan biridir. Manyetik rezonans görüntüleme (MR), anormal beyin tümörlerini tespit etmede kullanılan en yaygın görüntüleme tekniğidir. MR, beyindeki sinirsel aktiviteleri ölçmek amacıyla kullanılabilen noninvaziv bir tekniktir. MR görüntüleri, beyindeki çeşitli anomalilerin fizyoloji, histoloji, genetik, hemodinamik ve kimya ile ilgili yönlerine ilişkin değerli bilgi sağlama bağlamında yüksek bir potansiyele sahiptir (Mabray, 2016). MR görüntüleri sayesinde beyin anomalileri kolaylıkla tespit edilebilmekte ve sergilenebilmektedir (Gudigar et al, 2019). Geleneksel olarak, MR görüntüleri, beyindeki anormal durumları tespit ederken röntgen uzmanları tarafından manuel olarak analiz edilir. Çok sayıdaki görüntünün manuel olarak yorumlanması ise oldukça fazla zaman gerektirmekte ve

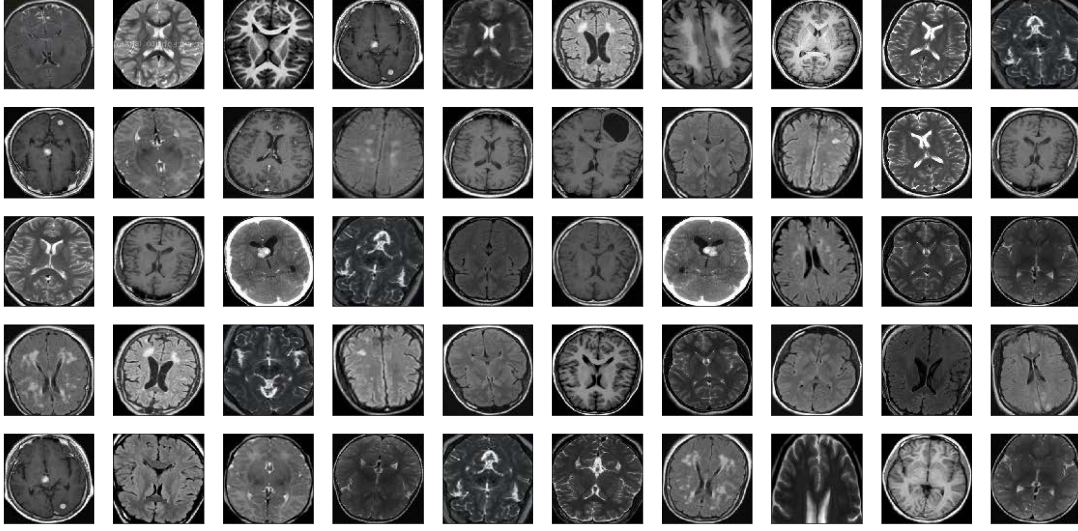
güç olmaktadır. Bu nedenle, bilgisayar destekli tespit, kesin ve hızlı tanı koymada yardımcı olmaktadır.

Çalışmada beyin tümörleri tanısı için Rembrandt veri seti kullanılmıştır (Scarpace et al., 2019). Rembrandt veri seti National Cancer Institute tarafından paylaşılan “The Cancer Imaging Archive (TCIA)” veri tabanına ait bir veri setidir. TCIA medikal görüntüler içeren erişilebilir geniş bir arşivdir. DICOM medikal görüntü formatında yaklaşık 10GB’lık, 130 hastadan alınan 110.020 MR görüntü verisi bulunmaktadır. Şekil 24’de kullanılan veri setinin öğelerini tanımlayan üst veri gösterilmiştir.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
Sample	Age at Dx (years)	Gender	Survival (months)	Disease	Grade	Race	Institution	Karnofsky	Neurologic	MMI Desc	Followup	1 Steroid Do	Anti Conv	Prior Ther	Prior Ther
900_00_1961	--	--	15	ASTROCYTOMA	II	--	TJU	--	--	--	--	--	--	--	--
900_00_5332	--	--	15	GBM	--	--	TJU	--	--	--	--	--	--	--	--
900_00_5308	--	--	17	ASTROCYTOMA	III	--	TJU	--	--	--	--	--	--	--	--
900_00_5316	--	--	17	GBM	--	--	TJU	--	--	--	--	--	--	--	--
900_00_5317	--	--	17	GBM	--	--	TJU	--	--	--	--	--	--	--	--
900_00_5338	--	--	15	GBM	--	--	TJU	--	--	--	--	--	--	--	--
900_00_5339	--	--	15	GBM	--	--	TJU	--	--	--	--	--	--	--	--
900_00_5341	--	--	15	OLIGODENDROGLIOMA	--	--	TJU	--	--	--	--	--	--	--	--
900_00_5342	--	--	15	GBM	--	--	TJU	--	--	--	--	--	--	--	--
900_00_5345	--	--	16	OLIGODENDROGLIOMA	II	--	TJU	--	--	--	--	--	--	--	--
900_00_5346	--	--	10	GBM	--	--	TJU	--	--	--	--	--	--	--	--
900_00_5379	--	--	16	GBM	--	--	TJU	--	--	--	--	--	--	--	--
900_00_5380	--	--	16	OLIGODENDROGLIOMA	--	--	TJU	--	--	--	--	--	--	--	--
900_00_5381	--	--	4	GBM	--	--	TJU	--	--	--	--	--	--	--	--
900_00_5382	--	--	16	ASTROCYTOMA	II	--	TJU	--	--	--	--	--	--	--	--
900_00_5384	--	--	9	GBM	--	--	TJU	--	--	--	--	--	--	--	--
900_00_5385	--	--	13	ASTROCYTOMA	III	--	TJU	--	--	--	--	--	--	--	--
900_00_5393	--	--	13	GBM	--	--	TJU	--	--	--	--	--	--	--	--
900_00_5396	--	--	13	GBM	--	--	TJU	--	--	--	--	--	--	--	--
900_00_5404	--	--	12	GBM	--	--	TJU	--	--	--	--	--	--	--	--
900_00_5412	--	--	12	OLIGODENDROGLIOMA	--	--	TJU	--	--	--	--	--	--	--	--
900_00_5413	--	--	2	GBM	--	--	TJU	--	--	--	--	--	--	--	--
900_00_5414	--	--	2	GBM	--	--	TJU	--	--	--	--	--	--	--	--
900_00_5445	--	--	2	GBM	--	--	TJU	--	--	--	--	--	--	--	--
900_00_5458	--	--	7	GBM	--	--	TJU	--	--	--	--	--	--	--	--
900_00_5459	--	--	10	ASTROCYTOMA	III	--	TJU	--	--	--	--	--	--	--	--
900_00_5462	--	--	2	GBM	--	--	TJU	--	--	--	--	--	--	--	--
900_00_5468	--	--	10	ASTROCYTOMA	II	--	TJU	--	--	--	--	--	--	--	--
900_00_5476	--	--	9	OLIGODENDROGLIOMA	II	--	TJU	--	--	--	--	--	--	--	--
900_00_5477	--	--	5	ASTROCYTOMA	II	--	TJU	--	--	--	--	--	--	--	--
900_00_5299	--	--	6	GBM	--	--	TJU	--	--	--	--	--	--	--	--
900_00_5393	--	--	5	GBM	--	--	TJU	--	--	--	--	--	--	--	--
HF0658	50-54	M	10.6	ASTROCYTOMA	II	WHITE	HENRY FORD HOSPITAL (RETRO)	--	--	rim enhancement and r	--	NASS	NASS	--	--
HF0652.4	50-54	M	19.6	GBM	IV	--	HENRY FORD HOSPITAL (RETRO)	--	--	--	--	--	--	--	--
HF0763	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
HF0828	35-39	M	--	OLIGODENDROGLIOMA	III	--	HENRY FORD HOSPITAL (RETRO)	--	--	--	--	--	--	--	--
HF0835	35-39	F	45.3	OLIGODENDROGLIOMA	II	WHITE	HENRY FORD HOSPITAL (RETRO)	90	--	non contrast enhancing	--	NASS	Yes NASS	--	--
HF0855	55-59	M	13.7	ASTROCYTOMA	II	WHITE	HENRY FORD HOSPITAL (RETRO)	90	--	ring enhancement; this irregular si	--	NASS	NASS	--	--

Şekil 24 Veri seti ile ilgili üst veri (metaveri)

Üst veride hastanın yaşı, cinsiyeti, hangi tümör tipine sahip olduğu ve verinin hangi hastaneden alındığına dair bilgiler bulunmaktadır. Veri setine dair bir örnek Şekil 25’de paylaşılmıştır.

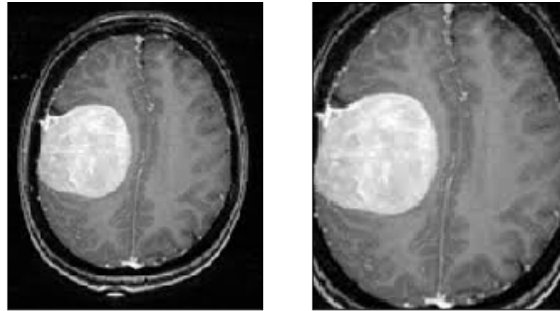


Şekil 25 Veri Seti Örneği

B. Yöntem

1. Veri Ön İşlemi

REMBRANDT veri seti dokümantasyonunda 130 hastadaki tümörlerin 110.020 MR görüntüsü olduğu belirtilmiş olsa da, bazı hasta kimliklerinin üst veride bulunmadığı ve bazı resimlerin kullanılamayacak düzeyde olması nedeniyle aykırı veri olarak değerlendirilmiş ve bu görüntüleri veri setinden kaldırılmıştır. 106.541 görüntü işlenmek üzere sınıflandırılmıştır. Resimlerde Şekil 26'deki gibi ön işlem uygulanmıştır. Şekil incelendiğinde, görüntünün sadece beynini içeren kısmı kırpma için, beynin aşırı üst, alt, sol ve sağ noktalarını bulmak için bir kontur algılama algoritması (Rosebrock, 2016) kullanıldığı görülmektedir. Yapılan bu ön işlem hesaplama karmaşıklığını azaltarak derin sinir ağlarının sadece istenilen yere odaklanmasını sağlamaktadır.



Şekil 26 Ön İşlem Yapılmış Resim Örneği

MR görüntüleri için standart format, hastanın kimliğine göre düzenlenen üst veriye sahip DICOM formatıdır. Bu görüntüler hesaplama boyutunu azaltmak için standart PNG formatına dönüştürülmüş ve tümör türlerine göre kategorilere ayrılmıştır. Kategorilere ayrılan resimler nesne serileştirme yöntemiyle kaydedilmiştir. Ön işlem sonunda kullanılan veri seti rakamları Çizelge 1’de gösterilmiştir.

Çizelge 1 Veri Seti Rakamları

Tümör Tipi	Görüntü Sayısı
Oligodendroglioma	15.293
Glioblastoma	32.837
Astrocytoma	41.324
Tanımlanamayan	17.087

Ön işlem aşamasında, evrişim katmanındaki filtreler daha sonra MR görüntüleri üzerindeki piksellere uygulanacağından, filtrenin çıktısı, MR resminin yoğunluk değerlerine bağlıdır. Bu piksel yoğunluğu tüm MR görüntülerinde aynı değildir ve görüntüler arasında farklılık gösterir. Bu görüntülerin yoğunluğu da MR makinasında farklılıklar gösterebilmekle birlikte görüntü elde etme ortamına bağlıdır. Bu sebeple her bir görüntü normalleştirilmelidir. Normalleştirmenin amacı, CNN modelinde farklı girdiler için aynı değer aralığını elde etmektir. Yapılan bu ön işlem, modelin istikrarlı yakınsamasına yardımcı olmaktadır. Bu nedenle ön işlem aşamasında minimum-maksimum normalizasyon yaklaşımı kullanılarak yoğunluk normalizasyonu sağlanmıştır. Sonuçta, görüntü pikseli değerleri [0,1] olarak ölçeklendirilmiştir. Bu durum matematiksel olarak Denklem 26 gibi ifade edilebilir:

$$y_i = (x_i - \min(x)) / (\max(x) - \min(x))$$

Denklem 26

y_i Normalleştirilmiş değerdir. Herhangi bir x_i ($i = 1... n$) değeri için görüntüdeki minimum $\min(x)$ ve maksimum $\max(x)$ değeri bulunarak normalleştirilmiş değer hesaplanır. Boyutlu girdi modeli için ön koşula göre üç kanal oluşturmak için üç kopyalama işlemi gerçekleştirilmiştir. Yoğunluk normalizasyonunun sonucu, tüm MR görüntülerinde tutarlı bir aralıkta olan yoğunluğu oluşturmaktır. Bu, CNN'nin öğrenme sürecine yardımcı olur. Görüntüler normalleştirildikten sonra 64 x 64 olarak yeniden boyutlandırılmıştır. Görüntü boyutunu küçültmek, bellek gereksinimlerini azaltırken eğitim sürecinin hızını artırmaktadır. Deneyler 13 GB belleğe sahip Google COLAB-NVIDIA GPU Tesla K80 makinasında gerçekleştirilmiştir.

2. Önerilen Yöntem

Derin sinir ağları eğitim süresince, geleneksel geriye yayılım metodunda sabit öğrenme katsayısı ve momentum, parametreleri ayarlayarak hızlı yakınsamayı kolaylaştırır. Küçük değerler ağıdaki değişimleri yavaşlatır, hızlı değerler ise salınımlara yol açar. Eğitim süresince doğru momentum değerini belirlemek hızlı ve doğru yakınsamayı elde etmek için gereklidir. Tez çalışmasında önerilen adaptif momentum şu şekilde ifade edilebilir:

$$\alpha(n) = \frac{\beta}{1 + \exp(-|1 + \sqrt{E(n) \times E(n-1)}|)}$$

Denklem 27

β , ($0 \ll \beta < 1$) aralığında hiperparametredir. Başlangıç değeri genellikle 1'e yakın olacaktır. Böylece $\alpha(n)$ başlangıç değeri büyük olacaktır. Güncellenen ağırlıkların hızlı bir şekilde yakınsamasının, momentum değeri küçüldükçe daha da artacak olan minimum sayıda yineleme yoluyla elde edilmesi beklenmektedir. Momentum, $E(n)$ hata oranının her bir eğitim sürecinde (epokta) bellekte saklar 0 ile 1 arasında azalır veya artar. Ayrıca her parametrede momentum değerini depolamak için bir hız (V) değişkeni yaratılmaktadır.

Hata fonksiyonunun w ve V 'ye göre gradyanı ve w_0, w_1, v_0, v_1 , başlangıç ağırlıkları verildiğinde, momentum algoritması w ve v ağırlıklarını yinelemeli olarak şu şekilde günceller;

$$\begin{aligned} W_{n+1} &= W_n - \eta E_W(W_n) + \alpha(W_n - W_{n-1}) \\ W_{n+1} &= w_n - \eta \sum_{j=1}^J f'_j(w_n \cdot F(V\delta_j)) F(V\delta_j) + \alpha(w_n - w_{n-1}) \\ v_{n+1}^i &= v_n^i - \eta \sum_{j=1}^J f'_j(w_n \cdot F(V\delta_j)) w_n^i f'(v_n^i \delta_j) \delta_j + \alpha(v_n^i - v_{n-1}^i), \\ i &= 1, \dots, N, \quad n = 1, 2, \dots \end{aligned}$$

Denklem 28

$\alpha \in (0,1)$, Denklem 27'de gösterildiği üzere adaptif momentum değeridir. $\eta \in (0,1)$ ise öğrenme katsayısıdır (0.01).

$$E_w(w_n, V_n) \equiv \sum_{j=1}^J f'_j(w_n \cdot F(V_{\xi_j})) F(V_{\xi_j})$$

$$E_{v_i}(w_n, V_n) \equiv \sum_{j=1}^J f'_j(w_n \cdot F(V_{\xi_j})) w_n^i f'(v_n^i \xi_j) \xi_j,$$

$$i = 1, \dots, N$$

Denklem 29

Denklem 28 şu şekilde güncellenebilir:

$$\begin{cases} \Delta w_{n+1} = \alpha \Delta w_n - \eta E_w(w_n, V_n) \\ \Delta v_{n+1}^i = \alpha_{n,i} \Delta v_n^i - \eta E_{v_i}(w_n, V_n) \end{cases} \quad i = 1, \dots, N, \quad n = 1, 2, \dots$$

Denklem 30

Uyarlanabilir momentum algoritmasının yakınsama sürecinin, aşağıdaki varsayımlar altında zayıf yakınsak olduğu söylenebilir:

- Gizli katmanın alt küme fonksiyonu $f(t)$, türevi $f'(t)$, ve $f''(t)$ tüm $t \in \mathcal{R}$ için düzgün sınırlıdır.
- $\|W_n\|$ ($n = 1, 2, \dots$) düzgün sınırlıdır
- Aşağıdaki küme sonlu sayıda elemana sahiptir.

$$\varphi = \{(w, V) | E_w(w, V) = 0, E_{v_i}(w, V) = 0, i = 1, \dots, N\}$$

Denklem 31

Denklem 29'da verilen hata fonksiyonunun ve Denklem 30 tarafından W_0 başlangıç ağırlık değeriyle üretilen $\{W_n\}$ ağırlık dizisinin, (a), (b) ve (c) varsayımlarının kullanılmasının son ağ çıkışı için geçerli olacağını varsayarsak:

- $E(W_{n+1}) \leq E(W_n), \quad n = 0, 1, \dots$
- $E^* \geq 0, \quad \lim_{n \rightarrow \infty} E(w_n, V_n) = E^*$
- $\lim_{n \rightarrow \infty} \|E_w(w_n, V_n)\| = 0$
- $\lim_{n \rightarrow \infty} \|E_{v_i}(w_n, V_n)\| = 0, \quad i = 1, \dots, N$

Herhangi bir $\xi \in \mathcal{R}^1$ değeri için, gizli nöronların çıkışı değeri $F(V_{\xi})$ olacaktır, ağ çıkışı şu şekilde ifade edilecektir:

$$\delta = f(w \cdot F(V_{\delta}))$$

Denklem 32

Dolayısıyla (c) varsayımı doğru olursa, Denklem yerel minimuma (w^*, V^*) yakınsar:

$$\lim_{n \rightarrow \infty} w_n = w^*, \lim_{n \rightarrow \infty} V_n = V^*$$

Denklem 33

$$E_w(w^*, V^*) = 0, E_{v_i}(w^*, V^*) = 0, \quad i = 1, \dots, N$$

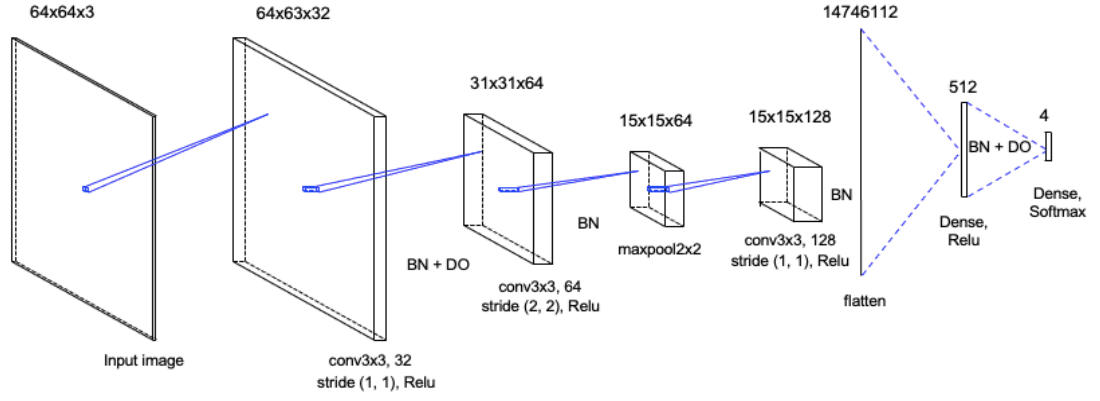
Denklem 34

Önerilen değişken momentum algoritması, momentum terimini güncellemek için hatayı kullandığından, yani momentum teriminin doğrudan hata değeriyle ilişkili olduğu ve hatayı azaltacak şekilde davrandığı için, tekniğin önceki bilinen yöntemlerinden farklı olduğu söylenebilir.

3. Derin Sinir Ağı Mimarisi ve Eğitimi

Sinir ağları, karmaşık görevleri yerine getirirken katmanlarına doğrusal olmayanlığı dâhil etmelidir. Aktivasyon fonksiyonu, her katmanın çıktısını hesaplar. Sigmoid gibi fonksiyonların türevinin zor alınmasından dolayı hesaplama karmaşıklığı daha fazladır. Relu ise daha az hesaplama ile doğrudan gradyanı verir. Bu, geri yayılım sırasında çok fazla zaman ve gradyan hesaplama gücü tasarrufu sağlar. Relu, doygunluk sağlamayan $x > 0$ için doğrusal bir fonksiyona sahiptir.

Evrişimli ve tam bağlantılı katmanlardaki tüm nöronlar, girdiyi hesaplamak ve çıktı üretmek için optimizasyon yöntemi kullanır. Bu süreçte ise havuzlama katmanı yerine adım eklemenin doğruluğu artırdığı iyi bir şekilde başarılmıştır. Ancak, ağ parametrelerini azaltmak için son 2 evrişim katmanı arasında da maksimum havuzlama katmanı kullanılması oldukça yaygındır. Tezde kullanılan modelin görseli Şekil 27'deki gösterilmiştir. Son katman, Softmax aktivasyonunu kullanarak her sınıflandırma olasılığını hesaplar:



Şekil 27 Derin Ağ Mimarisi (CNN)

Geliştirilen modelde hata fonksiyonu olarak çapraz entropi kullanılmıştır. Model katmanları, kapsamlı bir dizi deney yoluyla ayarlanmıştır. Öncelikle sıralı nesneler yaratılmış ve tek bir CNN katmanı ile başlanmıştır. Her 3 CNN katmanı için filtreler sırasıyla 32, 64 ve 128 olarak ayarlanmıştır. Aynı dolgu ile çekirdek boyutunu 3'e 3 olarak ayarlanmıştır. Sadece ikinci CNN katmanında 2'ye 2 adım kullanılmıştır. Diğer katmanda, adım parametreleri varsayılan olarak 1 ve 2x2 filtre boyutunda ve ortak maksimum havuz katmanı uygulanmıştır. Tam bağlantı katmanında 512 nöron eklenerek model tamamlanmıştır.

Çizelge 2 Önerilen Derin Sinir Ağı Mimarisi

Katman	Çıkış Boyutu	Toplam Parametre Sayısı
Evrişim	64 x 64 x 32	896
Yığın Normalleştirme	64 x 64 x 32	128
Seyreltme	64 x 64 x 32	0
Evrişim	31 x 31 x 64	18496
Yığın Normalleştirme	31 x 31 x 64	256
Maksimum Havuzlama	15 x 15 x 128	0
Evrişim	15 x 15 x 128	73586
Yığın Normalleştirme	15 x 15 x 128	512
Düzleştirme	28800	0
Tam Bağlantı	512	14746112
Yığın Normalleştirme	512	2048
Seyreltme	512	0
Çıkış Katmanı	4	2052

Önerilen modelin özeti Çizelge 2’de gösterilmektedir; görüldüğü üzere öğrenme sürecini stabilize etmek için bırakma ve toplu normalleştirme (BN) kullanılmıştır. Dropout (DO) için 0,3 değeriyle başlanmış ve tam bağlantılı katmandan önce 0,5 değeri kullanılmıştır. Maliyet fonksiyonunda ağırlıkların kare değerlerini ekleyen L2 düzenlemesi de (ağırlık azalması) kullanılmıştır. Burada yapılan ağırlık azaltması ile eğitim sürecinin doğruluğunun daha da iyileşmesi, ağırlıkları dağıtarak test performansının daha iyi olması amaçlanmıştır. Özel model düzenleme hiperparametresi 0.0005’tir. Model çıktı şekilleri, görüntülerin giriş şekillerinden dolayı her veri kümesinde farklılık göstermektedir. Özel bir model oluşturma ve kullanmanın amacı eğitim süresini azaltmaktır, ancak deneyler sırasında önerilen yöntem daha sonraki bölümlerde tartışılacak olan önceden eğitilmiş derin sinir ağı modelleriyle de test edilmiştir.

4. Tıbbi Teşhis Performans Metrikleri

Makine öğrenmesinde kullanılan sınıflandırma modellerinin performansını genellikle hata matrisiyle ölçülmektedir. Sınıflandırma tahminleri aşağıdaki değerlendirmeden birine sahip olacaktır:

Çizelge 3 Performans Metrikleri

Tahmin Edilen	Gerçekte Olan	Tanım
Pozitif	Pozitif	Doğru Pozitif Tahmin (TP)
Pozitif	Negatif	Yanlışlıkla Pozitif Tahmin (FP)
Negatif	Negatif	Doğru Negatif Tahmin (TN)
Negatif	Pozitif	Yanlışlıkla Negatif Tahmin (FN)

Derin sinir ağlarında doğruluk üzerinden modelin başarısı ölçülse de sınıflandırma problemlerinde doğruluk, kesinlik ve f1 değeri gibi diğer metriklerin de ölçülmesi gerekmektedir. Kesinlik ve duyarlılık değeri, verilerin dağılımların eşit olmadığı durumlarda başarılı bir performans metriğidir. Tıbbi teşhis alanında veriler genellikle dengesizdir. Kesinlik (Precision), pozitif olarak tahminlediğimiz değerlerin gerçekten kaç adedinin pozitif olduğunu göstermektedir. Duyarlılık (Recall) ise pozitif olarak tahmin etmemiz gereken işlemlerin ne kadarını pozitif olarak tahmin ettiğimizi gösteren bir metriktir. Duyarlılık değeri özellikle tıbbi teşhis alanında önemlidir. Duyarlılık değeri de FN olarak tahminlemenin maliyetinin

yüksek olduğu durumlarda bize yardımcı olacak bir metriktir. Mümkün olduğunca yüksek olması gereklidir. Örneğin tümör kanser tahmini için oluşturulan bir modelin kanser olan bir kişiyi kanser değil olarak sınıflandırması büyük sorun teşkil edecektir. F1 değeri ise kesinlik ve duyarlılık değerlerinin harmonik ortalamasını göstermektedir. Basit bir ortalama yerine harmonik ortalama olmasının sebebi ise uç durumları da gözardı edilmesi gerektiğidir. Eğer basit bir ortalama hesaplaması olsaydı kesinlik değeri 1 ve duyarlılık değeri 0 olan bir modelin F1 değeri 0,5 olarak gelecektir ve bu bizi yanıltacaktır. Doğruluk (Accuracy) yerine F1 değerinin kullanılmasının en temel sebebi eşit dağılmayan veri kümelerinde hatalı bir model seçimi yapmamaktır. Ayrıca sadece FN ya da FP değil tüm hata maliyetlerini de içerecek bir ölçme metriğine ihtiyaç duyulduğu için de F1 değeri çok önemlidir. Performans metriklerinin formülasyonları aşağıdaki gibidir.

$$\text{Doğruluk (Accuracy)} = \frac{TP+TN}{TP+FP+FN+TN}$$

$$\text{Kesinlik (Precision)} = \frac{TP}{TP+FP}$$

$$\text{Duyarlılık (Recall)} = \frac{TP}{TP+FN}$$

$$\text{F1 Değeri} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$



IV. BULGULAR

Önerilen yöntemin sınıflandırma performansını değerlendirmek için beyin tümörü veri seti %25'lik ve %75'lik oranlarda iki parçaya ayrılmıştır. %25'lik veri seti test seti olarak belirlenmiş ve %75'lik kısım önerilen modeli eğitmek için kullanılmıştır. Eğitim Scikit-Learn ve Tensorflow gibi çeşitli Python kütüphaneleri kullanılarak gerçekleştirilmiştir. Tasarlanan modelin katmanlarının modellenmesi Tensorflow aracılığıyla yapılmıştır. Önerilen yöntem Adam, RMSprop ve SGD (momentum değerine sahip) gibi adaptif optimizasyon algoritmaları ile karşılaştırılmıştır. Yöntemin yakınsama hızı ve başarı oranı test edilirken başarı oranı (accuracy) 5 epokta ve hata oranı (loss) ise 20 epokta değerlendirilmiştir. Farklı optimizasyon algoritmalarının karşılaştırılması ile yapılan sınıflandırma sonuçları Çizelge 4 ve Çizelge 5'te gösterilmektedir. Önerilen adaptif momentum, Adam, RMSprop ve SGD'den her iki metriktede daha başarılıdır.

Çizelge 4 Diğer Yöntemler Başarı Oranı Karşılaştırması

Epok	Adam	RMSProp	SGD	Önerilen Yöntem
1	0,72	0,70	0,70	0,73
2	0,79	0,76	0,77	0,83
3	0,81	0,79	0,82	0,88
4	0,81	0,80	0,85	0,90
5	0,83	0,80	0,88	0,91

Çizelge 5 Diğer Yöntemlerle Kayıp Değeri Karşılaştırması

Epok	Adam	RMSProp	SGD	Önerilen Yöntem
1	1,68	1,16	1,36	0,96
5	1,23	0,78	0,85	0,68
10	1,07	0,78	0,72	0,61
15	0,93	0,68	0,64	0,60
20	0,92	0,67	0,61	0,54

Çizelge 6 incelendiğinde, geliştirilen modelin çoklu sınıflandırmada %95 F1 puanı elde ettiği görülmektedir. Sınıfların eşit dağılmamasına rağmen tasarlanan modelin kesinlik ve duyarlılık değerinin ağırlıklı ve makro ortalamaları sırasıyla %95 ve %94'tür. Çizelge 4 ile karşılaştırıldığında tasarlanan modelin performansının 20 epok sonunda iyileştirildiği görülmektedir.

Çizelge 6 Önerilen Yöntemin Sınıflandırma Sonuçları

Tümör Tipi	Kesinlik	Duyarlılık	F1 Değeri
Oligodendroglioma	0,94	0,97	0,95
Glioblastoma	0,95	0,94	0,94
Astrocytoma	0,94	0,90	0,92
Tanımlanamayan	0,97	0,95	0,96
Doğruluk			0,95

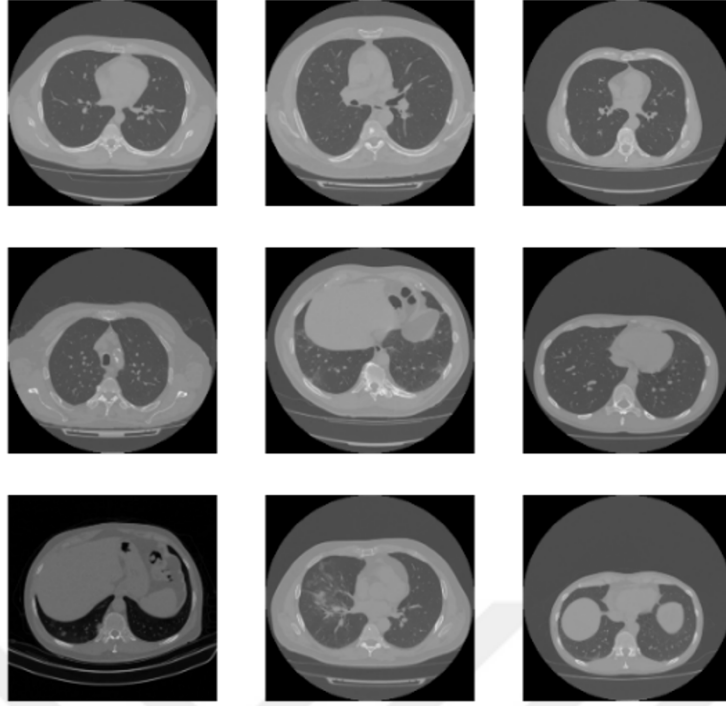
CNN'ler eğitim esnasında milyonlarca benzer görüntüyü analiz ederek beyin tümörü görüntülerinin sınıflandırılması konusunda daha iyi ve doğru bir yol sunmaktadır. Bu tezin kapsamı ise MR görüntülemeye dayanan beyin tümör tiplerini sınıflandırabilecek adaptif momentum ile daha basit bir CNN modeli geliştirmek olsa da çalışmanın sonuçlarını güçlendirmek için önerilen yöntemin derin CNN mimarilerindeki performans karşılaştırması araştırılmıştır. Daha küçük veri setleri için transfer öğrenimi başarılı olsa da (Afshary et al., 2018) büyük veri setleriyle yapılan denemeler hala zaman alabilmektedir. Tez kapsamında, Imagenette eğitilmiş önceden eğitilmiş 3 derin CNN modelinin (Xception, Resnet50 ve VGG16) sınıflandırma performansı karşılaştırılmıştır (Bhattacharjee & Pant, 2019). Xception

Xception Modeli'nin 71 derin katmanı vardır ve Francois Chollet tarafından tasarlanmıştır (Jothi & HannahInbarani, 2016). Resnet50 tezde bahsedildiği gibi 50 derin katmana sahiptir (Tiwari et al., 2020) ve VGG16 (Oxfordnet) olarak da bilinir. Tang ve arkadaşları tarafından önerilen 16 katman derinliğine sahip evrimsel bir sinir ağıdır (Tang et al., 2019). Çizelge 7'de görüldüğü üzere deney sonuçlarında önerilen adaptif momentuma sahip VGG16'nın en başarılı sonuca ulaştığı görülmektedir.

Çizelge 7 Aktarım Öğrenimi Model Karşılaştırması

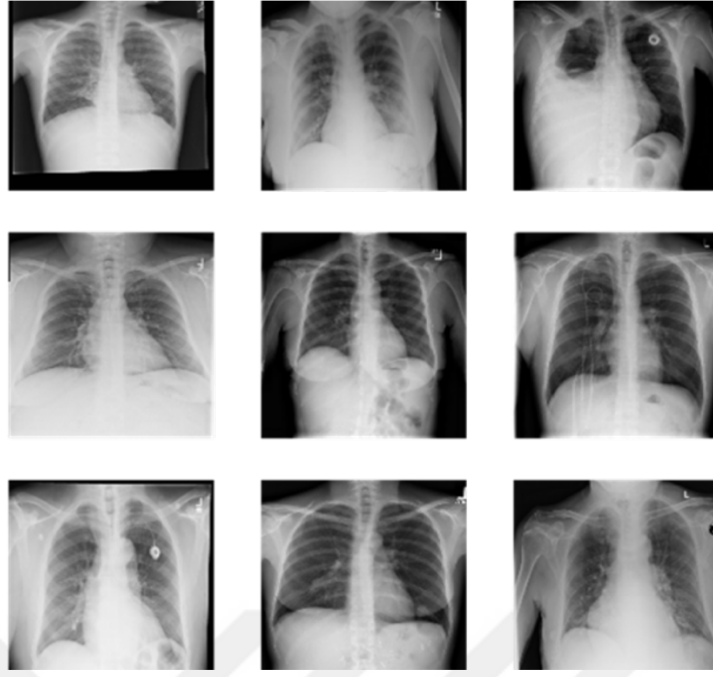
Epok	Xception	Resnet50	VGG16
5	0,84	0,85	0,92

Makine öğrenimi algoritmaları farklı veri setlerinde farklı sonuçlar verebildiğinden yöntemin birden çok veri seti ile valide edilmesi çok önemlidir. Bu nedenle tezin kapsamındaki deneyin sonuçları, diğer tıbbi veri setleriye de test edilmiştir. Pandemi sürecinde Maftouni ve arkadaşları (2021), 7 farklı veri setinden gelen verileri derleyerek Kaggle'da 8.2 kullanılabilirlik seviyesi ile şimdiye kadarki en büyük COVID-19 akciğer BT veri setini (Şekil 28) oluşturmuşlardır (Maftouni et al., 2021). Bu veri setleri COVID-19 tanı literatüründe herkese açık olarak kullanılmış ve derin öğrenme uygulamalarında etkinliklerini kanıtlamıştır. Dolayısıyla birleştirilmiş veri setinin tüm bu kaynaklardan bir arada öğrenerek derin öğrenme yöntemlerinin genelleme yetisini geliştirmesi beklenmektedir. Veri setinde örtüşen meta verileriyle birlikte COVID-19, Normal ve CAP, CT dilimleri vardır. Veri setlerinin bazıları kategorize edilmiş CT dilimlerinden oluşurken, bazıları da açıklamalı lezyon dilimlerine sahip CT hacimlerini içerir. Bu sebeple CT hacimlerinden eksenele dilimleri çıkarmak için dilim düzeyindeki açıklamaları kullanılmıştır. Tutarlı bir veri derinliği için tüm görüntüler 8 bit'e dönüştürülmüştür. Veri seti kalitesini sağlamak için akciğer içi belirtiler hakkında bilgi taşımayan kapalı akciğer normal dilimler çıkarılmıştır. Ayrıca net sınıf etiketleri veya hasta bilgileri içermeyen görüntüler de veri setinden çıkarılmıştır. Toplamda 466 hastadan 7.593 COVID-19 görüntüsü, 604 hastadan 6.893 normal görüntü ve 60 hastadan 2.618 CAP görüntüsü toplanmıştır. Tıbbi görüntülemelerde tüm sınıflandırma tiplerini test ederken ikili sınıflandırma performansı için bu çalışmada CAP etiketleri kaldırılmıştır.



Şekil 28 Covid-19 Görüntüleri (BT)

Göğüs röntgeni muayenesi en sık başvuru ve en uygun maliyeti olan tıbbi görüntüleme tetkiklerinden biridir. Fakat göğüs röntgenlerinin klinik teşhisi zor olabilir ve bazen göğüs BT görüntülemesi ile teşhis koymaktan daha da zor olabilmektedir. NIH Göğüs Röntgeni Veri Seti (Şekil 29) 30.805 farklı hastadan çoklu hastalık etiketli 112.120 X-ışını görüntüsünden oluşmaktadır. İlk veri çerçevesi CNN eğitim amaçları için daha uygun biçimlendirmek (TFRecords) üzere önceden işlenmiştir. Yalnızca görüntüler (600x600'e küçültülmüş ve 1 kanallı jpeg'ler olarak kodlanmış) ve ilgili tanı, tüm ek hasta bilgileri (örneğin; yaş, cinsiyet vb.) çıkartılarak bırakılmıştır. 112.120 örneğin tamamı alınmış olup filtreleme, gruplama veya çıkarma yapılmamıştır. Çalışmada atelettazi, konsolidasyon, infiltrasyon, pnömotoraks, ödem, amfizem, fibrozis, efüzyon, pnömoni, plevral kalınlaşma, kardiyomegali, nodül, kitle dâhil tüm kategorilere odaklanılmıştır.



Şekil 29 Göğüs Röntgen Görüntüleri (X-Ray)

Göğüs röntgeni ve Covid-19 görüntüleri %20-80 oranında eğitim ve test veri seti olarak rastgele örnekleme ile bölünmüştür. Eğitim, beyin tümörü veri setinde olduğu gibi Scikit-learn ve TensorFlow gibi çeşitli Python kütüphaneleri kullanılarak gerçekleştirilmiştir. Yineleme başına hafıza tüketimini sınırlamak için görüntülerin boyutu küçülmüştür. Önerilen yöntem, aynı şekilde Adam, RMSprop ve SGD(momentum) ile test edilmiştir. Adam için standart parametreler ($\beta_1 = 0.9$, $\beta_2 = 0.999$, ve $\epsilon=1e-07$), RMSProp için öğrenme oranı $\alpha= 0.001$, SGD için ise $\alpha= 0.01$, $m=0.9$) kullanılmıştır. Farklı veri setleri ile yapılan önerilen yöntemin validasyon sonuçları Çizelge 8 ve Çizelge 9’da gösterilmektedir.

Çizelge 8 Göğüs X-Ray Veri Seti Başarı Oranı Karşılaştırması

Epok	Adam	RMSProp	SGD	Önerilen Yöntem
1	0,83	0,80	0,84	0,82
2	0,84	0,76	0,83	0,83
3	0,84	0,84	0,84	0,84
4	0,83	0,79	0,83	0,85
5	0,79	0,83	0,84	0,85

Çizelge 9 Covid-19 Veri Seti Başarı Oranı Karşılaştırması

Epok	Adam	RMSProp	SGD	Önerilen Yöntem
1	0,85	0,58	0,90	0,87
2	0,89	0,93	0,91	0,90
3	0,84	0,93	0,93	0,92
4	0,92	0,82	0,94	0,82
5	0,93	0,48	0,93	0,92

Araştırma sonuçları, beyin tümörü ve göğüs röntgeni veri setinde adaptif momentumların Adam, RMSprop ve SGD'den daha iyi yakınsadığını göstermiştir. Çizelge 10'de görülebileceği gibi tasarlanan model çoklu sınıflandırma problemi için %95, çoklu etiketli sınıflandırma için %85 ve ikili sınıflandırma için %93 F1 puanı almıştır. Beyin görüntüsü ve göğüs röntgeni veri setlerinde eşit olmayan bir sınıf dağılımına rağmen kesinlik ve duyarlılık puanlarının ağırlıklı ve makro ortalamalarının başarılı olduğu görülmektedir.

Çizelge 10 Diğer Veri Setleriyle Validasyon Sınıflandırma Sonuçları

Veri Tipi	Kesinlik	Duyarlılık	F1 Değeri
REMBRANDT	0,94	0,97	0,95
NIH Chest X-Ray	0,83	0,84	0,85
Covid-19	0,94	0,92	0,93

Çizelge 11, tezde kullanılan veri setleriyle literatürde yapılan diğer çalışmaların karşılaştırmasını göstermektedir. (Bhattacharjee & Pant, 2019) çalışmalarında %62 doğruluk başarısı ile hibrit genetik algoritma ve parçacık sürü optimizasyonu (PSO) kullanırken, (Jothi ve HannahInbarani, 2016) ateşböceği algoritması ve toleranslı kaba set kümelemesini (TRS) kullanmış ve %90 doğruluk elde etmiştir. Tüm sonuçlar Tiwari ve arkadaşlarının orijinal çalışmalarından alınmıştır (Tiwari et al., 2020). Önerilen yöntem %95 ile en yüksek sınıflandırma doğruluğuna ulaşmıştır. Şunu belirtmek isteriz ki, Covid-19 çalışmalarıyla karşılaştırıldığında (Maftouni et al., 2021) çalışmalarında ağırları 50 epok boyunca eğitirken, bu çalışmada sadece 5 epok eğiterek 10 kat daha hızlı yakınsadığı görülmüştür.

Çizelge 11 Aynı Veri Seti Üzerinde Yapılan Diğer Çalışmalar ile Karşılaştırma

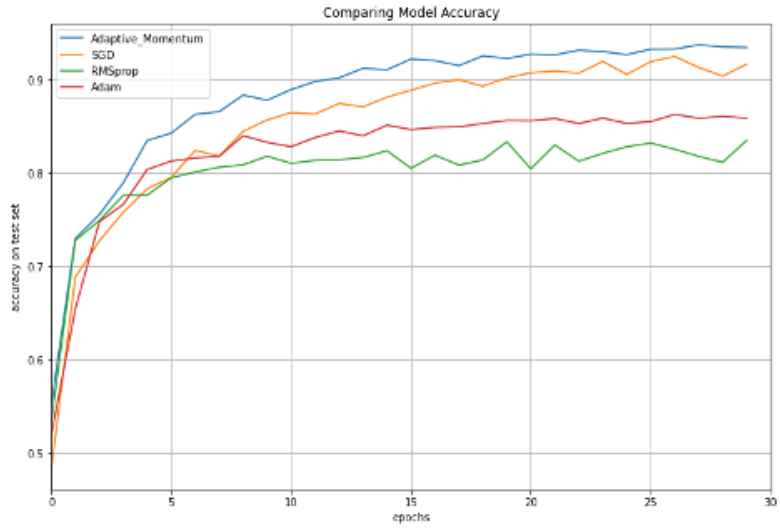
REMBRANDT	HGAPSO (Bhattacharjee et al., 2019)	FATRS (Jothi & Hannahlbnarani, 2016)	Önerilen Yöntem
Doğruluk	0,62	0,90	0,95
NIH Chest X-Ray	GAC (Tang et al., 2019)	DNT (Antin et al., 2017)	Önerilen Yöntem
Doğruluk	0,84	0,60	0,85
Covid-19	FC (Maftouni et al., 2021)	DN (Maftouni et al., 2021)	Önerilen Yöntem
Doğruluk	0,95	0,92	0,93

Çizelge 12 Aktarım Öğrenimi Model Karşılaştırması

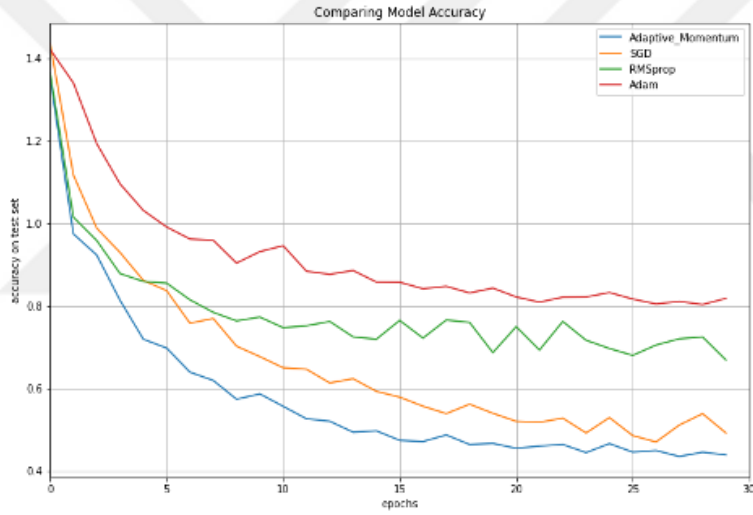
Veri Seti	Xception	Resnet50	VGG16	Önerilen Yöntem
REMBRANDT	0,87	0,91	0,94	0.95
NIH Chest X-Ray	0,83	0,84	0,88	0.85
Covid 19	0,96	0,98	0,52	0.93

Doğrudan hata varyasyonu ile ilgili olan adaptif momentum algoritması konvansiyonel optimize edicilerle kıyaslandığında minimuma çok daha hızlı yakınsar. Çizelge 12, farklı veri setleriyle, önerilen modelin önceden eğitilmiş modeller arasındaki karşılaştırma sonuçlarını göstermektedir. REMBRANDT veri setinde VGG16 daha iyi sonuç verirken, Covid verilerinde Xception, göğüs hastalıkları verisinde ise Resnet50 başarılı sonuç vermiştir. Sonuçlar beş epok sonrası elde edilmiştir.

Deneyler sırasında CNN modeline ince ayar yapılmıştır. Tasarlanan yöntemin yakınsama hızı ve ince ayar işleminin modelin performansını arttırdığı Şekil 30 ve Şekil 31’de görülmektedir.



Şekil 30 Model (İnce Ayarlı) Doğruluk Grafiği



Şekil 31 Model (İnce Ayarlı) Hata Grafiği

V. SONUÇ VE ÖNERİLER

Derin öğrenme algoritmaları, başarılı sınıflandırma performanslarıyla makine öğrenme alanında popülerlik kazanmışlardır. Derin öğrenme, özellik mühendisliğine (veri bilimi) daha az bağlı olan daha iyi öğrenme algoritmalarını ve temsillerini keşfetmek amacıyla, birçok veri işleme katmanı kullanarak özet kavramları açıklamaktadır. Bu tezde derin öğrenme yöntemleri geniş bir bakış açısıyla değerlendirilmiş, aynı zamanda adaptif teriminin doğrudan hata varyasyonu ile ilişkili olduğu uyarlanabilir bir derin öğrenme algoritması önerilmiştir. Önerilen algoritma ağırlık vektörünü girdi vektörüne göre güncellemektedir. Bu nedenle algoritma girdinin otokorelasyon matrisinin özdeğerlerine bağlı olan öğrenme oranı parametresi tarafından kontrol edilir. Bu da geleneksel adaptif optimizasyon yöntemleri ile karşılaştırıldığında daha başarılı sonuçlar vermiştir. Sonuçlar, önerilen yöntemin optimum çözümde daha hızlı yakınsama davranışı sergilediğini ve hata fonksiyonu değerini en aza indirdiğini göstermektedir.

Tasarlanan yöntem ortalama sınıflandırma hatasını %6,12'den %5,44'e düşürmüş, yakınsama hızını diğer optimizasyon yöntemlerine kıyasla %20 arttırmış ve diğer yöntemlere kıyasla en düşük hata ve en yüksek doğruluğa ulaşmıştır. Test sonuçları algoritmanın beyin tümörü, göğüs röntgeni ve Covid-19 veri setlerine karşı sırasıyla %95, %85 ve %93 başarı oranı sağladığını göstermektedir. Özel adaptif momentuma sahip önceden eğitilmiş modeller de deneyler sırasında test edilmiştir. Covid-19 veri seti için %87 doğruluk (Maftouni et al., 2021) sağlayan Xception ile karşılaştırıldığında tasarlanan yöntem %97 doğruluk elde etmiş olsa da VGG16 ve Resnet50'nin doğruluğu değişmemiştir. Test sonuçları hem VGG16 hem de Resnet50'nin tasarlanan yöntemden daha iyi performansa sahip olduğunu göstermektedir. Tasarlanan yöntemin eğitim veri seti boyutunun ve epok sayısının önceden eğitilmiş modellere göre küçük olduğu göz önüne alındığında, bu iki önceden eğitilmiş modele karşı çok iyi performans gösterdiği söylenebilir. Zaten önceden eğitilmiş modeller böyle bir sonucu elde etmek için milyonlarca görüntü üzerinde eğitildiğinden bu tür bir karşılaştırma adil bir karşılaştırma olarak kabul edilmeyecektir. Önceden eğitilmiş en iyi modellerle (Xception, ResNet50 ve

VGG16) karşılaştırıldığında %60 hızlı yakınsama ile göğüs röntgeni ve beyin tümörü verisinde literatürdeki en iyi sonuçlara ulaşılmıştır. Test ve validasyon için seçilen veri setlerinin tümü medikal görüntülerin doğası gereği oldukça dengesiz veri seti olduğundan algoritmanın dengesiz veri setlerine karşı iyi performans sergileyebileceğini kesinlik ve duyarlılık değerleriyle de doğrulanmıştır.

Modelleri eğitirken tıbbi görüntülerin kullanılmasının önünde üç temel engel vardır. Bunlar: sınıf dengesizliği sorunu, çoklu boyut sorunu ve veri seti boyutu sorunudur. Bu engelleri aşmak için başvurulabilecek çeşitli tekniklere bu çalışmada yer verilmiştir. Sınıf dengesizliği problemi için farklı sınıf ağırlıkları değerleri kullanılmış ve CNN modeli eğitilmiştir. Tıbbi görüntü sınıflandırma probleminde karşılaşılan bir diğer güçlük çoklu sınıflandırma probleminin zorluğudur. Çoklu sınıflandırma problemlerinde sıkça karşılaşılan, modelin eğitim sürecinde başarılı sonuca ulaşamama probleminin üstesinden gelmek için seyreltme ve yığın normalleştirme teknikleri kullanılmıştır. Modeli eğitmek için Google Colab'da yüksek RAM ile TPU/GPU kullanılmıştır. Bu durum hesaplama karmaşıklığını azaltmaya yardımcı olmuştur. DL algoritmaları, CPU'lar yerine GPU'lar kullanıldığında, grafik işleme birimlerinden (GPU) 100 kata kadar daha hızlı çalışmaktadırlar (Erickson et al., 2018). Tezde hesaplama karmaşıklığı probleminin üstesinden GPU sayesinde gelinmiştir. GPU tabanlı çözümler performans iyileşmesi sağlamalarına rağmen aynı zamanda donanım maliyetini artırmaktadırlar.

Evrişimsel sinir ağı birçok tıbbi görüntüleme probleminde varsayılan mimari olarak kullanılmaktadır. Bunlar x-ışınları gibi 2 boyutlu görüntüleri işlemek için tasarlanmıştır. Ancak varyantları gelecekteki çalışmalarda incelenebilecek CT taramaları gibi tıbbi sinyali işleme veya 3D tıbbi görüntüler için kullanımda da oldukça uygundur. Standart olan tıbbi sorunlarda istenen görevlerde birden fazla model denemek ve hangilerinin en iyi sonucu verdiğini görmektir. Önceden eğitilmiş modellerin eğitilmesi gereken yaklaşık 20 kat daha fazla parametre vardır. Çalışma sırasında parametre optimizasyonunun GPU'lar ve TPU'lar için bile zaman alıcı olduğu görülmüştür.

Tez kapsamında özel CNN modeli kullanılmasında amaç yapılacak ince ayarlamaların modelin özelliklerini ve çıktılarını nasıl değiştirdiğini görmektir. Bu sebeple başka içerikleri tespit etmek ve yerelleştirmek için öznetelik gösteriminde kullanılabilecek bir modelin genellemesi hedeflenmiştir.

Geliştirilen CNN modeli yalnızca 2D görüntüler ile çalışmaktadır ancak model 3D

görüntülerin analizi için de uyarlanabilir. Ama bu durumda görüntüler büyük olacak ve sonuçta ortaya çıkan eğitim verisi setleri ciddi bir hesaplama gücü gerektirecektir. Gelecekte kayıp değere bağlı olarak öğrenme oranını adaptif hale getirmek gibi çalışmalar gerçekleştirilebilecektir. Her epokta yeni öğrenme oranını çıktı olarak döndüren bir fonksiyon tanımlanabilecektir. Bu fonksiyon kayıp değerleri ile kontrol edilecek ve belirtilen aralık içinde artacak veya azalacaktır. Çalışma sonucunda önerilen yöntem ile beyin tümörü ve diğer hastalık tipleri kolayca belirlenebilecektir.





VI. KAYNAKÇA

KİTAPLAR

GONZALEZ, R. C., WOODS, R. E. (2001) **“Digital Image Processing”**, Person Education International, 3. Edition.

HAUGELAND, J. (1985). **“Artificial intelligence: the very idea”**, The MIT Press, Cambridge, Mass.

MAKALELER

ABBASI, S., TAJERİPOUR, F. (2017). **“Detection of brain tumor in 3D MRI images using local binary patterns and histogram orientation gradient”**, Neurocomputing, 219, 526–535.

AFSHARY, P., MOHAMMADIY, A., PLATANIOTIS, K. (2018). **“Brain tumor type classification via capsule networks”**, arXiv preprint arXiv:1802.10200. 1-5. Doi:10.1109/icip.2018.8451379.

AGRAWAL, T., GUPTA, R., NARAYANAN S. (2019). **“On Evaluating CNN Representations for Low Resource Medical Image Classification”**, in ICASSP - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, United Kingdom. 1363–1367. doi: 10.1109/ICASSP.2019.8682397.

AHMADVAND, A., KABİRİ, P. (2016). **“Multispectral MRI image segmentation using Markov random field model. Signal”**, Image and Video Processing, 10, 251–258.

AHN, E., KUMAR, A., KİM, J., Lİ, C., FENG, D., FULHAM, M. (2016). **“X-ray image classification using domain transferred convolutional neural networks and local sparse spatial pyramid”**. 2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI), Prague, 855-858. <https://doi.org/10.1109/ISBI.2016.7493400>

ALZUBAIDI, L., ZHANG, J., HUMAIDI, A. J., AL-DUJAILI, A., DUAN, Y., AL-SHAMMA, SANTAMARIA, J., FADHEL, M. A., AL AMIDIE, M.,

- FARHAN, L. (2021). **“Review of deep learning: concepts, CNN architectures, challenges, applications, future directions”**, Journal of Big Data, 8, 53. doi: 10.1186/s40537-021-00444-8.
- AMIN, J., SHARIF, M., YASMIN, M., FERNANDES, S.L. (2017). **“A distinctive approach in brain tumor detection and classification using MRI”**, Pattern Recognition Letters. 139, 118-127, ISSN 0167-8655, <https://doi.org/10.1016/j.patrec.2017.10.036>.
- ANDERMATT, S., PEZOLD, S., CATTIN, P. (2016). **“Multi-dimensional gated recurrent units for the segmentation of biomedical 3D-data”**. In: Proceedings of the Deep Learning in Medical Image Analysis (DLMIA). In: Lecture Notes in Computer Science, 10008, 142–151.
- ANTIN, B., KRAVITZ, J., MARTAYAN, E. (2017) **“Detecting Pneumonia in Chest X Rays with Supervised Learning”**. <http://cs229.stanford.edu/proj2017/final-reports/5231221.pdf>
- ANTONY, J., MCGUINNESS, K., CONNOR, N.E.O., MORAN, K. (2016). **“Quantifying radio-graphic knee osteoarthritis severity using deep convolutional neural networks”**. arxiv:1609.02469.
- ARMATO, S.G., McLennan, G., BIDAUT, L., MCNITT-GRAY, M.F., MEYER, C.R., REEVES, A.P., ZHAO, B., ... CROFT, B.Y. (2011). **“The lung image database consortium (LIDC) and image database resource initiative (IDRI): a completed reference database of lung nodules on CT scans”**. Med. Phys. 38, 915–931.
- BALASOORIYA, N. M., NAWARATHNA, R. D. (2017). **“A sophisticated convolutional neural network model for brain tumor classification”**, 2017 IEEE International Conference on Industrial and Information Systems (ICIIS), Peradeniya. 1–5. doi: 10.1109/ICIINFS.2017.8300364.
- BAR, Y., DIAMANT, I., WOLF, L., GREENSPAN, H. (2015). **“Deep learning with non-medical training used for chest pathology identification”**. Proceedings of SPIE Medical Imaging, International Society for Optics and Photonics, Orlando, FL, USA, February 2015, 94140V. <https://doi.org/10.1117/12.2083124>
- BHATTACHARJEE, K., PANT, M. (2019) **“Hybrid particle swarm optimization-genetic algorithm trained multi-layer perceptron for classification of**

- human glioma from molecular brain neoplasia data".** Cognit. Syst. Res. 58, 173–194.
- BROSCH, T., TAM, R. (2013). **"Manifold learning of brain MRIs by deep learning"**. In: Proceedings of the Medical Image Computing and Computer-Assisted Intervention. In: Lecture Notes in Computer Science, 8150, 633–640. doi:10.1007/978-3-642-40763-5_78.
- CHAPLOT S., PATNAIK, L. M. & JAGANNATHAN, N. R. (2006). **"Classification of magnetic resonance brain images using wavelets as input to support vector machine and neural network"**, Biomedical Signal Processing and Control, 1(1), 86-92.
- CHEN, H., WANG, X., HENG, P.A. (2016). **"Automated mitosis detection with deep regression networks"**. In: Proceedings of the IEEE International Symposium on Biomedical Imaging, 1204–1207. doi:10.1109/ISBI.2016.7493482.
- CHENG, J., HUANG, W., CAO, S., YANG, R., YANG, W., YUN, W., Z., FENG, Q. (2015). **"Enhanced performance of brain tumor classification via tumor region augmentation and partition"**, PloS one. 10 (10). Doi: 10.1371/journal.pone.0144479
- CHENG, J.-Z., NI, D., CHOU, Y.-H., QIN, J., TIU, C.-M., CHANG, Y.-C., HUANG, C.-S., SHEN, D., CHEN, C.-M. (2016). **"Computer-aided diagnosis with deep learning architecture: applications to breast lesions in US images and pulmonary nodules in CT scans"**. Nat. Sci. Rep. 6, 24454. doi:10.1038/srep24454.
- CHOLLET, F. (2017) **"Xception: Deep Learning with Depthwise Separable Convolutions"**. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 1251-1258 Doi: 10.1109/CVPR.2017.195
- CLARK, K., VENDT, B., SMITH, K., FREYMAN, J., KIRBY, J., KOPPEL, P., MOORE, S., PHILLIPS, S., MAFFITT, D., PRINGLE, M., TARBOX, L., PRIOR, F. (2013) **"The Cancer Imaging Archive (TCIA): maintaining and operating a public information repository"**. Journal of Digital Imaging. 26(6), 1045-57 Doi: 10.1007/s10278-013-9622-7
- COUNCIL, N. C. VICTORIA, C. C. and AUSTRALIA, C. C. (2014). **"Understanding brain tumours: a guide for people with brain or spinal cord tumours and their families and friends"**. Cancer Council NSW

- Woolloomooloo, NSW, 2014 [Online].
<http://www.cancercouncil.com.au/wpcontent/uploads/2014/05/CAN879-Brain-NSW.pdf>
- ÇİÇEK, Ö., ABDULKADİR, A., LIENKAMP, S.S., BROX, T., RONNEBERGER, O., (2016). **“3D U-Net: learning dense volumetric segmentation from sparse annotation”**. In: Proceedings of the Medical Image Computing and Computer-Assisted Intervention. In: Lecture Notes in Computer Science, 9901, 424–432. 1606.06650v1
- DONG, H., YANG, G., LIU, F., MO, Y., GUO, Y. (2017). **“Automatic brain tumor detection and segmentation using U-Net based fully convolutional networks”**, Annual Conference on Medical Image Understanding and Analysis, pp. 506-517. Springer, Cham
- DUTTA, P., UPADHYAY, P., DE, M., KHALKAR, G. (2020). **“Medical Image Analysis using Deep Convolutional Neural Networks: CNN Architectures and Transfer Learning”**, 2020 International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India. pp. 175–180. doi: 10.1109/ICICT48043.2020.9112469.
- EL-DAHSHAN, E. S. A., HOSNY, T. & SALEM, A. B. M. (2010). **“Hybrid intelligent techniques for MRI brain images classification”**, Digital Signal Processing, 20 (2), 433-441
- EMAN, A.M., MOHAMMED, E., RASHID A.A. (2015). **“Brain tumor segmentation based on a hybrid clustering technique”**, Egyptian Informatics Journal, 16 (1), 71–81.
- ERICKSON, B. J., KORFIATIS, P., KLINE, T. L., AKKUS, Z., PHILBRICK, K. & WESTON, A. D. (2018). **“Deep learning in radiology: Does one size fit all?”**, Journal of the American College of Radiology, 15 (3), 521-526.
- ESTEVA, A., KUPREL, B., NOVOA, R.A., KO, J., SWETTER, S.M., BLAU, H.M., THRUN, S. (2017). **“Dermatologist-level classification of skin cancer with deep neural networks”**. Nature 542, 115–118. doi:10.1038/nature21056.
- FUKUSHIMA, K. (1980). **“Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position”**, Biol Cybern 36 (4), 193–202.
- GAO, X., LIN, S., WONG, T.Y. (2015). **“Automatic feature learning to grade**

- nuclear cataracts based on deep learning**". IEEE Trans Biomed. Eng. 62 (11), 2693–2701. doi:10.1109/TBME.2015.2444389.
- GUDIGAR, A., RAGHAVENDRA, U., SAN, T. R., CIACCIO, E. J. & ACHARYA, U. R. (2019). **"Application of multiresolution analysis for automated detection of brain abnormality using MR images: A comparative study"**, Future Generation Computer Systems, 90, 359-367.
- GULSHAN, V., PENG, L., CORAM, M., STUMPE, M.C., WU, D., NARAYANASWAMY, A., VENU-GOPALAN, S., WIDNER, K., MADAMS, T., CUADROS, J., KIM, R., RAMAN, R., NELSON, P.C., MEGA, J.L., WEBSTER, D.R. (2016). **"Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs"**. J. Am. Med. Assoc. 316, 2402–2410. doi:10.1001/jama.2016.17216
- GURUSAMY, R., SUBRAMANIAM, V. (2017). **"A machine learning approach for MRI brain tumor classification"**, Computers, Materials & Continua. 53(2), 91-108.
- HAQUE, K. F., HAQUE, F.F., GANDY, L., ABDELGAWAD, A. (2020). **"Automatic Detection of COVID-19 from Chest X-ray Images with Convolutional Neural Networks"**, 2020 International Conference on Computing, Electronics & Communications Engineering (iCCECE), Southend, United Kingdom. pp. 125–130. doi: 10.1109/iCCECE49321.2020.9231235.
- He, K., ZHANG, X., REN, S. and J. SUN. (2015) **"Deep residual learning for image recognition"**, CoRR, vol. abs/1512.03385, 2015
- HWANG, S., KIM, H. (2016). **"Self-transfer learning for fully weakly supervised object localization"**. arxiv:1602.01625.
- ISTOOK, E., MARTINEZ, T. (2002) **"Improved backpropagation learning in neural networks with windowed momentum"**. Int. J. Neural Syst. 12 (3–4)) 303–318 Doi : 10.1142/S0129065702001114
- JONES, T. L., BYRNES, T. J., YANG, G., HOWE, F. A., BELL, B. A., BARRICK, T. R. (2015). **"Brain tumor classification using the diffusion tensor image segmentation (D-SEG) technique"**, Neuro-oncology. 17(3), 466-476. Doi: 10.1093/neuonc/nou159

- JOTHI, G., HANNAHINBARANI, H. (2016) **“Hybrid tolerance rough set-firefly based supervised feature selection for mri brain tumor image classification”**. Appl. Soft Comput. 46, 639–651
- KADAM, M., & DHOLE, A. (2017). **“Brain tumor detection using GLCM with the help of KSVM”**, International Journal of Engineering & Technical Research, 7 (2), 2454–4698.
- KALLENBERG, M., PETERSEN, K., NIELSEN, M., Ng, A., DIAO, P., IGEL, C., VACHON, C., HOLLAND, K., KARSSEMEIJER, N., LILLHOLM, M. (2016). **“Unsupervised deep learning applied to breast density segmentation and mammographic risk scoring”**. IEEE Trans. Med. Imaging 35, 1322–1331. doi:10.1109/TMI.2016.2532122.
- KAWAHARA, J., HAMARNEH, G. (2016). **“Multi-resolution-tract CNN with hybrid pretrained and skin-lesion trained layers”**. In: Proceedings of the Machine Learning in Medical Imaging. In: Lecture Notes in Computer Science, 10019, pp. 164–171. doi:10.1007/978-3-319-47157-0_20.
- KHARRAT, A., HALIMA, M. B., AYED, M. B. (2015). **“MRI brain tumor classification using support vector machines and meta-heuristic method”**, Paper presented at the 2015 15th International Conference on Intelligent Systems Design and Applications (ISDA). Doi: 10.1109/isda.2015.7489271
- KIM, E., CORTRE-REAL, M., BALOCH, Z. (2016). **“A deep semantic mobile application for thyroid cytopathology”**. In: Proceedings of the SPIE on Medical Imaging, 9789, p. 97890A.
- KOLESNIKOV, A., BEYER, L., ZHAI, X., PUIGSERVER, J., YUNG, J., GELLY, S. and HOULSBY, N. (2019) **“Large scale learning of general visual representations for transfer”**. arXiv preprint arXiv:1912.11370
- KONG, B., ZHAN, Y., SHIN, M., DENNY, T., ZHANG, S. (2016). **“Recognizing end-diastole and end-systole frames via deep temporal regression network”**. In: Proceedings of the Medical Image Computing and Computer-Assisted Intervention. In: Lecture Notes in Computer Science, 9901, pp. 264–272. doi:10.1007/ 978-3-319-46726-9_31.
- KOOI, T., van GINNEKEN, B., KARSSEMEIJER, N., den HEETEN, A. (2017). **“Discriminating solitary cysts from soft tissue lesions in mammography using a pretrained deep convolutional neural**

- network**". Med. Phys 44 (3), 1017–1027.
- LECUN, Y., BOTTOU, L., BENGIO, Y., HAFFNER, P. (1998). **"Gradient-based learning applied to document recognition"**, Proceedings of the IEEE 86, 2278–2324.
- LEUNG, H., & HAYKIN, S. (1991). **"The complex backpropagation algorithm"**. IEEE Transactions on Signal Processing. 39(9), 2101–2104 Doi: 10.1109/78.134446
- LI, S., LIU, Y., SUI, X., CHEN, C., TIJO, G., TING, D. S. W., GOH, R. S. (2019). **"Multi-Instance Multi-Scale CNN for Medical Image Classification"**, Medical Image Computing and Computer Assisted Intervention – MICCAI 2019, vol. 11767, D. Shen, T. Liu, T. M. Peters, L. H. Staib, C. Essert, S. Zhou, P.-T. Yap, and A. Khan, Eds. Cham: Springer International Publishing. pp. 531–539 (2019)
- LIN C.-J., JENG, S.-Y. (2020). **"Optimization of Deep Learning Network Parameters Using Uniform Experimental Design for Breast Cancer Histopathological Image Classification"**, Diagnostics. vol. 10, no. 9, p. 662. doi: 10.3390/diagnostics10090662.
- LITJENS, G., KOOL, T., BEJNORDI, B. E., SETIO, A. A. A., CIOMPI, F., GHAFORIAN, M., ... SÁNCHEZ, C. I. (2017). **"A survey on deep learning in medical image analysis"**, Medical Image Analysis, 42, 60-88.
- LOUIS, D. N., PERRY, A., REIFENBERGER, G., VON DEIMLING, A., FIGARELLA-BRANGER, D., CAVENEE, W. K., OHGAKI, H., WIESTLER, O. D., KLEIHUES, P. and ELLISON, D. W. (2016). **"The 2016 world health organization classification of tumors of the central nervous system: a summary,"** Acta Neuropathologica, vol. 131, no. 6, pp. 803–820. <https://doi.org/10.1007/s00401-016-1545-1>
- MABRAY, M. C., & CHA, S. (2016). **"Advanced MR imaging techniques in Daily practice"**, Neuroimaging Clinics, 26(4), 647-666.
- MAFTOUNI, M., LAW, A.C, SHEN, B., ZHOU, Y., YAZDI, N., and KONG, Z.J. (2021) **"A Robust Ensemble-Deep Learning Model for COVID-19 Diagnosis based on an Integrated CT Scan Images Database"**. Proceedings of the 2021 Industrial and Systems Engineering Conference, Virtual Conference, May 22-25.
- MALLAT, S. G. (1989). **"A theory for multiresolution signal decomposition: The**

- wavelet representation**", IEEE Transactions on Pattern Analysis and Machine Intelligence, 11(7), 674-693.
- MCBEE, M. P., AWAN, O. A., COLUCCI, A. T., GHOBADI, C. W., KADOM, N., KANSARGA, A. P., ... AUFFERMANN, W. F. (2018). **"Deep learning in radiology"**, Academic Radiology, 25(11), 1472-1480.
- MENEGOLA, A., FORNACIALI, M., PIRES, R., AVILA, S., VALLE, E. (2016). **"Towards automated melanoma screening: exploring transfer learning schemes"**. arxiv:1609.01228.
- MORITZ, C. H., HAUGHTON, V. M., CORDES, D., QUIGLEY, M. & MEYERAND, M. E. (2000). **"Whole-brain functional MR imaging activation from a finger-tapping task examined with independent component analysis"**, American Journal of Neuroradiology, 21(9), 1629-1635.
- MOUSAVI, S.M., NAGHSH, A., ABU-BAKAR, S.A.R. (2015). **"A heuristic automatic and robust ROI detection method for medical image watermarking"**, Journal of Digital Imaging, 28 (4), 417–427.
- NABIZADEH, N., & KUBAT, M. (2015). **"Brain tumors detection and segmentation in MR images: Gabor wavelet vs. statistical features"**, Computers & Electrical Engineering, 45, 286–301.
- NAWI, N. M., HAMZAH, F., HAMID, N. A., REHMAN, M. Z., AAMIR, M., & RAMLI, A. A.(2017) **"An optimized back propagation learning algorithm with adaptive learning rate"**. International Journal on Advanced Science, Engineering and Information Technology. 7 (5), 1693 (2017) Doi: 10.18517/ijaseit.7.5.2972
- NAYAK, D. R., DASH, R. & MAJHI, B. (2016). **"Brain MR image classification using two-dimensional discrete wavelet transform and AdaBoost with random forests"**, Neurocomputing, 177, 188-197.
- NAYAK, D. R., DASH, R. & MAJHI, B. (2017). **"Stationary wavelet transform and AdaBoost with SVM based pathological brain detection in MRI scanning"**, CNS & Neurological Disorders – Drug Targets (Formerly Current Drug Targets-CNS & Neurological Disorders), 16(2), 137-149.
- PANDIYAN, V., TJAHJOWIDODO, T., CAESARENDRA, W., MURUGAN, P. (2019). **"In-process virtual verification of weld seam removal in robotic abrasive belt grinding process using deep learning"**. Robotics and

- PAPANDRIANOS, N., PAPAGEORGIOU, E. I. (2020). **“Anagnostis, A.: Development of Convolutional Neural Networks to identify bone metastasis for prostate cancer patients in bone scintigraphy”**, Ann. Nucl. Med., vol. 34, no. 11, pp. 824–832. doi: 10.1007/s12149-020-01510-6.
- PEREIRA, S., PINTO, A., ALVES, V., SILVA, C. A. (2016). **“Brain tumor segmentation using convolutional neural networks in MRI images”**, IEEE transactions on medical imaging 35(5): 1240-1251.
- PLIS, S.M., HJELM, D.R., SALAKHUTDINOV, R., ALLEN, E.A., BOCKHOLT, H.J., LONG, J.D., JOHNSON, H.J., PAULSEN, J.S., TURNER, J.A., CALHOUN, V.D. (2014). **“Deep learning for neuroimaging: a validation study”**. Front. Neurosci. doi:10.3389/fnins.2014.00229.
- POUDEL, R. P. K., LAMATA, P., MONTANA, G. (2016). **“Recurrent fully convolutional neural networks for multi-slice MRI cardiac segmentation”**. arxiv:1608.03974.
- PRIYANKA P., KUMAR, D. (2020). **“Meta-Heuristic Optimization Based Convolutional Neural Network for Medical Image Classification: A Survey”**, 2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), Greater Noida, India. pp. 752–756. doi: 10.1109/ICACCCN51052.2020.9362793.
- QIU, G., VARLEY, M.R., TERRELL, T.J. (1992) **“Accelerated training of backpropagation networks by using adaptive momentum step”**. Electron. Lett. 28 (4) 377–379 Doi: 10.1049/el:19920236
- QURATUL, A., JAFFAR, M. A., CHOI, T.-S. (2014). **“Fuzzy anisotropic diffusion based segmentation and texture based ensemble classification of brain tumor”**, Applied Soft Computing. 21, 330-340. Doi: 10.1016/j.asoc.2014.03.019
- RAMOS, R. M., RALHA, C. G., KURC, T. M., SALTZ, J. H., TEODORO, G. (2019) **“Increasing Accuracy of Medical CNN Applying Optimization Algorithms: An Image Classification Case”**, 2019 8th Brazilian Conference on Intelligent Systems (BRACIS), Salvador, Brazil. pp. 233–238. doi: 10.1109/BRACIS.2019.00049.

- RONNEBERGER, O., FISCHE, P., BROX, T. (2015). **“U-net: convolutional networks for biomedical image segmentation”**. In: Proceedings of the Medical Image Computing and Computer-Assisted Intervention. In: Lecture Notes in Computer Science, 9351, pp. 234–241. doi:10.1007/978-3-319-24574-4_28.
- RANA P. S. and SINGH, Parvinder (2015). **“Brain tumor detection through mr images: A review of literature”**. IOSR Journal of Computer Engineering (IOSR-JCE). 17(5) 7-18.
- RUMELHART, D. E., HINTON, G. E., WILLIAMS, R. J. (1986) **“Learning representations by back-propagating errors”**. Nature. 323 (6088), 533-536 Doi: 10.1038/323533a0
- SAJJAD, M., KHAN, S., MUHAMMAD, K., WU, W., ULLAH, A., BAIK, S. W. (2019). **“Multi-grade brain tumor classification using deep CNN with extensive data augmentation”**, Journal of computational science. 30, 174-182. Doi: 10.1016/j.jocs.2018.12.003
- SAMALA, R.K., CHAN, H.-P., HADJIISKI, L., HELVIE, M.A., WEI, J., CHA, K. (2016). **“Mass detection in digital breast tomosynthesis: deep convolutional neural network with transfer learning from mammography”**. Med. Phys. 43 (12), 6654–6666.
- SARITHA, M., JOSEPH, K. P. & MATHEW, A. T. (2013). **“Classification of MRI brain images using combined wavelet entropy based spider web plots and probabilistic neural network”**, Pattern Recognition Letters, 34(16), 2151-2156.
- SCARPACE, L., FLANDERS, A. E., JAIN, R., MIKKELSEN, T., & ANDREWS, D. W. (2019). **“Data From REMBRANDT [Data set]”**. The Cancer Imaging Archive.
- SETIO, A.A.A., CIOMPI, F., LITJENS, G., GERKE, P., JACOBS, C., van RIEL, S., WILLE, M.W., NAQIBULLAH, M., SANCHEZ, C., van GINNEKEN, B. (2016). **“Pulmonary nodule detection in CT images: false positive reduction using multi-view convolutional networks”**. IEEE Trans. Med. Imaging 35 (5), 1160–1169. doi:10.1109/TMI.2016.2536809.
- SHEN, D., WU, G., SUK, H.-I. (2017). **“Deep learning in medical image analysis”**, Annu. Rev. Biomed. Eng. 19:221–248. - [PMC](#) - [PubMed](#)
- SHEN, W., ZHOU, M., YANG, F., YANG, C., TIAN, J. (2015). **“Multi-scale**

- convolutional neural networks for lung nodule classification**". In: Proceedings of the Information Processing in Medical Imaging. In: Lecture Notes in Computer Science, 9123, 588–599. doi:10.1007/978-3-319-19992-4_46
- STOLLENGA, M.F., BYEON, W., LIWICKI, M., SCHMIDHUBER, J. (2015). **"Parallel multi-dimensional LSTM, with application to fast biomedical volumetric image segmentation"**. In: Proceedings of the Advances in Neural Information Processing Systems, Cornell University, 2998–3006. <https://doi.org/10.48550/arXiv.1506.07452>
- SUDHARANI, K., SARMA, T.C., PRASAD, K.S. (2016). **"Advanced morphological technique for automatic brain tumor detection and evaluation of statistical parameters"**, Procedia Technology, 24, 1374–1387.
- SUK, H.-I., LEE, S.-W., SHEN, D. (2014). **"Hierarchical feature representation and multimodal fusion with deep learning for AD/MCI diagnosis"**. Neuroimage 101, 569– 582. doi:10.1016/j.neuroimage.2014.06.077
- SUK, H.-I., SHEN, D. (2013). **"Deep learning-based feature representation for AD/MCI classification"**. In: Proceedings of the Medical Image Computing and Computer-Assisted Intervention. In: Lecture Notes in Computer Science, 8150, 583– 590. doi:10.1007/978-3-642-40763-5_72.
- SUN, W., TSENG, T.-L. B., ZHANG, J., QIAN, W. (2016). **"Enhancing deep convolutional neural network scheme for breast cancer diagnosis with unlabeled data"**. Comput. Med. Imaging Graph. 2017; 57:4–9. doi: 10.1016/j.compmedimag.2016.07.004. - [DOI](#) - [PubMed](#)
- SWATI Z. N. K., ZHAO, Q., KABIR, M., ALI, F., ALI, Z., AHMED, S., LU, J. (2019). **"Brain tumor classification for MR images using transfer learning and fine- tuning"**, Comput. Med. Imaging Graph., 75, 34–46. Doi:10.1016/j.compmedimag.2019.05.001.
- ŞAHİNER, B., CHAN, H.-P., PETRICK, N., WEI, D., HELVIE, M.A., ADLER, D.D., GOODSITT, M.M. (1996). **"Classification of mass and normal breast tissue: a convolution neural network classifier with spatial domain and texture images"**. IEEE Trans. Med. Imaging 15, 598–610. doi:10.1109/42.538937
- TANG, Y. – X., TANG, Y.- B., HAN, M., JING, X., SUMMERS, R. M. (2019)

- “Abnormal Chest X-Ray Identification with Generative Adversarial One-Class Classifier”**, 2019 IEEE 16th International Symposium on Biomedical Imaging <https://ieeexplore.ieee.org/document/8759442>
- TIWARI, A., SRIVASTAVA, S., PANT, M. (2020) **“Brain tumor segmentation and classification from magnetic resonance images: Review of selected methods from 2014 to 2019”**. Pattern Recognition Letters. 131, 244-260
Doi: doi.org/10.1016/j.patrec.2019.11.020
- USMAN, K., RAJPOOT, K. (2017). **“Brain tumor classification from multi-modality MRI using wavelets and machine learning”**, Pattern Analysis and Applications. 20 (3), 871-881. Doi: 10.1007/s10044-017-0597-8
- VAN TULDER, G., de BRUIJNE, M. (2016). **“Combining generative and discriminative representation learning for lung CT analysis with convolutional Restricted Boltzmann machines”**. IEEE Trans. Med. Imaging 35 (5), 1262–1272.
- WANG, L., ZENG, Y., & CHEN, T. (2015a) **“Back propagation neural network with adaptive differential evolution algorithm for time series forecasting”**. Expert Systems with Applications. 42(2), 855-863 Doi: 10.1016/j.eswa.2014.08.018
- WANG, S., ZHANG, Y., DONG, Z., DU, S., JI, G., YAN, J., ... PHILLIPS, P. (2015b). **“Feed-forward neural network optimized by hybridization of PSO and ABC for abnormal brain detection”**, International Journal of Imaging Systems and Technology, 25(2), 153-164
- WANG, X, PENG Y, LU L, LU Z, BAGHERI M, SUMMERS RM. (2017) **“ChestX-ray8: Hospital-scale Chest X-ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases”**. IEEE CVPR, *ChestX-ray8Hospital-ScaleChestCVPR2017_paper.pdf*
- WEI, X., LI, W., ZHANG, M., LI. Q. (2019). **“Medical Hyperspectral Image Classification Based on End-to-End Fusion Deep Neural Network,”** IEEE Trans. Instrum. Meas. 68 (11), 4481–4492. doi: 10.1109/TIM.2018.2887069.
- WESTPHAL, E., SEITZ, H. (2021). **“A Machine Learning Method for Defect Detection and Visualization in Selective Laser Sintering based on**

- Convolutional Neural Networks**", Additive Manufacturing (IF10.998). 41, DOI: 10.1016/j.addma.2021.101965
- WOLTERINK, J.M., LEINER, T., de VOS, B.D., van HAMERSVELT, R.W., VIERGEVER, M.A., ISGUM, I. (2016). **"Automatic coronary artery calcium scoring in cardiac CT angiography using paired convolutional neural networks"**. Med. Image Anal. 34, 123–136. doi:10.1016/j.media.2016.04.004.
- XIE, Y., ZHANG, Z., SAPKOTA, M., YANG, L. (2016b). **"Spatial clockwork recurrent neural network for muscle perimysium segmentation"**. In: Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention. In: Lecture Notes in Computer Science, 9901. Springer, pp. 185–193. doi:10.1007/978-3-319-46723-8_22.
- YAN, Z., ZHAN, Y., PENG, Z., LIAO, S., SHINAGAWA, Y., ZHANG, S., METAXAS, D.N., ZHOU, X.S. (2016). **"Multi-instance deep learning: discover discriminative local anatomies for bodypart recognition"**. IEEE Trans. Med. Imaging 35 (5), 1332–1343.
- YAQUB, M., JAVAID, M. K., COOPER, C., NOBLE, J. A. (2014). **"Investigation of the role of feature selection and weighted voting in random forests for 3–D volumetric segmentation"**, IEEE Transactions on Medical Imaging, 33 (2), 258– 271.
- YU, L., YANG, X., CHEN, H., QIN, J., HENG, P.A. (2017). **"Volumetric convnets with mixed residual connections for automated prostate segmentation from 3D MR images"**. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence.
- YU, W-J., SHEN, M., CHEN, W-N., ZHAN, Z-H., GONG, Y-J., LIN, Y., LIU, O., ZHANG, J. (2014). **"Differential Evolution with Two-Level Parameter Adaptation"**. IEEE Transactions on Cybernetics. 44(7), 1080-1099 Doi: 10.1109/tcyb.2013.2279211
- ZHANG, Q., BAI, C., LIU, Z., YANG, L. T., YU, H., ZHAO, J., YUAN, H. (2020). **"A GPU-based residual network for medical image classification in smart medicine"**, Inf. Sci. vol. 536, pp. 91–100 (2020) doi: 10.1016/j.ins.2020.05.013.

- ZHANG, Q., XIAO, Y., DAI, W., SUO, J., WANG, C., SHI, J., ZHENG, H. (2016). **“Deep learning based classification of breast tumors with shear-wave elastography”**. Ultrasonics 72, 150–157. doi:10.1016/j.ultras.2016.08.004.
- ZHANG, Y., DONG, Z., WANG, S., JI, G. & YANG, J. (2015). **“Preclinical diagnosis of magnetic resonance (MR) brain images via discrete wavelet packet transform with Tsallis entropy and generalized eigen-value proximal support vector machine (GEPSVM)”**, Entropy, 17(4), 1795-1813.
- ZHANG, Y., WANG, S., JI, G. & DONG, Z. (2013). **“An MR brain images classifier system via particle swarm optimization and kernel support vector machine”**, The Scientific World Journal. Article ID 130134, 9 pages, 2013. <https://doi.org/10.1155/2013/130134>

ELEKTRONİK KAYNAKLAR

- PRAMODITHA, R. **“The concept of artificial neurons (Perceptrons) in neural networks”**. <https://towardsdatascience.com/the-concept-of-artificial-neurons-perceptrons-in-neural-networks-fab22249cbfc>, Last Accessed Date: 20.03.2021
- URL-1 Gradient Descent. <https://www.ibm.com/cloud/learn/gradient-descent#:~:text=There%20are%20three%20types%20of,and%20mini%2Dbatch%20gradient%20descent> Last Accessed Date: 17.08.2021
- URL-2 **“About brain tumors, a primer for patients and caregivers”**. American Brain Tumor Association, 2015. [Online]. Available: [about-brain-tumors-a-primer-1.pdf \(abta.org\)](https://www.abta.org/about-brain-tumors-a-primer-1.pdf) Last Accessed Date: 10.01.2022
- URL-3 https://www.researchgate.net/publication/350319854_A_Machine_Learning_Method_for_Defect_Detection_and_Visualization_in_Selective_Laser_Sintering_based_on_Convolutional_Neural_Networks Last Accessed Date: 27.01.2021
- URL-4 <https://blog.devgenius.io/resnet50-6b42934db431> Last Accessed Date: 18.04.2022
- ROSEBROCK, A. (2016). **“Finding extreme points in contours with OpenCV”**. [Finding extreme points in contours with OpenCV - PyImageSearch](https://www.pyimagesearch.com/2016/04/18/finding-extreme-points-in-contours-with-opencv/) Last Accessed Date: 12.02.2022

TEZLER

BAYDAR, B. (2018) **“Convolutional neural network based brain MRI segmentation”**, (Yüksek lisans tezi), Electrical and Electronics Engineering, Middle East Technical University.

KADİROĞLU, Z. (2019). **“Histopatolojik meme kanseri görüntülerinin evrimsel sinir ağları kullanılarak sınıflandırılması”**, (Yüksek Lisans Tezi), Fen Bilimleri Enstitüsü, Fırat Üniversitesi.

SERTKAYA, M. E. (2018) **“Derin öğrenme tekniklerinin biyomedikal imgeler üzerine uygulamaları”**, (Yüksek Lisans Tezi), Fen Bilimleri Enstitüsü, Fırat Üniversitesi.

WANG, H. (2018) **“A CNN Based method for brain tumor detection”**, (Master thesis), Data Science Program for Electrical and Computer Engineering, Carleton University.



EKLER

EK A: Kaynak Kodlar

- https://github.com/utkucanaytac/Adaptive_Momentum

#Data Pre-Process

```
import pandas as pd
import numpy as np
import seaborn as sns
import sys
import pickle
sns.set()
np.set_printoptions(threshold=sys.maxsize)
from tensorflow.keras.applications.densenet import DenseNet121
from tensorflow.keras.layers import Dense, GlobalAveragePooling2D
from tensorflow.keras.models import Model

train_df = pd.read_csv("/Users/utkucanaytac/Desktop/REMBRAND/phd.csv") # the
metada of rembrandt
print(train_df.shape)

pd.set_option('display.expand_frame_repr', False)

print(train_df.head())

print(f'There are {train_df.shape[0]} rows and {train_df.shape[1]} columns in this
data')

print(train_df.info())

print(f'the total number of patiens are {train_df['Sample'].count()}, unique ids are
{train_df['Sample'].value_counts().shape[0]}')
columns = train_df.keys()
columns = list(columns)
columns.remove('Sample')

for i in columns:
    print(f'{i} hastalığına sahip {train_df[i].sum()} hasta vardır. ")

images = train_df['Sample'].values
```

```

random_images = [np.random.choice(images) for i in range (9)]

img_dir = '/Users/utkucanaytac/Desktop/REMBRAND/REMBRANDT'

# TumorA = astrocytoma = 0
# TumorB = glioblastoma_multiforme = 1
# TumorC = oligodendroglioma = 2
# healthy = 3
# unknown = 4
## Load from the file for X(image data) and Y(tumor type)

file = open('/Users/utkucanaytac/Desktop/full_dataset_final.pkl', 'rb')
print("pickle file open")
allX, allY = pickle.load(file) # allX = training data; allY = training label

arr = np.array(allY)
df = pd.DataFrame(data=arr.flatten())
df.columns = ['A']

uca = pd.get_dummies(df.A)

uca.columns = ['Astrocytoma','Glioblastoma','Oligodendroglioma',
               'Healthy Brain','Unidentified']

#uca.insert(0, 'Sample',(range (0,96115)))
print(uca)

a = []
for i in uca.keys():
    a.append(uca[i].sum())
    print(f' {i} hastalığına sahip {uca[i].sum()} hasta vardır. ")
print(a)
from matplotlib import pyplot as plt
plt.figure(figsize=(2,2))
plt.imshow(allX[0], interpolation='nearest')
plt.axis('off')
plt.show()

sns.set(font_scale=0.8,)
sns.barplot(uca.columns, a, color='b')
plt.title('Distribution of Classes for Training Dataset', fontsize=12)
plt.xlabel('Diseases', fontsize=10)
plt.ylabel('Number of Data', fontsize=10)
plt.show()

```

```

y_pred_1 = 0.9 * np.ones(allY.shape)
print(y_pred_1)

import ssl
ssl._create_default_https_context = ssl._create_unverified_context

base_model = DenseNet121(weights= 'imagenet',include_top = False);

layers = base_model.layers

print(layers[0:6])
print(base_model.input)

labels = uca.keys()
print(labels)
classes= len(labels)

print(f"toplam {classes} hastalığı tahmin edeceğiz")

print(f"Predictions have {classes} units, one for each class")


file.close()

#Data Pre process metadata

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import os
import seaborn as sns
import cv2

sns.set()

train_df = pd.read_excel("/clinical_2014-01-16.xlsx") #metada of rembrandt
print(train_df.shape)

pd.set_option('display.expand_frame_repr', False)

print(train_df.head())

print(f'There are {train_df.shape[0]} rows and {train_df.shape[1]} columns in this
data')

print(train_df.info())

```

```
print(f"the total number of patiens are {train_df['Sample'].count()}, unique ids are  
{train_df['Sample'].value_counts().shape[0]}")
```

```
#for i in columns:
```

```
    #print(f'{i} hastalığına sahip {train_df[i].sum()} hasta vardır. ")
```

```
#images = train_df['Image'].values
```

```
#random_images = [np.random.choice(images) for i in range (9)]
```

```
train_df.columns = train_df.columns.str.replace(' ', ")
```

```
columns = train_df.keys()
```

```
columns = list(columns)
```

```
print(columns)
```

```
hastalanan = pd.get_dummies(train_df.Disease)
```

```
for i in hastalanan:
```

```
    print(f'{i} hastalığına sahip {hastalanan[i].sum()} hasta vardır. ")
```

```
train_df = pd.concat([train_df, hastalanan], axis=1)
```

```
train_df.columns = train_df.columns.str.replace(' ', ")
```

```
print(train_df.head())
```

```
new = train_df[['Sample','--
```

```
','ASTROCYTOMA','GBM','MIXED','OLIGODENDROGLIOMA']].copy()
```

```
new = new.rename(columns={"--": "OTHER"})
```

```
new['Sample'].replace({'_': '-'}, inplace=True, regex=True)
```

```
print(new.head())
```

```
new.to_csv('/Users/utkucanaytac/Desktop/REMBRAND/phd.csv')
```

```
from __future__ import division, print_function, absolute_import  
import os
```

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
import sys
```

```
np.set_printoptions(threshold=sys.maxsize)
```

```
import cv2
```

```
from skimage.transform import resize
```

```
np.set_printoptions(suppress=True)
```

```
import skimage
```

```

import imutils
from six.moves import cPickle
import pickle

def crop_brain_contour(image, plot=False):

    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    thresh = cv2.threshold(gray, 0, 255, cv2.THRESH_OTSU +
cv2.THRESH_BINARY)[1]

    cnts = cv2.findContours(thresh.copy(), cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
    cnts = imutils.grab_contours(cnts)
    c = max(cnts, key=cv2.contourArea)

    extLeft = tuple(c[c[:, :, 0].argmin()][0])
    extRight = tuple(c[c[:, :, 0].argmax()][0])
    extTop = tuple(c[c[:, :, 1].argmin()][0])
    extBot = tuple(c[c[:, :, 1].argmax()][0])

    new_image = image[extTop[1]:extBot[1], extLeft[0]:extRight[0]]

    if plot:
        plt.figure()

        plt.subplot(1, 2, 1)
        plt.imshow(image)

        plt.tick_params(axis='both', which='both',
                        top=False, bottom=False, left=False, right=False,
                        labelbottom=False, labeltop=False, labelleft=False, labelright=False)

        plt.title('Original Image')

        plt.subplot(1, 2, 2)
        plt.imshow(new_image)

        plt.tick_params(axis='both', which='both',
                        top=False, bottom=False, left=False, right=False,
                        labelbottom=False, labeltop=False, labelleft=False, labelright=False)

        plt.title('Cropped Image')

        plt.show()

    return new_image

```

```

FolderPathName_1 = '/Users/utkucanaytac/Desktop/IMAGES/ASTO'
FolderPathName_2 = '/Users/utkucanaytac/Desktop/IMAGES/GBM'

```

```
FolderPathName_3='/Users/utkucanaytac/Desktop/IMAGES/OLIGO'  
FolderPathName_4='/Users/utkucanaytac/Desktop/IMAGES/UNKNOWN'
```

```
tumor1 = [] # create an empty list  
for dirName, subdirList, fileList in os.walk(FolderPathName_1):  
    for filename in fileList:  
        if ".png" in filename.lower(): # check whether the file's PNG  
            tumor1.append(os.path.join(dirName,filename))  
  
tumor2 = []  
for dirName, subdirList, fileList in os.walk(FolderPathName_2):  
    for filename in fileList:  
        if ".png" in filename.lower(): # check whether the file's PNG  
            tumor2.append(os.path.join(dirName,filename))  
  
tumor3=[]  
for dirName, subdirList, fileList in os.walk(FolderPathName_3):  
    for filename in fileList:  
        if ".png" in filename.lower(): # check whether the file's PNG  
            tumor3.append(os.path.join(dirName,filename))  
  
tumor4= []  
for dirName, subdirList, fileList in os.walk(FolderPathName_4):  
    for filename in fileList:  
        if ".png" in filename.lower(): # check whether the file's PNG  
            tumor4.append(os.path.join(dirName,filename))  
  
n_files = len(tumor1) + len(tumor2) + len(tumor3) + len(tumor4)  
size_image = 64  
allX = np.zeros((n_files, size_image, size_image, 3), dtype='float32')  
ally = np.zeros((n_files), dtype='int32')  
count = 0  
y_count = 0  
  
for f in tumor1:  
    img = cv2.imread(f)  
    img = crop_brain_contour(img, plot=False)  
    new_img = skimage.transform.resize(img, (size_image, size_image, 3))  
    allX[count] = np.array(new_img)  
    ally[count] = 0  
    count +=1  
    y_count +=1  
  
print('done1')  
  
for f in tumor2:  
    img = cv2.imread(f)
```

```

img = crop_brain_contour(img, plot=False)
new_img = skimage.transform.resize(img, (size_image, size_image, 3))
allX[count] = np.array(new_img)
ally[count] = 1
count +=1
y_count +=1

print('done2')

for f in tumor3:
    img = cv2.imread(f)
    img = crop_brain_contour(img, plot=False)
    new_img = skimage.transform.resize(img, (size_image, size_image, 3))
    allX[count] = np.array(new_img)
    ally[count] = 2
    count +=1
    y_count +=1

print('done3')

for f in tumor4:
    try:
        img = cv2.imread(f)
        img = crop_brain_contour(img, plot=False)
        new_img = skimage.transform.resize(img, (size_image, size_image, 3))
        allX[count] = np.array(new_img)
        ally[count] = 3
        count +=1
        y_count +=1
    except:
        continue
print("images are arrayed")

f = open('/Users/utkucanaytac/Desktop/final.pkl', 'wb')
cPickle.dump((allX, ally), f, protocol=cPickle.HIGHEST_PROTOCOL)
f.close()

print('finish')

# Adapvite Momentum
"""

from google.colab import drive
drive.mount('/content/drive')

!pip install pyyaml h5py
import tensorflow as tf
tf.test.gpu_device_name()
import pickle

```

```

import tensorflow.keras as keras
import os
from keras.callbacks import Callback
import matplotlib.pyplot as plt
from tensorflow.keras.models import Sequential
from tensorflow.keras import backend as K
from tensorflow.keras.utils import to_categorical
import numpy as np
os.environ['TF_KERAS'] = '1'
from sklearn.utils.class_weight import compute_class_weight
import logging, warnings
from sklearn.metrics import classification_report
from tensorflow.python.training import optimizer
from tensorflow.python.ops import math_ops, state_ops, control_flow_ops
from tensorflow.python.framework import ops
import tensorflow as tf
from tensorflow.keras import backend as K
from tensorflow.keras.optimizers import SGD
from tensorflow.keras.layers import Conv2D, Input, BatchNormalization, Activation,
MaxPooling2D, Flatten, Dense, Dropout
from tensorflow.keras import regularizers
from tensorflow.keras.optimizers import Adadelta, Adam, Adagrad, RMSprop
from keras.callbacks import EarlyStopping, ModelCheckpoint
import os
import sys
np.set_printoptions(threshold=sys.maxsize)

def loggings():

    logging.disable(logging.WARNING)
    os.environ["TF_CPP_MIN_LOG_LEVEL"] = "3"
    warnings.filterwarnings('ignore')

loggings()

from google.colab import files
from google.colab import drive

file = open("/content/drive/My Drive/Colab Notebooks/data_uca.pkl" , "rb")

print("pickle file open")
allX, allY = pickle.load(file) # allX = training data; allY = training label
file.close()

print ('file is ok')

from sklearn.model_selection import train_test_split
from tensorflow.keras.callbacks import LearningRateScheduler

```

```

x_train, x_test, y_train, y_test = train_test_split(allX, allY, test_size=0.25,
random_state=42)

y_train = to_categorical(y_train,4)
y_test_cat = to_categorical(y_test,4)
batch_size = 128
epochs = 20

print ("number of training examples = " + str(x_train.shape[0]))
print ("number of test examples = " + str(x_test.shape[0]))

y_integers = np.argmax(y_train, axis=1)
#print(y_integers)
class_weights = compute_class_weight('balanced', np.unique(y_integers),
y_integers)
#print(class_weights)
d_class_weights = dict(enumerate(class_weights))
#print(d_class_weights)

def cnn_model():
    weight_decay = 0.0005
    model = Sequential()
    model.add(Conv2D(32, (3, 3), padding='same', input_shape=(64,64,3),
kernel_regularizer=regularizers.l2(weight_decay),activation='relu'))
    model.add(BatchNormalization())
    model.add(Dropout(0.3))
    model.add(Conv2D(64, kernel_size=(3, 3), strides=(2,2),
kernel_regularizer=regularizers.l2(weight_decay),activation='relu'))
    model.add(BatchNormalization())
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(Conv2D(128, (3, 3),
padding='same',kernel_regularizer=regularizers.l2(weight_decay),activation='relu'))
    model.add(BatchNormalization())
    model.add(Flatten())

    model.add(Dense(512,kernel_regularizer=regularizers.l2(weight_decay),activation='r
elu'))
    model.add(BatchNormalization())
    model.add(Dropout(0.5))
    model.add(Dense(4,activation = 'softmax'))
    return (model)
model = cnn_model()
model.summary()

model = cnn_model()
sgd = SGD(lr=0.01, momentum=0.9, decay=0.001, nesterov=False)
model.compile(loss=keras.losses.categorical_crossentropy, optimizer=sgd,
metrics=['accuracy'])
init_loss = model.evaluate(x_test, y_test_cat)[0]

```

```

loss_list = [init_loss]
class history_(keras.callbacks.Callback):
    def on_epoch_end(self, epoch, logs={}):
        loss = model.evaluate(x_test, y_test_cat)[0]
        loss_list.append(loss)
        for i in range(epoch+1):
            loss_t_minus = np.array(loss_list[i-2], np.float32)
            loss_t = np.array(loss_list[i-1], np.float32)
            beta = 0.90
            momentum = beta / (1. + np.exp(-1. * abs(1. / np.sqrt(np.exp(-1*(loss_t *
loss_t_minus))))))
            print("loss", loss)
            print("momentum", momentum)
            return momentum
loss_history_ = history_()

sgd = SGD(lr=0.01, momentum=history_.on_epoch_end(epochs, epochs),
nesterov=False)

earlyStopping = EarlyStopping(monitor='val_loss', patience=10, verbose=0,
mode='min')

checkpoint_path = "/Users/utkucanaytac/Desktop/model"
checkpoint_dir = os.path.dirname(checkpoint_path)

cp_callback = tf.keras.callbacks.ModelCheckpoint(filepath=checkpoint_path,
                                                save_best_only=True,
save_weights_only=True,monitor = 'val_loss',
                                                verbose=1)
callbacks_list_ = [loss_history_,earlyStopping,cp_callback]

model.compile(loss=keras.losses.categorical_crossentropy, optimizer=sgd,
metrics=['accuracy'])

history = model.fit(x_train, y_train,
                    validation_data=(x_test, y_test_cat),
                    epochs=epochs,
                    batch_size=batch_size,
                    callbacks=callbacks_list_,
                    verbose = 1)

K.clear_session()
model2 = cnn_model()

sgd_old = SGD(learning_rate=0.01, nesterov=False)

model2.compile(loss=keras.losses.categorical_crossentropy, optimizer=sgd_old,
metrics=['accuracy'])
history2 = model2.fit(x_train, y_train,

```

```

validation_data=(x_test, y_test_cat),
epochs=epochs,
batch_size=batch_size,
verbose = 1)

# fit CNN model using Adadelta optimizer

K.clear_session()
model3 = cnn_model()
model3.compile(loss=keras.losses.categorical_crossentropy,
               optimizer=Adadelta(lr=1.0, rho=0.95, epsilon=1e-08, decay=0.0),
               metrics=['accuracy'])
history3 = model3.fit(x_train, y_train,
                    validation_data=(x_test, y_test_cat),
                    epochs=epochs,
                    verbose=1)

# fit CNN model using RMSprop optimizer
K.clear_session()
model4 = cnn_model()
model4.compile(loss=keras.losses.categorical_crossentropy,
               optimizer=RMSprop(lr=0.001, rho=0.9, epsilon=1e-08, decay=0.0),
               metrics=['accuracy'])
history4 = model4.fit(x_train, y_train,
                    validation_data=(x_test, y_test_cat),
                    epochs=epochs,
                    verbose=1)

# fit CNN model using Adam optimizer
K.clear_session()
model5 = cnn_model()
model5.compile(loss=keras.losses.categorical_crossentropy,
               optimizer=Adam(lr=0.001, beta_1=0.9, beta_2=0.999, epsilon=1e-08,
                             decay=0.0),
               metrics=['accuracy'])
history5 = model5.fit(x_train, y_train,
                    validation_data=(x_test, y_test_cat),
                    epochs=epochs,
                    verbose=1)

# fit CNN model using Adagrad optimizer
K.clear_session()
model6 = cnn_model()
model6.compile(loss=keras.losses.categorical_crossentropy,
               optimizer=Adagrad(lr=0.01, epsilon=1e-08, decay=0.0),
               metrics=['accuracy'])
history6 = model6.fit(x_train, y_train,
                    validation_data=(x_test, y_test_cat),
                    epochs=epochs,
                    verbose=2)

```

```

fig = plt.figure(figsize=(12,8))
plt.plot(range(epochs),history.history['val_loss'],label='Adaptive_Momentum')
plt.plot(range(epochs),history2.history['val_loss'],label='SGD')
#plt.plot(range(epochs),history3.history['val_loss'],label='Adadelta')
plt.plot(range(epochs),history4.history['val_loss'],label='RMSprop')
plt.plot(range(epochs),history5.history['val_loss'],label='Adam')
#plt.plot(range(epochs),history6.history['val_accuracy'],label='Adagrad')

plt.legend(loc=0)
plt.xlabel('epochs')
plt.xlim([0,epochs])
plt.ylabel('accuracy on test set')
plt.grid(True)
plt.title("Comparing Model Accuracy")
plt.show()
fig.savefig(r'C:\Users\uacan.aytac\Desktop\compare-accuracy.jpg')
plt.close(fig)

target_names = ['Oligodendroglioma', 'Glioblastoma', 'Astrocytoma','Unidentified']

predictions = model.predict_classes(x_test)
print('Classification Report of Proposed Model')
print(classification_report(y_test,predictions,target_names=target_names))

fig = plt.figure(figsize=(12,8))
plt.plot(range(epochs),history.history['val_accuracy'],label='Adaptive_Momentum')
plt.plot(range(epochs),history2.history['val_accuracy'],label='SGD')
plt.plot(range(epochs),history4.history['val_accuracy'],label='RMSprop')
plt.plot(range(epochs),history5.history['val_accuracy'],label='Adam')

plt.legend(loc=0)
plt.xlabel('epochs')
plt.xlim([0,epochs])
plt.ylabel('accuracy on test set')
plt.grid(True)
plt.title("Comparing Model Accuracy")
plt.show()

m = [model,model2,model3,model4,model5,model6]

for i in m:
    predictions = i.predict_classes(x_test)
    print('Classification Report of Proposed Model')
    print(classification_report(y_test,predictions,target_names=target_names))

```

ÖZGEÇMİŞ

Ad-Soyad : Utku Can AYTAÇ

ÖĞRENİM DURUMU:

- **Lisans** : 2009, Kadir Has Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği
- **Yüksek Lisans** : 2011, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Mühendislik Yönetimi
- **Yüksek Lisans** : 2016, İstanbul Aydın Üniversitesi, Bilgisayar Mühendisliği

MESLEKİ DENEYİM VE ÖDÜLLER

08.2021 – Devam : Azerion, Veri Bilimi Mühendisi

01.2017 – 07.2021 : Triodor Ar-ge, Proje Koordinatörü

06.2012 – 12.2017 : İnovasyon Proje, Proje Uzmanı

TEZDEN TÜRETİLEN YAYINLAR, SUNUMLAR VE PATENTLER

Aytaç, U.C., Güneş, A. & Ajlouni, N. (2022) “**A novel adaptive momentum method for medical image classification using convolutional neural network**”. BMC Med Imaging 22, 34. <https://doi.org/10.1186/s12880-022-00755-z>

DİĞER YAYIN SUNUM VE PATENTLER

AKALP Gizem, YENİMAN Ebru, AYTAÇ, Utku Can. (2011). “**Bankamatik sistemlerinde ergonomik arayüz tasarımı ve kullanılabilirlik**” 17. Ulusal Ergonomi Kongresi, 14-16 Ekim 2011, Esişehir Osmangazi Üniversitesi, Eskişehir

HUSEYİNOV, I. and AYTAÇ, U. C. (2017) “**Identification of association rules in buying patterns of customers based on modified apriori and Eclat algorithms by using R programming language,**” 2017 International Conference on Computer Science and Engineering (UBMK). pp. 516-521, doi: 10.1109/UBMK.2017.8093450.

