

Endoskopi Görüntülerindeki Poliplerin
Derin Öğrenme Algoritması Kullanılarak İncelenmesi

Emine Cengiz

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

Haziran 2020

Investigation of Polyps in Endoscopy Images
by Using Deep Learning Algorithm

Emine Cengiz

MASTER OF SCIENCE THESIS

Department of Computer Engineering

June 2020

Endoskopi Görüntülerindeki Poliplerin
Derin Öğrenme Algoritması Kullanılarak İncelenmesi

Emine Cengiz

Eskişehir Osmangazi Üniversitesi
Fen Bilimleri Enstitüsü
Lisansüstü Yönetmeliği Uyarınca
Bilgisayar Mühendisliği Anabilim Dalı
Bilgisayar Bilimleri Bilim Dalında
YÜKSEK LİSANS TEZİ
Olarak Hazırlanmıştır.

Danışman: Prof. Dr. Eyyüp Gülbandılar
Danışman: Doç. Dr. Faik YAYLAK

Haziran 2020

ETİK BEYAN

Eskişehir Osmangazi Üniversitesi Fen Bilimleri Enstitüsü tez yazım kılavuzuna göre, Prof. Dr. Eyyüp Gülbandılar danışmanlığında hazırlamış olduğum “Endoskopi Görüntülerindeki Poliplerin Derin Öğrenme Algoritması Kullanılarak İncelenmesi” başlıklı YÜKSEK LİSANS tezimin özgün bir çalışma olduğunu; tez çalışmamın tüm aşamalarında bilimsel etik ilke ve kurallara uygun davrandığımı; tezimde verdiğim bilgileri, verileri akademik ve bilimsel etik ilke ve kurallara uygun olarak elde ettiğimi; tez çalışmamda yararlandığım eserlerin tümüne atıf yaptığımı ve kaynak gösterdiğimi ve bilgi, belge ve sonuçları bilimsel etik ilke ve kurallara göre sunduğumu beyan ederim.

Kütahya Sağlık Bilimleri Üniversitesi Girişimsel Olmayan Klinik Araştırmalar Etik Kurulu tarafından gerçekleştirilen 27.06.2019 tarih ve 2019/07 sayılı toplantıda alınan karar formu ekte sunulmuştur.02/06/2020

Emine Cengiz
İmza

ÖZET

Makine öğrenmesi ve özellikle derin öğrenme konularındaki son gelişmeler, tıbbi görüntülerdeki nesnelere tanımaya ve sınıflandırmaya yardımcı olmaktadır. Bu çalışmada endoskopi görüntüleri incelenmiş, normal ve polipli görüntülerin tespit edilerek sınıflandırılması için derin öğrenme yöntemi kullanılmıştır. Önerilen sistem için Kütahya Sağlık Bilimleri Üniversitesi Genel Cerrahi Anabilim Dalı, Evliya Çelebi Eğitim ve Araştırma Hastanesi Genel Cerrahi Endoskopi Ünitesi arşiv görüntüleri kullanılarak normal ve poliplerin olduğu görüntüler alınarak veri tabanı oluşturuldu. Oluşturulan veri tabanında 54 arşiv kaydından alınan, 93 polipli ve 216 normal görüntü bulunmaktadır. Bu verilerden rastgele olarak seçilen verilerin 2/3 ü modelin eğitim için ayrılırken, 1/3 ü modelin test için ayrılmıştır.

Bu tez çalışmasında, derin öğrenmedeki en iyi sınıflandırma modelini bulmak için farklı aktivasyon ve optimizasyon fonksiyonları kullanılarak 48 farklı model oluşturulmuştur. Elde edilen verilere göre; modellerin sonuçlarının parametrelere göre farklılık gösterdiği, en iyi modelin gizli katmanında 32 nöron, aktivasyon fonksiyonu ReLU ve optimizasyon yönteminin Adagrad olduğu durumda elde edildiği tespit edilmiştir (%98). Bunun aksine, en kötü modelin gizli katmanında 32 nöron, aktivasyon fonksiyonu ReLU ve optimizasyon fonksiyonu SGD olduğu durumda gözlenmiştir (%69).

Sonuç olarak, farklı aktivasyon ve optimizasyon yöntemleri kullanılarak derin öğrenme ile polip görüntü verilerinin başarılı sınıf tahmini yapan bir sistem tasarımı yapılabilir.

Anahtar Kelimeler: Derin öğrenme, aktivasyon fonksiyonu, optimizasyon yöntemi, nöron, polip

SUMMARY

Recent advances in machine learning, particularly with regard to deep learning, help to recognize and classify objects in medical images. In this study, endoscopy images were examined and deep learning method was used to classify healthy and polyp cells. For the proposed system, a database was created from the archives of Kutahya Health Sciences University General Surgery Department Endoscopy Unite in Kutahya Evliya Celebi Training and Research Hospital. The database contains 93 polyps and 216 normal images from 54 archive records. Of these data, 2/3 of the randomly selected data is reserved for the model's training, while 1/3 of the data is reserved for the test.

In this study, 48 different models were created by using different activation and optimization functions to find the best classification model in deep learning. According to the experimental results; It was determined that accuracy results of the models depend on the selected parameters. We observed the highest accuracy results by means of 32 neurons in the hidden layer, whereas ReLU activation function and Adagrad optimization method are utilized during the best model with the accuracy rate of 98%. In contrast, when we designed a second model with 32 neurons, utilizing ReLU activation and SGD optimization functions, the accuracy rate reduced to 69%.

In summary, classification performance of polyp images can be optimized by utilizing different activation and optimization methods during the design of deep learning models.

Keywords: Deep learning, activation function, optimization method, neuron, polyp

TEŐEKKÜR

Bu tez alıőmasının planlanması, araőtırılması ve yürütölmesi sırasında yardım ve desteęini esirgemeyen, bilgi ve tecrübelerinden yararlandıęım, yönlendirmesi ve bilgilendirmesi ile tezime őekil veren, her sıkıntıda yanına rahatlıkla gidebileceęim deęerli danıőman hocalarım Prof. Dr. Eyyüp GÜLBANDILAR ve Do. Dr. Faik YAYLAK'a sonsuz teőekkürlerimi sunarım.

Bana her zaman maddi manevi desteklerini esirgemeyen beni hiçbir zaman yalnız bırakmayan hayattaki en büyük gurur kaynaęım olan sevgili annem Aliye Cengiz'e, babam Mehmet Cengiz'e, kardeőim Mustafa Cengiz'e sevgileri, özverili destekleri, anlayıőları ve kesintisiz teőviklerinden dolayı teőekkür ederim.

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET	vi
SUMMARY	vii
TEŞEKKÜR	viii
İÇİNDEKİLER	ix
ŞEKİLLER DİZİNİ	xi
SİMGELER VE KISALTMALAR DİZİNİ	xii
1.GİRİŞ VE AMAÇ	1
2. LİTERATÜR ARAŞTIRMASI	4
2.1. Yapay Sinir Ağları	7
2.1.1 Biyolojik sinir hücresinin yapısı	7
2.1.2 Yapay sinir hücresinin yapısı	8
2.1.3 Yapay sinir ağı modeli ve çok katmanlı algılayıcı	9
2.1.4 Yapay sinir ağlarında öğrenme	10
2.1.5 İleri beslemeli yapay sinir ağları	11
2.1.6 Geri beslemeli yapay sinir ağları	12
2.1.7 Aktivasyon yöntemleri	13
2.1.7.1. <u>Sigmoid aktivasyon fonksiyonu</u>	13
2.1.7.2. <u>Tanh aktivasyon fonksiyonu</u>	14
2.1.7.3. <u>ReLU aktivasyon fonksiyonu</u>	15
2.1.8 Optimizasyon yöntemleri	16
2.1.8.1. <u>Skolastik gradyan inişi optimizasyon yöntemi</u>	16
2.1.8.2. <u>Adagrad, RMSProp ve Adam optimizasyon yöntemleri</u>	17
2.1.9 Hata fonksiyonları	17
2.1.9.1. <u>Ortalama karesel hata (Mean Squared Error)</u>	18
2.1.9.2. <u>Çapraz entropi (Cross Entropi)</u>	18
2.2. Derin Öğrenme Modelleri.....	19
2.2.1. Derin öğrenme.....	19
2.2.2. Derin öğrenme ve makine öğrenmesi arasındaki farklar.....	21
2.3. Konvolüsyonel Yapay Sinir Ağları ve Parametreleri	22
2.3.1. Konvolüsyon katmanı (Convolutional Layer)	23

İÇİNDEKİLER(devam)

	<u>Sayfa</u>
2.3.2. Alt örnekleme katmanı (Pooling Layer)	25
2.3.3. Tam bağlantılı katman (Fully Connected Layer)	27
2.3.4. Seyreltme katmanı (Dropout Layer)	27
2.3.5. Sınıflandırma katmanı (Classification Layer)	28
2.3.6. Eğitim dönemi sayısı (Epoch sayısı)	28
2.3.7. Mini-batch boyutu	28
2.3.8. Öğrenme katsayısı (Learning Rate)	29
2.4. Tekrarlayan sinir ağları (RNN)	29
2.4.1. Uzun kısa vadeli hafıza ağları (LSTM)	30
2.4.2. Kapalı yinelenen birim (GRU)	32
3. MATERYAL VE YÖNTEM	33
3.1. Veri Seti	33
3.1.1. Veri artırma	33
3.2. Hiperparametreler	34
3.3 Kullanılan Teknolojiler	35
3.3.1. Python	35
3.3.2. Pycharm	36
3.3.3. Numpy ve scipy	36
3.3.4. Tensorflow	37
3.3.5. Keras	37
3.4. Değerlendirme Kriterleri	37
3.4.1. Karmaşıklık matrisi	38
3.4.2. Sınıflandırma doğruluk oranları	39
4. BULGULAR VE TARTIŞMA	41
5. SONUÇ VE ÖNERİLER	47
KAYNAKLAR DİZİNİ	48
EK AÇIKLAMALAR	53
Ek Açıklama-A: Kütahya Sağlık Bilimleri Üniversitesi Girişimsel Olmayan Klinik Araştırmalar Etik Kurulu Raporu	54

ŞEKİLLER DİZİNİ

<u>Sekil</u>	<u>Sayfa</u>
1.1. Endoskopi işleminin genel görüntüsü	2
2.1. Sinir hücresi	7
2.2. Yapay bir nöron hücresi	8
2.3. Çok katmanlı yapay sinir ağları	9
2.4. İleri beslemeli yapay sinir ağı	11
2.5. Geri beslemeli ağ yapısı	12
2.6. Aktivasyon fonksiyonu	13
2.7. Sigmoid aktivasyon fonksiyonu	14
2.8. Tanh aktivasyon fonksiyonu	15
2.9. ReLU aktivasyon fonksiyonu	16
2.10. Derin öğrenme ve yapay zeka arasındaki ilişki	20
2.11. Derin öğrenme ve makine öğrenmesi arasındaki fark	21
2.12. Konvolüsyonel sinir ağları	23
2.13. 5*5 lik giriş verisine 3*3 lük filtrenin uygulanışı ve özellik haritasının çıkarımı	24
2.14. Alt örnekleme işleminde derinlik boyutunun değişimi	25
2.15. 5x5'lik giriş görüntüsüne 2x2 filtre ile bir ve iki adım kaymalı max pooling işleminin uygulanması	26
2.16. Tam bağlı katmanla tek boyuta düşürme işlemi	27
2.17. Yapay sinir ağına seyreltme tekniğinin uygulanması	28
2.18. Tekrarlayan sinir ağları	30
2.19. LSTM yapay sinir ağları	31
2.20. Kapılı yinelenen birim	32
4.1. 10'nolu modele ait hata grafiği	43
4.2. 10'nolu modele ait doğruluk grafiği	43

ÇİZELGELER DİZİNİ

<u>Çizelge</u>	<u>Sayfa</u>
2.1. Biyolojik ve yapay sinir ağları arasındaki ilişki	9
3.1. Sağlıklı ve polipli görüntüler	33
3.2. Veri artırma teknikleri	34
3.3. Çalışmada oluşturulan modellere ait hiper parametreler	35
3.4. Karmaşıklık matrisi	39
4.1. Çalışmada kullanılan modellere ait sonuçlar	41
4.2. En başarılı modele ait sonuçlar	44
4.3. En düşük başarıya sahip modellere ait sonuçlar	44

SİMGELER VE KISALTMALAR DİZİNİ

<u>Kısaltmalar</u>	<u>Açıklama</u>
API	Program Uygulama Arayüzü
CNN	Konvolüsyonel Sinir Ağları
CPU	Merkezi İşlem Birimi
ÇKA	Çok Katmanlı Algılayıcı
DBN	Derin İnanç Ağları
DCNN	Derin Konvolüsyonel Sinir Ağları
GPU	Grafik İşleme Birimi
GRU	Kapılı Yinelenen Birim
LSTM	Uzun Kısa Vadeli Hafıza Ağları
ReLU	Rectified Linear Units
SGI	Stokastik Gradyan İnişi
SVM	Destek Vektör Makineleri
RNN	Tekrarlayan Sinir Ağları
YSA	Yapay Sinir Ağları

1. GİRİŞ VE AMAÇ

Makine öğrenmesi algoritmalarının son yıllarda görüntü tanıma üzerindeki başarısı, elektronik tıbbi kayıtların ve tanısal görüntülemenin önemli bir şekilde arttığı zamanla örtüşmektedir. Bu durum, tıbbi görüntü analizine uygulanan makine öğrenmesi algoritmalarını ve Konvolüsyonel Sinir Ağlarını (CNN) ön plana çıkarmaktadır.

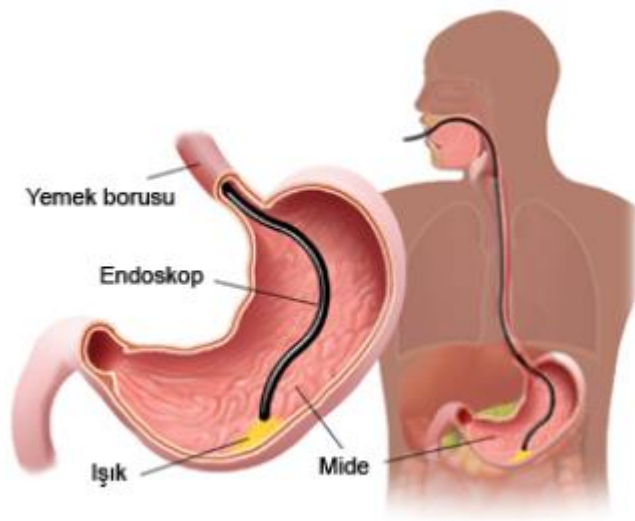
Bilgisayar algoritmaları, özellik çıkarımının fazla olduğu sistemlerde en uygun karar sınırlarını belirler. Bu tür sistemlerin tasarımında önemli bir adım ayırt edici özelliklerin görüntülerden çıkarılmasıdır. Bir sonraki mantıksal adım ise bilgisayarların eldeki problem için verileri en iyi şekilde temsil eden özellikleri öğrenmelerini sağlamaktır. Makine öğrenmesi uygulamalarında, verilere ait iyi tanımlanmış bilgilendirici özellikleri bulma ve öğrenme, tıbbi görüntü analizinde önemli bir rol oynar. Genel olarak bu özellikler, uzman kişilerin bu alandaki bilgilerinden veya makine öğrenmesi tekniğini kendi çalışmalarında kullanacak kişiler tarafından oluşturulur. Makine öğrenme teknikleri, işlenmemiş bir bilgiyi, uzman yardımı olmadan ve ön işlemden geçirilmeden işleyemezler. Makine öğrenmesinin aksine derin öğrenmede ise öğrenme işlemi, işlenmemiş veri üzerinde gerçekleşir. Bunun için gerekli olan bilgiyi farklı katmanlarda oluşturulan modellerde elde etmektedir.

Tıbbi görüntü yorumu çoğunlukla radyologlar veya doktorlar gibi uzman kişiler tarafından yapılmaktadır. Bununla birlikte patolojideki geniş çeşitlilik ve uzmanların potansiyel yorgunluğu göz önüne alındığında, araştırmacılar ve doktorlar bilgisayar desteklerinden tanı için yararlanmışlardır. Tıbbi görüntü analizindeki ilerleme oranı, tıbbi görüntüleme teknolojisindeki kadar hızlı olmamasına rağmen bu durum makine öğrenmesi teknikleriyle gelişmektedir (Shen vd., 2017).

Son yıllarda bilgisayarlı tomografi, manyetik rezonans görüntüleme, mamografi, endoskopi ve ultrason gibi tıbbi görüntüleme teknikleri erken teşhis ve tanı desteklemek için kullanılmaktadır.

Ucunda kamera olan, kendine ait özel bir ışık kaynağı bulunan, 8-12 milimetre çapındaki ve uzunluğu yaklaşık 110-120 santimetre olan yumuşak, kıvrılabilen bir alet

yardımı ile üst gastrointestinal sistem denilen yemek borusu, mide, onikiparmak bağırsağının incelenmesine olanak veren endoskopik işleme endoskopi denir (Anonim, 2019). Endoskopi bilgisayarlı tomografi ve manyetik rezonans görüntüleme gibi diğer tanısal yöntemlerle hastaya tam bir teşhis konulamadığı durumlarda yapılır. Endoskopi ile hastalıklı organın doğrudan görülmesi ve ihtiyaç halinde doku örneği alınması mümkündür (İstanbulu, 2019). Şekil 1.1. endoskopi işleminin genel görüntüsü gösterilmiştir (Anonim, 2018).



Şekil 1.1. Endoskopi işleminin genel görüntüsü (Anonim, 2018)

Endoskopi, mide poliplerinin taranması için standart bir yöntem olarak kabul edilir. Mide polipleri, farklı nedenlerle yapılan endoskopi işlemleri sırasında tesadüf olarak saptanan ve nadir görülen lezyonlardır. Endoskopik işlemler sırasında poliplerin görülme oranı %2-6 olarak bilinmektedir. Polip görülme yaşı 22-87 arasında değişebilir. Ancak 60 yaş üstü hastalarda polip bulunma olasılığının arttığı gözlenmiştir (Soytürk vd., 2012).

Poliplerin çoğu başlangıçta iyi huylu olmasına rağmen zamanla kötü huylu da olabilir. Bu nedenle, poliplerin erken evrelerinde saptanması çok önemlidir. Endoskopi yoluyla genel polip tespitinde, tamamen cihaza bağlı olarak yanlış polip tespitinin %25'ler civarında olduğu bilinmektedir (Shin ve Balasingham, 2017).

Endoskopi işleminin çözünürlüğü yüksek cihazlarla beraber kullanılması akıllı sistemlerin bu alanda kullanılmasına yönelik araştırmaları artırmıştır. Akıllı sistemler, polip tanısına birçok yönden yardımcı olabilir. Örneğin, mevcut polipleri ve polip olabilecek yerleri vurgulamak veya daha fazla dikkat gerektiren endoskopi videoları için otomatik rapor oluşturmak.

Sağlık sektörü diğer sektörlerden tamamen farklı olarak yüksek öncelikli bir sektördür. İnsanlar maliyetine bakmaksızın en üst düzeyde bakım ve hizmet beklemektedir. Bu yüzden gerçek dünya uygulamalarında derin öğrenmenin, tıbbi görüntüleme için heyecan verici, doğru çözümler ürettiği ve sağlık sektöründe gelecek uygulamalar için önemli bir yöntem olduğu görülmektedir.

Bu tez çalışmasında son yıllarda mühendislik alanında popüler olan derin öğrenme algoritmalarından biri olan CNN yöntemi kullanılarak endoskopi görüntülerindeki polipli ve normal görüntülerin sınıflandırılmasını yaparak hekimlere destek sisteminin tasarımını ortaya koymaktadır.

2. LİTERATÜR ARAŞTIRMASI

Görüntü tanıma işlemlerinde kullanılan makine öğrenmesi algoritmalarının son zamanlardaki başarısı, özellikle medikal alandaki tanısal görüntülemenin önemli bir şekilde arttığı zamana denk gelmektedir. Bu durum tıbbi görüntü analizine uygulanan, CNN'ye odaklanan ve alanın klinik yönlerini vurgulayan makine öğrenme algoritmalarını tanıtmaktadır. Makine öğrenmesinin medikal veri çağındaki bir avantajı, verilerdeki önemli hiyerarşik ilişkilerin manuel ile tanıma olmadan, algoritmik olarak keşfedilir olmasıdır.

Makine öğrenmesinin bir alt dalı olan derin öğrenme algortimalarının kullanımını hızla artmakta ve çeşitli tıbbi uygulamalarda performansın iyileşmesini sağlamaktadır. Derin öğrenme anatomik ve hücrel yapıların tespiti, doku segmentasyonu, bilgisayar destekli hastalık tanısı gibi uygulamalarda kullanılmaktadır.

Shin ve Balasingham (2017) çalışmalarında, görüntülerin şekil ve renk özelliklerini çıkarıp Destek Vektör Makineleri (SVM) ve CNN kullanılarak bu iki sınıflandırma yöntemlerini karşılaştırmışlardır. Bu çalışma için üç tane konvolüsyon katmanı ve adamax optimizasyon fonksiyonu kullanılmıştır. Ayrıca öğrenme oranı 0.002 ve devir sayısı da 200 olarak belirlenmiştir. Aynı veri setinde benzer resimler olduğu ve abartılı sonuçlar elde edilebileceği için; CVC-Clinic, ETIS-Larib ve Asu-Maya veri setleri kullanılmıştır. CVC-Clinic veri setinde 612 görüntü eğitim için, ETIS-Larib 196 ve Asu-Maya veri setindeki 170 görüntü test için kullanılmıştır. Çalışma sonunda SVM ile %84, CNN ile %91 doğruluk elde edilmiştir.

Ribeiro vd. (2016) çalışmalarında, CNN derin öğrenme algoritması kullanarak polip sınıflandırması üzerine çalışmışlardır. Bunun için 18 hastadan 25 sağlıklı görüntü ve 56 hastadan anormal görüntüler almışlardır. Veri sayısının artması eğitimi geliştireceğinden şekilleri döndürerek ve boyutunu değiştirerek toplamda 800 resim elde etmişlerdir. Bu çalışmada sınıflandırmadaki filtre boyutunun, sayısının ve tam bağlı katmandaki çıkış birimlerinin sayısının etkisini değerlendirmek için farklı mimariler test edilmiştir.

Yixuan ve Max (2017) çalışmalarında kapsül endoskopiden alınan görüntüleri kullanmışlardır. Kapsül endoskopi; kapsülün yutulması ve bu cihazın sindirim sisteminde

hareket ederek görüntüler alması ve sonunda hastanın bu kapsülü dışkısı ile atması işlemidir (Taviloğlu, 2017). Bu çalışmanın amacı kapsüldeki polip görüntülerinin hekimlere otomatik olarak tanımlarını sağlamayı amaçlamaktadır. Kapsül görüntüleri için ortalama tanınma doğruluğu %98 olarak bulunmuştur.

Zou vd. (2015) çalışmalarında, Evrimsel Sinir Ağı (DCNN) modelini kullanarak kapsül endoskopisi görüntülerindeki sindirim organlarının sınıflandırma problemini incelemiştir. Bu çalışma için bir milyon görüntü kullanılmıştır. Eğitim için 60 bin, test için 15 bin görüntü rastgele seçilmiştir. Mide, ince bağırsak ve kolon için, eğitimde 20 bin resim ve her bir sınıfın testi için beş bin resim bulunmaktadır. Öğrenme oranı 0.01 olarak belirlenmiştir. Sistemi eğitmek için Skolastik Gradyan İnişinde (SGI) tam bağlı katmanların her biri 64 nörona sahiptir. SVM yöntemi ile %90.31, CNN ile %95.52 doğruluk elde edilmiştir.

Şengür vd. (2010) çalışmalarında, kolonoskopik video görüntülerindeki poliplerin yerlerini belirlemek için dalgacık dönüşümü eş oluşum matrislerinden çıkarılan matrisler ile Yapay Sinir Ağları (YSA) sınıflandırıcısı kullanılmıştır. YSA mimarisi üç katmanlı olarak oluşturulmuştur. Giriş katmanı 36, ara katman 10, çıkış katmanı ise bir nörondan oluşmaktadır. Aktivasyon fonksiyonu olarak tanjant ve sigmoid kullanılmıştır. Öğrenme oranı ise 0.8 olarak belirlenmiştir. Yapılan çalışma sonucunda duyarlılık %90.2 ve özgüllük %88.7 olarak bulunmuştur.

Kaçmaz ve Yılmaz (2018) çalışmalarında, 100 tane normal ve 100 tane ülserli görüntü kullanmışlardır. Ülserli görüntüler SVM ve en yakın komşuluk matrisi kullanılarak ayırt edilmiştir. Kolonoskopi görüntülerden sadece 5 özelliğin çıkarılmasıyla %95 doğruluk elde edilmiştir.

Sarraf ve Tofighi (2016) çalışmalarında, alzheimer beynini normal ve sağlıklı bir beyinden ayırmak için derin öğrenme yöntemi olan CNN kullanmışlardır. Bu çalışmada, 24 kadın ve 19 erkekten oluşan, 28 Alzheimer hastası ve 15 sağlıklı bireye ait görüntüler Alzheimer Hastalığı Nörogörüntüleme Girişimi veri tabanından seçilmiştir. Veriler üç bölüme ayrılmıştır. Eğitim için % 60, doğrulama için % 20 ve test için % 20 oranında ayrılmıştır. Bu yöntemi kullanarak sınıflandırma doğruluk oranı %96,85 bulunmuştur.

Suzuki vd. (2016) çalışmalarında, mamografide ki kitleleri belirlemek için derin öğrenme yöntemlerinden Derin Konvolüsyonel Sinir Ağları (DCNN) kullanmışlardır. Çalışmada, ağırlıklı sekiz katman oluşturmuşlardır. Bu katmanların beş tanesi konvolüsyon katmanı, üç tanesi de tam bağlantılı katmandır. Çalışma, kitleye sahip 99 ve sağlıklı 99 görüntü olmak üzere toplamda 198 mamografi görüntüleri üzerinde gerçekleştirilmiştir. Kitle sınıflandırma doğruluğu %89,90 bulunmuştur.

Abdel-Zaher ve Eldeib (2016), çalışmalarında meme kanserini tespit etmek için Derin İnanç Ağları (DBN) yöntemini ve Wisconsin Meme Kanseri Veri Kümesi (WBCD) veri seti testini kullanmışlardır. Sınıflandırma doğruluk oranı %99,68 bulunmuştur.

Doğan ve Türkoğlu (2018) çalışmalarında, derin öğrenme yaklaşımlarından AlexNet, Vgg16, Vgg19, ResNet50, GoogleNet gibi derin öğrenme algoritmaları kullanmış ve bunların başarımları değerlendirilmiştir. Çalışma için UCI-Machine Learning Repository yaprak görüntü veri tabanı kullanılmıştır. 32 sınıf ve her bir görüntü sınıfı için yaklaşık 60 adet görüntü kullanılmıştır. Derin öğrenmede veri sayısının fazla olması başarımları artırdığından her bir sınıftaki görüntüler tersleme ve yansıma yapılarak, her sınıftaki görüntü sayısı 240 adet olacak şekilde artırılmıştır. Toplamda ise 7628 yaprak görüntüsü kullanılmıştır. Görüntülerin %80 eğitim, %20 test için kullanılmıştır. Çalışma sonunda AlexNet algoritmasını için %97,72, GoogleNet %97,77; Vgg16 %99,12; Vgg19 %98,36 ve ResNet50 için %99,15 başarı elde edilmiştir.

Yang vd. (2016) çalışmalarında, SVM ve DCNN'nin performans karşılaştırmasını yapmışlardır. Bunun için böbrekteki 334 tümörlü, 243 normal hücre görüntüsünü kullanmışlardır. Resim boyutları 300×300 pikseldir. Veri sayısını artırmak için 577 görüntü kırılıp veya döndürülerek toplamda 21.349 görüntü elde edilmiştir. Bu veri setinin 14.233 örneğini eğitim, 3.558 örneğini doğrulama ve 3.558 örneği test için ayrılmıştır. DCNN sınıflandırmasında ara katman sayısı beş ve yedi olarak iki şekilde test edilmiştir. SVM'nin doğruluk oranı %85, DCNN-5 %97.37 ve DCNN-7 %97.91 olarak bulunmuştur.

Dhungel vd. (2015) çalışmalarında, mamografide kitleleri belirlemek için derin öğrenme yöntemlerini ve random forecast sınıflandırma yöntemini kullanmışlardır. Birinci

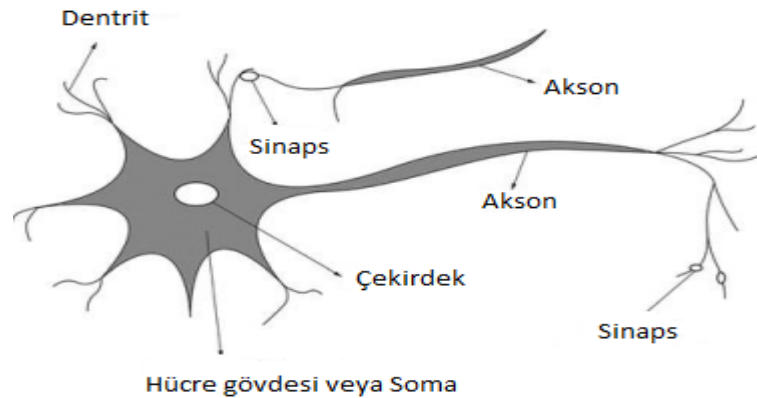
aşamalı sınıflandırıcı, iki ara katmanlı derin CNN kullanılmak üzere kitleye sahip bölgeleri seçen çok ölçekli bir DBN'den oluşur. Bu derin öğrenme analizinden sonra kalan bölgeler, seçilen bölgelerden elde edilen morfolojik ve doku özelliklerini kullanan iki seviyeli random forecast sınıflandırıcıları tarafından işlenir. Bu yaklaşım, yüksek doğru pozitif oranını korurken, yanlış pozitif bölgelerin azaldığını göstermektedir.

2.1. Yapay Sinir Ağları

Yapay Sinir Ağları canlı organizmalarda bulunan biyolojik sinir yapısından esinlenerek öğrenme sürecinin matematiksel olarak modellenmesi sonucu ortaya çıkmıştır.

2.1.1. Biyolojik sinir hücresinin yapısı

İnsan beyni nöron olarak adlandırılan yaklaşık 100.000.000.000 sinir hücresinden oluşmaktadır. Biyolojik sinir ağını oluşturan nöronların temel elemanları soma, dentrit ve aksondur ve Şekil 2.1 de sinir hücresinin şekli gösterilmiştir (Pratap, 2013).

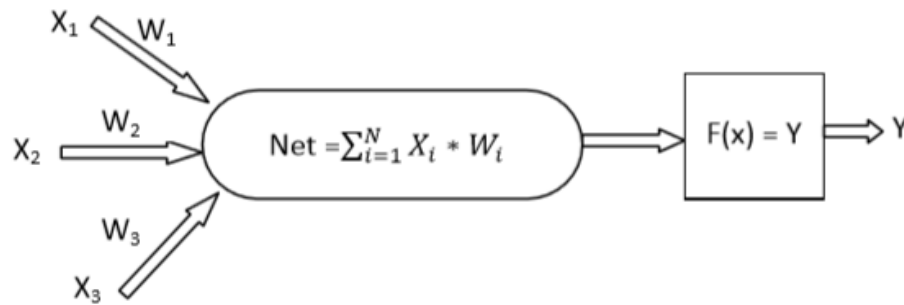


Şekil 2.1. Sinir hücresi (Pratap, 2013)

Soma olarak adlandırılan hücre gövdesi, hücre etkinliklerini yönetir ve hücreyi kontrol eder. Sinir hücresine diğer sinir hücrelerinden gelen bilgiler dentritler aracılığıyla, iletim hatları olarak kullanılan uzun fiberlerden oluşmuş aksonlarla iletilir ve hücre gövdesine taşınır. Aksonlar ise, hücre gövdesindeki bilgiyi diğer nöronların dentritlerine taşımakla sorumludur. Sinir sistemi yapısı nöronlardan meydana gelir ve bu yapı bilgisayarla örtüşecek şekilde tasarlanırsa insan beyninin özelliklerine sahip bilgisayar modeli oluşturulmuş olur.

2.1.2. Yapay sinir hücresinin yapısı

Biyolojik sinir ağlarının içinde bulunduğu sinir hücresi gibi yapay sinir ağlarının da yapay sinir hücreleri vardır. Yapay bir nöron hücresi girdi, girdilerin ağırlıkları, toplama fonksiyonu, aktivasyon fonksiyonu ve çıktı olmak üzere beş bölümden oluşmaktadır. Şekil 2.2 de bu bölümler gösterilmiştir (Çağıröğlü, 2019).



Şekil 2.2. Yapay bir nöron hücresi (Çağıröğlü, 2019)

Girdiler: Yapay sinir hücresine dışarıdan gelen bilgiler girdi olarak adlandırılır. Gelen bu veriler toplanmak üzere nöron çekirdeğine gönderilir.

Ağırlıklar: Yapay sinir hücresine gelen bilginin önemini ve çıktı üzerindeki etkisini göstermektedir.

Toplama Fonksiyonu: Yapay sinir hücresine gelen girdileri ağırlıklarla çarpıp toplayarak o hücrenin net girdisini hesaplayan bir fonksiyondur.

Aktivasyon Fonksiyonu: Bu fonksiyon hücreye gelen net girdiyi işleyerek, hücreye bu girdiye karşılık üreteceği çıktığı belirler.

Çıktılar: Aktivasyon fonksiyonları tarafından belirlenen çıktı değeridir. Bu değer yapay sinir ağının çıktısı olarak dış dünyaya ya da hücre kendi çıktısını kendisine girdi olarak da gönderebilir.

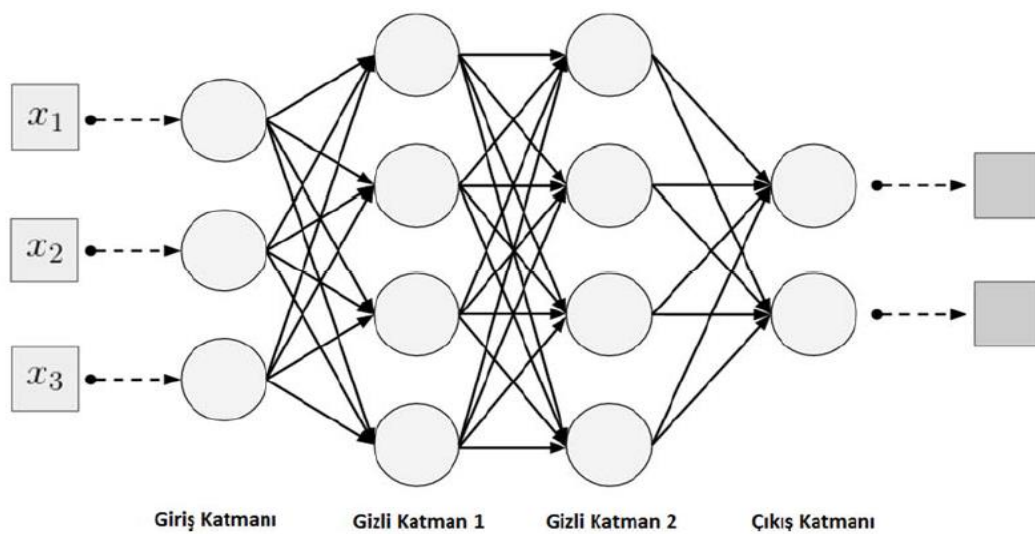
Bütün sinir ağı modellerinde x_i giriş değerleri, w_i sinir eklemi ağırlık katsayıları ile çarpılıp toplanır. Toplama birimi biyoloji nöronun gövdesine uygun olup ağırlıklandırılmış girişler ile toplanır ve çıkışı verir. Çizelge 2.1 de biyolojik ve yapay sinir ağları arasındaki ilişki gösterilmiştir.

Çizelge 2.1. Biyolojik ve yapay sinir ağları arasındaki ilişki

SİNİR SİSTEMİ	YAPAY SİNİR AĞI
NÖRON	İŞLEM ELEMANI (X_1, X_2, \dots, X_n)
DENTRİT	TOPLAMA FONKSİYONU
HÜCRE GÖVDESİ	AKTİVASYON FONKSİYONU
AKSON	ÇIKTI
SİNAPS	AĞIRLIKLAR

2.1.3. Yapay sinir ağı modeli ve çok katmanlı algılayıcı

Tek katmanlı yapay sinir ağları doğrusal olan olayları çözebilmelerine karşın, doğrusal olmayan XOR gibi problemlere çözüm üretememektedir. Doğrusal olaylar çıktı verilerini tek bir doğru ile iki veya daha fazla sınıfa ayırmak olarak tanımlanabilir. Günümüzde yaygın olarak kullanılan çok katmanlı algılayıcılar, XOR problemini çözmek için yapılan çalışmalar sonucu ortaya çıkmıştır. Çok katmanlı algılayıcı (ÇKA), eğitim aşamasında girdilerin ve girdilere karşılık üretilmesi beklenen çıktıların gösterildiği modeldir. Genelde bu ağ katmanları bir girdi katmanı, en az bir gizli katman ve bir de çıkış katmanı olmak üzere 3 farklı katmandan oluşur. Şekil 2.3 de ÇKA'nın yapısı gösterilmiştir (Anwer, 2017).



Şekil 2.3. Çok katmanlı yapay sinir ağları (Anwer, 2017)

Çok katmanlı algılayıcılar her bir katmanında matematiksel işlemler gerçekleşen gizli katmana sahip olmaları yönünden tek katmanlı algılayıcılardan ayrılır.

Dış dünyadan giriş katmanına gelen bilgiler, herhangi bir işleme tabi olmadan gizli katmana iletilir. Gizli katman sayısı modelden modele farklılık gösterebilir. Gizli katmandaki nöron sayısı giriş ve çıkış katmanındaki nöron sayısından bağımsızdır. Birden fazla gizli katmana sahip olan modellerde de gizli katmanın kendi aralarındaki nöron sayıları farklı olabilir. Gizli katmanların ve bu katmanlarda nöron sayılarının artması hesaplama karmaşıklığını ve süresini artırmasına rağmen yapay sinir ağının daha karmaşık problemlerin çözümünde kullanılmasını sağlar.

2.1.4. Yapay sinir ağlarında öğrenme

Öğrenme, sinir ağlarının temelini oluşturmaktadır. Öğrenme işleminde başlangıçta giriş katmanına girdiler verilir ve bu girdilere karşılık çıktı değerleri elde edilir. Amaç en optimum değerlerle problemi çözmek için ağın en uygun ağırlık değerlerini ve yapısını bulmaktır. Ağa verilen girdi değerleri için çıktı verilerinin doğru bir şekilde elde edilmesi açısından ağırlık değerlerinin doğru olması gerekir. Bu amaç ile başlangıçta ağırlık değerleri rastgele seçilir ve ağırlıklar her iterasyonda güncellenir. Ağırlık değerleri rastgele değil de uygun değerlerin seçilmesi ile olursa ağın öğrenmesi daha hızlı sağlanmış olur.

Yapay sinir ağında sisteme verilen giriş değerleri ile istenilen çıkış değerine ulaşılmışsa ağırlıklar güncellenmez ve bu ağırlıklar tutulur. Bu zaman içinde geçen süreye “öğrenme” denir. İstenilen ağırlık değerlerinin ne olduğu, hangi anlamları taşıdığı bilinmemektedir. Bu sebeple YSA'nın eylemlerini yorumlamak ve açıklamak mümkün olmadığından kendisi için “kara kutu” deyimini kullanılmaktadır.

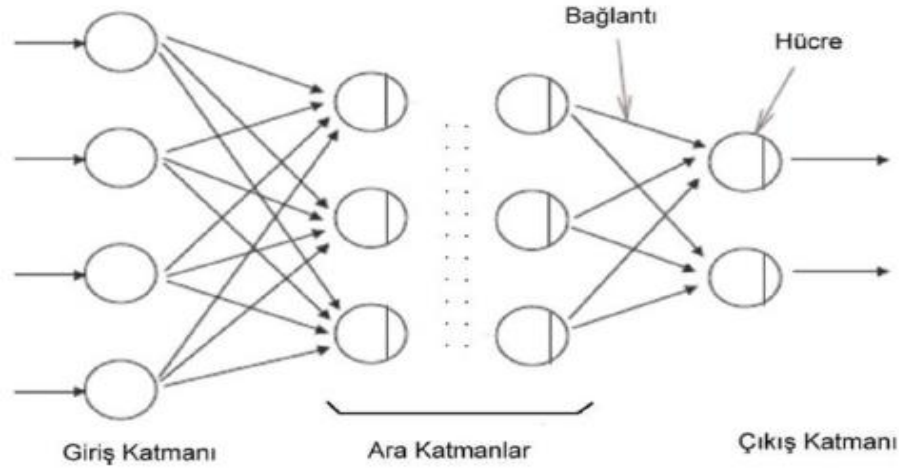
YSA öğrendikten sonra sisteme daha önceden verilmeyen giriş değerleri verilerek test edilir. Eğer ağ daha önce karşılaşmadığı yeni verilen bu örnekler için de doğru sonuç verebiliyorsa sinir ağı öğrenmiş demektir. Öğrenme işlemini test etmek için veriler eğitim ve test seti olarak iki gruba ayrılır. Eğitim setinde bulunan veriler ile öğrenmenin sağlanması için giriş ve çıkış değerleri sisteme sırasıyla verilir. Test setindeki veriler ise sisteme daha

önce verilmemiş, çıktısı bilinmesine rağmen sadece girdilerin sunulduğu veriler olup, bu veriler ağa iletilerek ağ başarımı ölçülür.

Aşırı uyum öğrenme sürecinde yaygın olarak karşılaşılan problemlerden biridir. Bu durum genellikle eğitim veri kümesinin çok küçük olduğu ve öğrenmenin çok uzun sürede gerçekleştirildiği durumlarda meydana gelir. Aşırı uyum sorunu ile karşılaşmamak için veriler eğitim ve doğrulama setine bölünür. Veri kümesinin eğitim ve test seti olarak ikiye bölünmesinde iyi bir başlangıç noktası olarak eldeki verilerin %80 ini eğitim setine, %20 sini ise doğrulama setine koymaktır (Brownlee, 2016).

2.1.5. İleri beslemeli yapay sinir ağları

İleri beslemeli ağlar çok katmanlı yapay sinir ağının bir türü olup, birimler arasındaki bağlantıların bir döngü oluşturmadığı, tek yönlü girdi sinyal akışına izin veren bir yapıdır. Şekil 2.4 de ileri beslemeli yapay sinir ağı modeli gösterilmiştir (Kutlu, 2018).

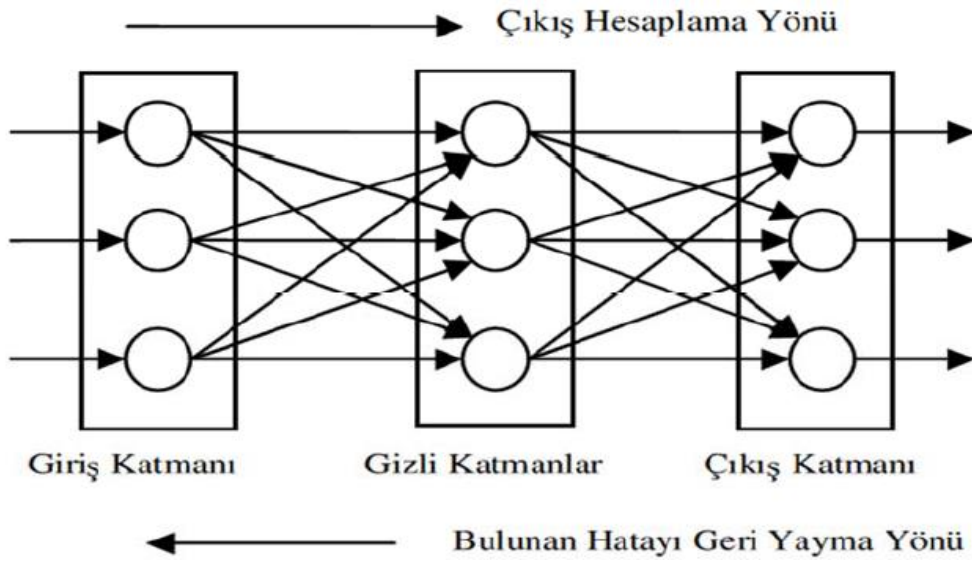


Şekil 2.4. İleri beslemeli yapay sinir ağı (Kutlu, 2018)

Bu yapı giren verilerin sadece tutulup hiçbir değişiklik yapılmadığı giriş katmanı, işlemlerin yapıldığı ve istenilen sonuca göre kendisini eğiten gizli katman ve çıkış değerini gösteren çıkış katmanı olmak üzere üç katmandan oluşur. YSA'ya hem eğitim için ayrılmış örnekler hem de bu örneklerden elde edilmesi gereken çıktılar verilmektedir. Ağ kendisine verilen örnek giriş ve çıkışlardan problem uzayını temsil eden bir çözüm uzayı oluşturur.

2.1.6. Geri beslemeli yapay sinir ağı

Geri beslemeli yapay sinir ağında bir nöronun çıktısı sadece kendinden sonra gelen nöron katmanına girdi olarak verilmez. Kendinden önceki katmana veya kendi katmanında bulunan herhangi bir nörona da girdi olarak bağlanabilir. Bu yapı ile geri beslemeli ağ doğrusal olmayan bir davranış göstermektedir. Şekil 2.5 da geri beslemeli ağ yapısı gösterilmiştir (Sivanandam, 2006).



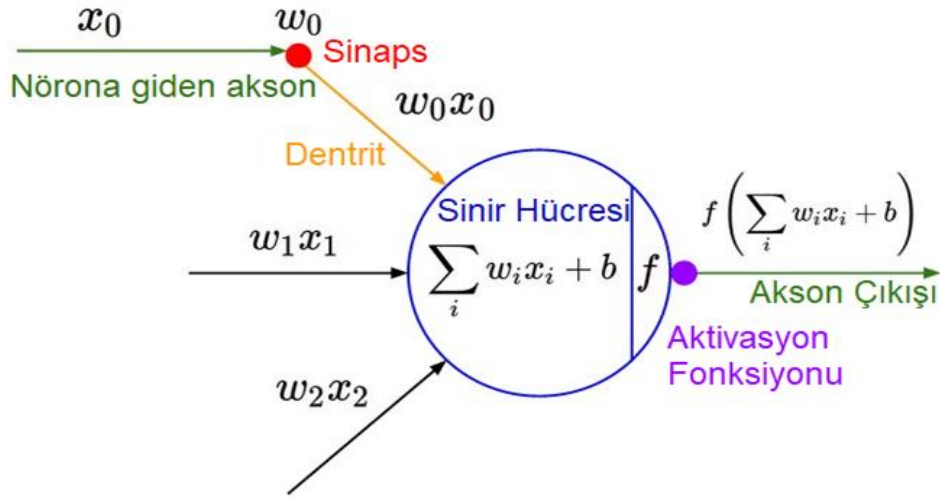
Şekil 2.5. Geri beslemeli ağ yapısı (Sivanandam, 2006)

Geri beslemeli yapay sinir ağında bir andaki çıkış hem o andaki hem de önceki girişleri yansıttığından dolayı özellikle tahmin uygulamaları için uygundur.

YSA'da öğrenme çoklu dönemler halinde gerçekleşir. Dönem, eğitim verisinde veriye bağlı çıktının bir kere işlenmesi olarak tanımlanır ve dönem süresi eğitim verisinin boyutuna ve ağ mimarisine bağlıdır. Her dönem ağın ürettiği çıktı değeri, ağın beklenen çıktı değeri ile kıyaslanır. Bu iki değer arasındaki fark hata değeridir ve oluşan toplam hatayı bulmak için bütün hataların toplanması gerekir. Bazı hata değerleri negatif olacağından tüm hata değerlerinin kareleri toplanır ve toplamın karekökü alınır. Hata değeri tolere edilebilir bir düzeye geldiğinde, bazı minimum düzeylerde başarı elde edilmiş olduğundan eğitim bu noktada sonlandırılır.

2.1.7. Aktivasyon yöntemleri

Konvolüsyon katmanından sonra aktivasyon fonksiyonlarından biri uygulanır. Aktivasyon fonksiyonu uygulanmazsa sinir ağı sınırlı öğrenme yetisine sahip doğrusal bir fonksiyon olur. Doğrusal olmayan işlemler için de sonuç elde edebilmek için çoğunlukla aktivasyon fonksiyonu doğrusal olmayan bir fonksiyon seçilir. Şekil 2.6 de x girdisi, w ağırlığı ve ağına çıkışına gönderilen $f(x)$ aktivasyon işlemi gösterilmiştir (Kızrak, 2019). Elde edilen sonuç ya bir çıkış ya da başka bir katmanın girişi olur. Aktivasyon fonksiyonu olarak genellikle sigmoid, tanh ve ReLU kullanılmaktadır.



Şekil 2.6. Aktivasyon fonksiyonu (Kızrak, 2019)

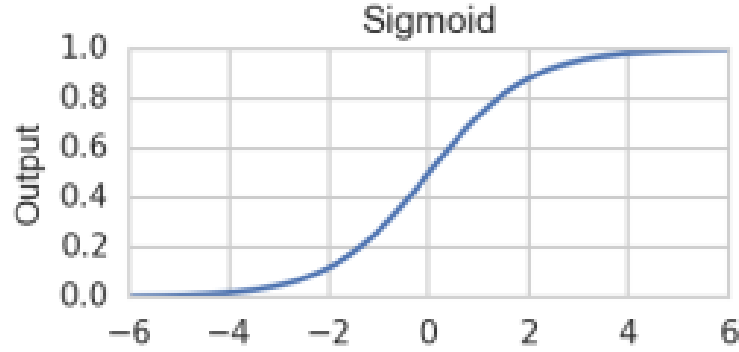
2.1.7.1. Sigmoid aktivasyon fonksiyonu

Sigmoid fonksiyonu sürekli, türevi alınabilir ve doğrusal olmayan bir fonksiyondur. Bu fonksiyon girdi değerleri için çıkışını 0 ile 1 arasında basit olasılıklara dönüştüren bir fonksiyondur. Yalnız sigmoid fonksiyonunun türev değerleri çok küçük olur ve 0'a yakınsar ve bu durum da öğrenme olayı minimum düzeyde gerçekleşir.

Yavaş bir öğrenme olayı gerçekleştiğinde ise ağ modelinden elde edilecek maksimum verim alınamaz. Denklemi ise aşağıdaki gibidir.

$$F(x) = \frac{1}{1 + e^{-x}} \quad (2.1)$$

Sigmoid fonksiyonu 0 ve 1 arasındaki y değerleri için tanımlı olup x=0 ekseninden önce 0'a sonrasında ise 1'e yakınsamaktadır.



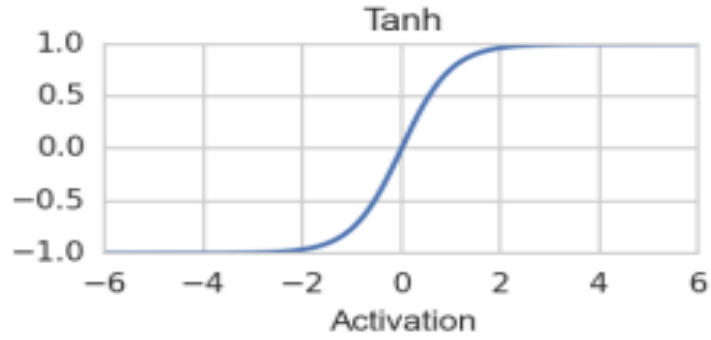
Şekil 2.7 Sigmoid aktivasyon fonksiyonu (Çarkacı, 2018)

2.1.7.2. Tanh aktivasyon fonksiyonu

Sigmoid fonksiyonuna çok benzer bir yapıya sahiptir. Sigmoid fonksiyonunda çıkış değerleri 0 ile 1 arasında değişirken tanjant fonksiyonunun çıkış değerleri -1 ile 1 arasında değişmektedir. Sigmoid fonksiyonuna göre avantajı ise türevinin daha dik olmasıdır. Bu sayede daha fazla değer alabilir ve bu değerlerle daha hızlı öğrenme ve sınıflama işlemi gerçekleşir. Denklemi aşağıdaki gibidir.

$$F(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.2)$$

Tanh fonksiyonunda x değerleri yüksekse +1'e, düşükse -1'e yakınsamaktadır. Genel olarak sıfır değer odaklı çalışır.



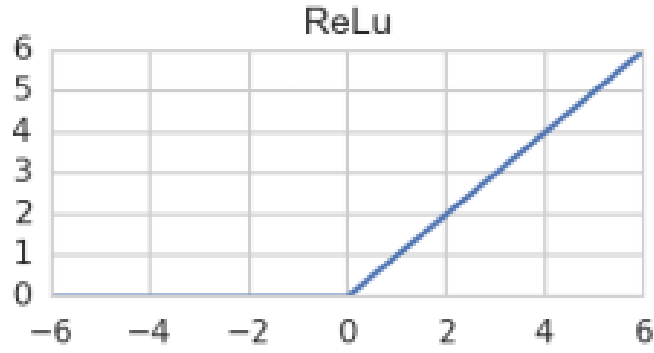
Şekil 2.8 Tanh aktivasyon fonksiyonu (Çarkacı, 2018)

2.1.7.3. ReLU aktivasyon fonksiyonu

Son yıllarda özellikle derin öğrenmede en çok tercih edilen ReLU (Rectified Linear Unit) aktivasyon fonksiyonudur. ReLU ile eğitilen CNN, tanh ile eğitilen CNN'den birkaç kat daha hızlıdır (Krizhevsky vd., 2012). Bu fonksiyon, giriş değeri sadece belirli bir değerin üzerinde olduğunda çıkış değeri üretir. Girdi değeri sıfırın altında olursa, çıkış sıfırdır, ancak girdi belirli bir değerin üzerinde olursa çıkış değeri üretir. Denklemi aşağıdaki gibidir.

$$F(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases} \quad (2.3)$$

Genellikle aktivasyon fonksiyonu olarak sigmoid ve ReLU fonksiyonları seçilir. Aktivasyon fonksiyonu seçilirken dikkat edilmesi nokta fonksiyonların türevinin sonucudur. Çünkü ÇKA'da öğrenilen ağırlıklar aktivasyon fonksiyonunun türevine bağlı olarak hesaplanır. Türev işlemi sonucunda sıfır değeri çıkarsa, kısmi türev bilgisinin kaybolmasına neden olur. Sigmoid fonksiyonun bire doğru yaklaşırken aldığı değerler türevde sıfıra doğru yaklaştığından ReLU'ya göre dezavantajlıdır (Kızrak, 2019).



Şekil 2.9. ReLU aktivasyon fonksiyonu (Çarkacı,2018)

2.1.8. Optimizasyon yöntemleri

YSA’da optimum öğrenme elde edilene kadar her adımda ağırlık değerlerinin güncellenmesi gerekir ve bu işlem belirli yöntemler dahilinde gerçekleşir. Ağırlık güncellemesinde “Dereceli Azalma (Gradient Descent)” yaygın kullanılan metotlardan birisidir (Yazan ve Talu, 2017). YSA’da ağırlıkların güncellenmesi için farklı gradyan tabanlı optimizasyon algoritmaları kullanılmaktadır. Bu algoritmaların birbirlerine göre avantaj ve dezavantajları vardır (Ruder, 2016). YSA’ da yaygın olarak ağırlık katsayılarını güncellemek için Adam, Stokastik Gradyan İnişi, Adagrad, Adamax gibi optimizasyon algoritmaları kullanılmaktadır.

2.1.8.1. Stokastik gradyan inişi optimizasyon yöntemi

Stokastik Gradyan İnişi (SGI), doğrusal sınıflandırıcıların öğreniminde kullanılan etkili bir yaklaşımdır. Bu yaklaşım makine öğrenmesi alanında uzun süredir kullanılmasına rağmen son zamanlarda büyük veri öğrenme bağlamında dikkat çekmektedir (Anonim, 2017). SGI yönteminde, her bir eğitim seti için güncelleme işlemi gerçekleştirildiğinden hızlı çalışır ve yerel minimum değerine daha kısa sürede ulaşır (Yazan ve Talu, 2017). SGI’nın en büyük avantajı, eğitim seti boyutu artmasına rağmen çalışma süresinin artmamasıdır. Bu durum eğitim örneği sayısı ile paralel olan verimliliğini gösterir (Anonim, 2017).

2.1.8.2. Adagrad, RMSProp ve Adam optimizasyon yöntemleri

Adagrad optimizasyon yönteminde, dereceli azalma yöntemindeki sabit öğrenme katsayısından kaynaklanan sorunu gidermek için her adımda parametrelere bağlı olarak öğrenme katsayısı güncellenir (Duchi vd., 2011).

RMSProp, öğrenme oranını ölçeklendirmek için kare gradyanları kullanır.

Adam optimizasyon algoritması, bilgisayarla görme ve doğal dil işleme uygulamaları için SGI optimizasyon yönteminin bir uzantısıdır. Farklı ağırlıklar için bireysel öğrenme oranı hesapladığından dolayı uyarlanabilir öğrenme oranı yöntemi de denilebilir. Adam hızlı ve iyi sonuçlar aldığı için YSA'da popüler bir algoritmadır. Birçok avantajı vardır. Bunlar;

- Uygulaması basittir.
- Daha az hafızaya ihtiyaç duyar.
- Veri ve parametre açısından büyük olan problemler için uygundur.

Adam, problemlerde seyrek geçişi kaldırabilen bir optimizasyon algoritmasını sağlamak için Adagrad ve RMSProp algoritmalarının en iyi özelliklerini bir araya getirir. Her ikisine sahip olmak Adam'ı daha geniş işlevler için kullanmamızı sağlar.

RMSProp ölçeklendirilmiş gradyandaki momentumu kullanarak parametre güncellemesi yapmasına rağmen Adam güncellemeleri gradyanın ilk ve ikinci momentinin ortalamasını kullanarak doğrudan tahmin eder (Kingma ve Ba, 2014).

2.1.9. Hata fonksiyonları

Algoritmaların verilen verileri ne kadar iyi modellediğinin değerlendirilmesi için kullanılan bir yöntemdir. Hata fonksiyonları, tasarlanan modelin yaptığı tahmini değer, gerçek değerden ne kadar farklı olduğunu hesaplamaktadır. Tahmin edilen değerler, gerçek değerlerden ne kadar fazla uzaklaşırsa hata fonksiyonları da o denli artacaktır.

YSA'da algoritmalara uyan tek bir hata fonksiyonu yoktur. Uygulanılan sorunun etki alanına ve eşsiz bağlamına bağlı olarak değişir. Seçilen makine öğrenmesi algoritmaların

türü, türevlerin hesaplanmasındaki kolaylık ve belli bir değere kadar veri setindeki aykırı değerlerin yüzdesi gibi belirli bir sorun için hata fonksiyonunun seçiminde rol oynayan çeşitli faktörler vardır.

Genel olarak hata fonksiyonları öğrenme görevinin türüne göre ikiye ayrılır. Regresyon ve sınıflandırma hataları. Sınıflandırmada sonlu kategorik değerler kümesinden çıktı tahmin etmeye çalışılır. Regresyon da ise sürekli ve gerçek değerlerin tahmini ile ilgilenmektedir.

2.1.9.1. Ortalama karesel hata (Mean Squared Error)

Ortalama karesel hata, bir regresyon hata yöntemidir. Öngörülen ve gerçekleşen değerler arasındaki kare farkların ortalaması alınarak hesaplanır. Anlaşılması ve uygulanması kolaydır ve genellikle iyi çalışır. Yönlerine bakılmaksızın yalnızca ortalama hata büyüklüğü ile ilgilidir. Ancak kareler nedeniyle gerçek değerlerden uzak olan tahminler, daha az sapma tahminlerine kıyasla daha büyük sonuçlara neden olur. Ortalama karesel hata eşitliği Denklem 2.4 de gösterilmiştir.

$$C(W, b) = \frac{1}{2n} \sum_x \|y(x) - a\|^2 \quad (2.4)$$

Denklem 2.4 de $C(W, b)$; hata fonksiyonu, W ; ağdaki tüm ağırlıkların toplamı, b ; ağdaki tüm biasların toplamı, n ; eğitim girdilerinin toplam sayısı, x ; ağ girdisi, a ; çıktı vektörü ve $y(x)$; ağın beklenen çıktısıdır.

2.1.9.2. Çapraz entropi (Cross Entropi)

Çapraz entropi, doğru sınıfı tahmin etmek için gerçek ve öngörülen olasılık dağılımları arasındaki ortalama farkı özetleyen bir değer hesaplar. Çapraz entropi hatasında gerçek değerden ayrıldıkça öngörülen olasılık artar. Matematiksel olarak, maksimum olasılık çıkarım çerçevesi altında tercih edilen hata işlevidir. Sınıflandırma problemleri için en yaygın olan hata fonksiyonudur. Sınıflandırma problemleri spesifik bir olası çıktı değerleriyle çalışılır ve bundan dolayı ortalama karesel hata yeterince tanımlanamamıştır.

İkili çapraz entropi (binary cross entropi), ikili sınıflandırma problemleri için kullanılacak varsayılan kayıp fonksiyonudur. Hedef değerlerin $\{0,1\}$ kümesinde olduğu ikili sınıflandırma ile kullanılması amaçlanmıştır. Denklem 2.5 de çapraz entropi hata fonksiyonunun denklemini gösterilmiştir.

$$C = -\frac{1}{n} \sum_x [y \ln a + (1 - y) \ln(1 - a)] \quad (2.5)$$

Bu denklemde, n eğitim verilerinin toplam sayısı, x tüm eğitim girdileri, y ise beklenen çıktıdır.

2.2. Derin Öğrenme Modelleri

2.2.1. Derin öğrenme

Yapay zekâ; bilgisayarları, insanların ve hayvanların sahip olduğu düşünme, öğrenme, karar verme gibi görevleri yerine getirmekle ilgili bir kavramdır (Millington ve Funge, 2009). Bilgisayar bilimlerinde geniş bir araştırma alanıdır. Temelde yapay zekânın yapmaya çalıştığı, bilgisayarların düşünebilen cihazlar olmasını sağlamaktır. Yapay zekâ kendi içerisinde birçok alt kümeye ayrılır. Farklı alandaki bu alt kümeler, bu düşünebilen cihazları kendi yöntemleriyle farklı yaklaşımlar kullanarak geliştirmeye çalışırlar.

Yapay zekânın alt dallarından birisi makine öğrenmesi alanıdır. Temelde bu alanda hedeflenen, istatistiksel yöntemler kullanılarak elde edilen verilerden öğrenebilen yazılımlar ve bilgisayar sistemleri geliştirmektir. Bu makine öğrenmesi de kendi içerisinde alt kümelere ayrılır ve bu alt kümelere bir tanesi de derin öğrenmedir. Şekil 2.10.'da derin öğrenme makine öğrenmesi, yapay zekâ arasındaki ilişki gösterilmiştir (Kutlu, 2018).



Şekil 2.10. Derin öğrenme ve yapay zeka arasındaki ilişki (Kutlu, 2018)

Derin Öğrenme kullanım amacına bağlı olarak birçok şekilde tanımlanabilir. Temel olarak derin öğrenme, bilgiyi doğrusal olmayan bir şekilde işlemek için birçok katmana sahip makine öğrenmesinde gelişen bir daldır. Başka bir tanımla ise, çok katmanlı sinir ağlarını kullanarak öznelik öğrenimini sağlayan bir algoritmalar ailesidir.

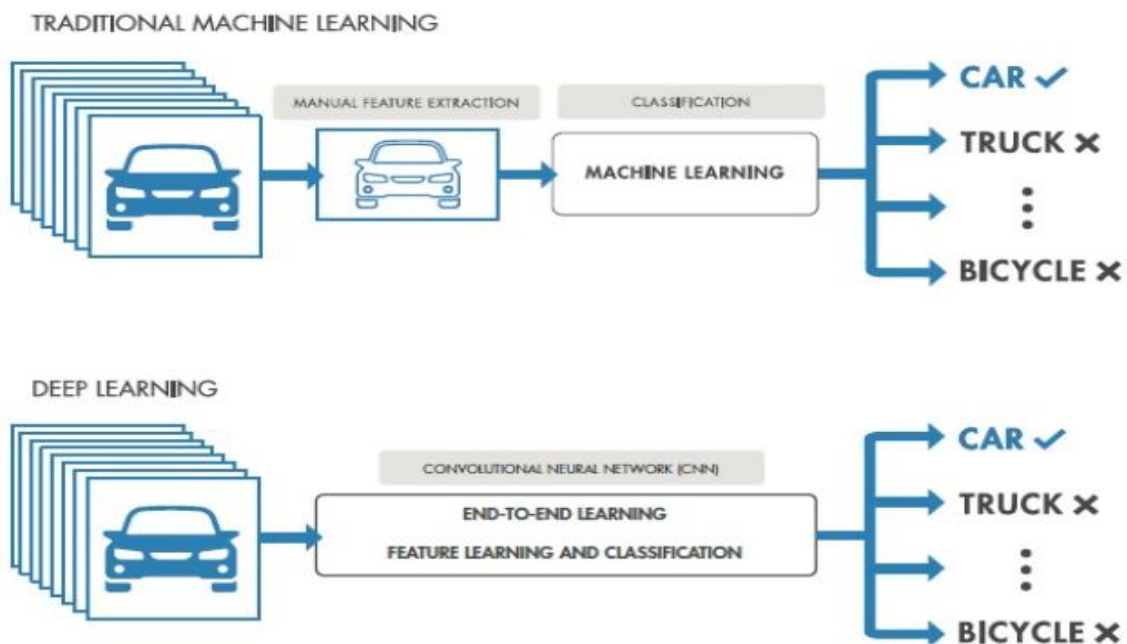
Derin öğrenme temelde YSA'yı kullanır. Derin öğrenmeyi YSA'dan ayıran başlıca unsur, derin sinir ağlarının YSA'ya oranla çok daha fazla sayıda katmandan oluşmalarıdır. İşlenecek verilerden daha çok özellik elde edilmesi ve öğrenimin daha iyi gerçekleşmesi için gizli katman sayısının artırılması ile yapılmaktadır (Bengio ve LeCun, 2007). Böylece yapay sinir ağlarından daha verimli sonuçlar alabilmek ve gizli katman sayısı artırılarak elde edilen derinliğin modelin performansına etkisine vurgulamak için “Derin Öğrenme” kavramı hayatımızda yaygınlaşmıştır.

2000'li yıllara doğru çeşitli problemlerden dolayı yapay zekâya dayalı öğrenme yaklaşımlarının önemi azalmaya ve giderek unutulmaya başlanmıştır (Schmidhuber, 2015). Eğitim kümesi olabilecek kadar çok verinin bulunmaması ve bu veriyi işleyecek donanımın yetersiz kalması bu duruma etkendi. Ancak son birkaç yılda, işlemci teknolojisindeki özellikle de Grafik İşlem Birimleri (GPU) alanındaki gelişmeler, daha büyük miktarda veriyi ve paralel işlemeyi çok sınırlı bir süre içinde incelememize olanak sağladığından derin öğrenmeye ilgiyi artırmıştır.

GPU'lar makine öğrenmesine katılan matematik işlemleri için çok uygundur. GPU'ların eğitim algoritmalarını önemli ölçüde hızlandırır. Bu nedenle artan eğitim veri boyutu artık sorun olmamaktadır.

2.2.2. Derin öğrenme ve makine öğrenmesi arasındaki farklar

Derin öğrenme, makine öğrenmesinin bir alt dalıdır. Makine öğrenmesi ile model oluşturabilmek için yapılması gereken ilk işlem özellik vektörünün oluşturulmasıdır. Bunun için alanında uzman kişilere ihtiyaç vardır. Ayrıca özellik vektörünün çıkarılması çok fazla zaman alır. Makine öğrenme teknikleri, herhangi bir işlem yapılmamış bilgiyi, uzman desteği olmadan ve ön işlemden geçirilmeden işleyemezler. Derin öğrenme bu problemleri ortadan kaldırır. Veriye konvolüsyon, havuzlama ve aktivasyon işlemini birçok katmanda tekrar ederek, ham veriden özellik çıkarımını yapıp öğrenme işlemini gerçekleştirir. Ayrıca makine öğrenmesinde küçük veri setleriyle iyi sonuçlar elde edilirken, derin öğrenme için daha büyük veri setlerine ihtiyaç vardır. Makine öğrenmesinde eğitim daha hızlıken, derin öğrenme yoğun hesaplama ihtiyacı duyar. Şekil 2.11 derin öğrenme ve makine öğrenmesi farkı gösterilmiştir (Çınar, 2017).



Şekil 2.11. Derin öğrenme ve makine öğrenmesi arasındaki fark (Çınar, 2017)

2.3. Konvolüsyonel Sinir Ağları ve Parametreleri

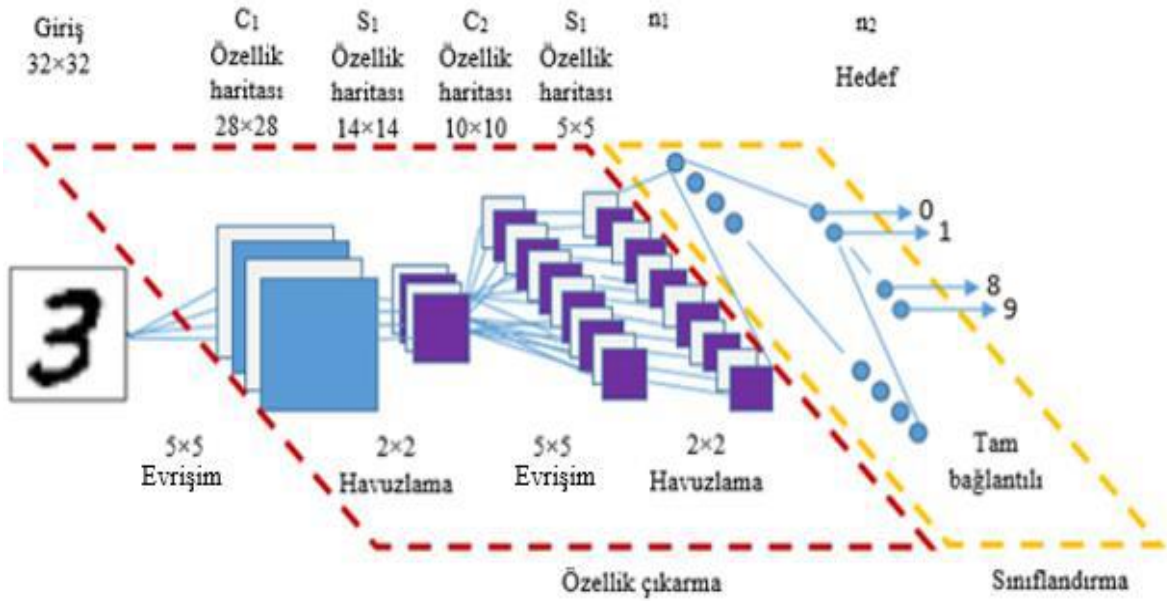
Konvolüsyonel sinir ağları (CNN- Convolutional Neural Networks), özellikle bilgisayarda görme ve görüntü işlemede kullanılacak derin öğrenme algoritmalarından biridir. Beynin görsel bilgiyi işleme tekniğinden esinlenerek ortaya çıkarılmıştır (Bengio, 2008).

CNN, 2012 yılından sonra derin öğrenme alanında sınıflandırma işlemindeki iyileştirmeler yaşandıktan sonra popülerlik kazanmaya başladı. Bu iyileştirmelerden sonra, bazı ileri teknoloji şirketleri çeşitli hizmetler için CNN'i kullanmaktadır. Amazon CNN'i ürün tavsiyeleri, Google fotoğraf araştırması ve Facebook da temel olarak otomatik etiketleme algoritmaları için kullanmaktadırlar (Dev, 2017).

CNN'ler görüntülerdeki karmaşık kalıpları tanımakta çok iyidir. Aslında CNN dediğimiz YSA'nın resim ve videoların eğitilmesi konusunda özelleşmiş bir halidir. Ama CNN'nin nasıl çalıştığı ilkesi, standart YSA'dan biraz farklıdır. CNN'de bir katmandaki her nöron, görüntünün sadece küçük bir bölümüne yoğunlaşır. Bir nöron aynı özelliği arar ve incelenen görüntünün başka bir bölümünde farklı özellikleri aramaya devam etmek için nöronların geri kalanına bırakır. CNN'ler genel matris çarpımı yerine, katmanların bir veya daha fazlasında evrişim işlemini kullanırlar.

Bir CNN modelin iki önemli avantajı vardır. Birincisi, öğrenilmesi gereken önemli ölçüdeki filtrelerin ağırlıklarını indirgemektir. Bu durum çok fazla nörona ihtiyaç duyulması ve bu nöronların öğrenebilmesi için çok sayıda ağırlığa sahip olmasından dolayı oluşabilecek birçok sorunu ortadan kaldırır. İkinci avantaj ise ağırlık paylaşımıdır. CNN ile öğrenilen parametreler bir sonraki katmana girdi olarak paylaşılabilir. Bu nedenle aynı ağırlıklar katmanda tekrar kullanılır, yeniden öğrenmeye gerek yoktur. Bu şekilde, daha fazla katman daha karmaşık özellikleri ve kalıpları öğrenir.

CNN genel olarak üç katmandan oluşur. Bunlar evrişim, havuzlama ve tamamen bağlı katmanlardır. Şekil 2.12 de CNN'nin genel yapısı gösterilmiştir (LeCun vd., 2010).



Şekil 2.12. Konvolüsyonel sinir ağı (LeCun vd., 2010)

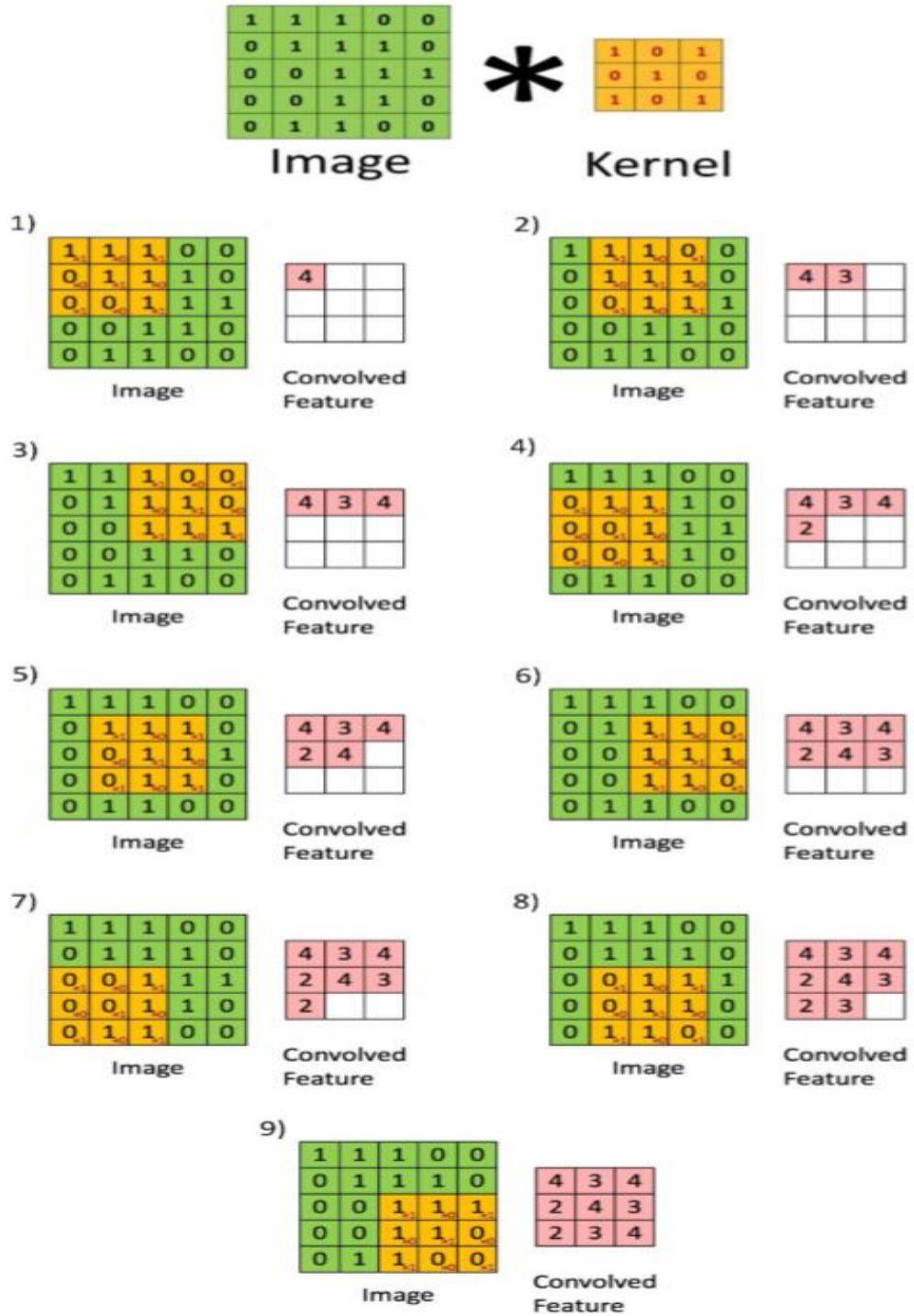
2.3.1. Konvolüsyon katmanı (Convolutional Layer)

Konvolüsyon katmanı, CNN'nin ilk gizli katmanıdır. Bu katmanda, bir görüntü filtrelenmiş görüntülerin bir yığını haline gelir. Konvolüsyon katmanı ağıdaki ağırlık sayısını azaltarak sistemin daha hızlı eğitilebilmesini sağlar. Filtreler, daha küçük bir veri seti üretmek ve onu bir sonraki katmana iletmek için giriş görüntüsü ile işlenir.

Konvolüsyon işleminde görüntü matrisi üzerinde 3x3, 5x5, 7x7, 9x9, 11x11 boyutundaki filtreler kullanılır. Belirtilen filtreler tüm görüntü matrisi üzerinde dolaşarak görüntüdeki öznitelikleri belirginleştirir ve görüntü boyutunda yeni bir görüntü matrisi elde edilir (Liu vd., 2015). Ancak kullanılan filtre boyutu büyüdükçe çıkış görüntüsünün boyutu azalacağı için, bilgi kaybı olacaktır. Bu yüzden genellikle en küçük filtre boyutu olan 3x3 kullanılır.

İlk başta filtre, görüntünün sol üst köşesine yerleştirilir. Resim ve filtre indisleri birbirleri ile çarpılır ve sonuçlar toplanır, elde edilen sonuç çıktı matrisinde depolanır. Daha sonra filtreyi bir piksel sağa kaydırıp işlem tekrarlanır. Giriş görüntüsünde birinci satır bittikten sonra ikinci satıra geçilir ve işlem tekrarlanır.

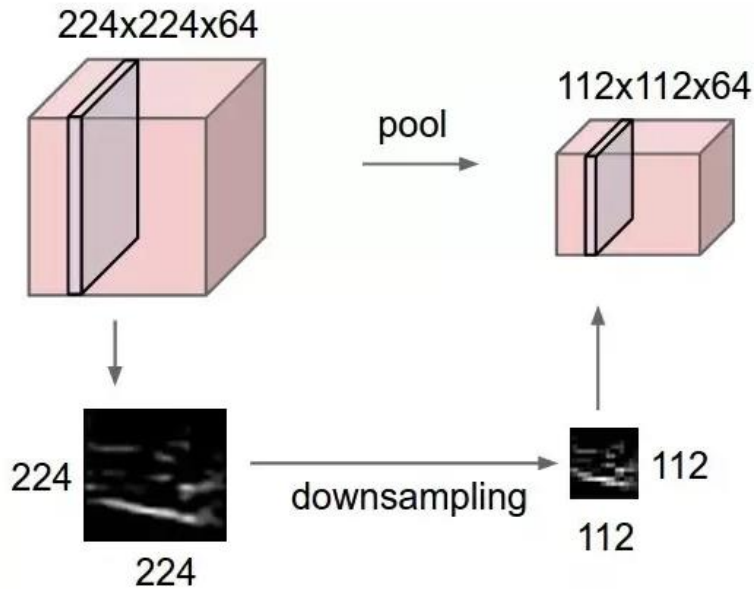
Tüm işlemler bittikten sonra çıktı matrisi elde edilmiş olur. Şekil 2.13 de 5*5 lik giriş verisine 3*3 lük filtrenin uygulanışı ve özellik haritasının çıkarımı gösterilmiştir (Çoşar, 2018).



Şekil 2.13. 5*5 lik giriş verisine 3*3 lük filtrenin uygulanışı ve özellik haritasının çıkarımı (Çoşar, 2018)

2.3.2. Alt örnekleme katmanı (Pooling Layer)

Havuzlama katmanı olarak bilinen bu katmanda genellikle konvolüsyon işlemi sonrasında gerçekleşen bir katmandır (Castelluccio vd., 2015). Ağdaki parametre ve hesaplama miktarını azaltmak için görüntünün uzamsal boyutunu aşamalı olarak azaltmak ve böylece aşırı uyumu kontrol etmek için uygulanan katmandır. Alt örnekleme katmanı, girişin her derinlik diliminde bağımsız olarak çalışır böylece derinlik boyutu değişmeden kalır. Çünkü derinlik boyutu resmin renkli kısmını temsil eden boyutudur. Şekil 2.14 de alt örnekleme işleminde derinlik boyutunun değişimi gösterilmiştir (Anonim, 2018).



Şekil 2.14. Alt örnekleme işleminde derinlik boyutunun değişimi (Anonim, 2018)

Konvolüsyon katmanında olduğu gibi alt örnekleme katmanında da filtreler kullanılır. Bu filtreler görüntü üzerinde belli bir adım atma değerine göre kaydırılarak görüntüdeki piksellerin en büyük değeri olarak (max pooling), en küçük değeri olarak (min pooling), değerlerin ortalamasını alarak (mean pooling) gerçekleştirilebilir. Daha iyi performans gösterdiği için max pooling işlemi kullanılır. Max pooling katmanı, bir sonraki katman için veri boyutunu azaltan ve bilgiyi bir katmandan diğerine aktarırken elde edilecek olan girdilerin daha yüksek bir gösterimini mümkün kılan katmandır.

Bunu yaparken, filtrenin temsil ettiği bölgelerin her biri için o bölgenin maksimumu alınır ve her elemanın orijinal girdideki maksimum bir bölge olduğu yeni bir çıktı matrisi oluşturulur.

Maksimum değeri elde edebilmek için eğitim sırasında yalnızca maksimum girdi sağlayan nöron ağırlıkları güncellenir. Şekil 2.15 de giriş boyutu 4x4 olan bir görüntüye 2x2 boyutlu filtre uygulanışı gösterilmiştir (İnik ve Ülker, 2017). Bu işlemde bir adım kaydırılarak elde edilen görüntü boyutu 3x3, iki adım kaydırılarak elde edilen görüntü boyutu 2x2 olur. Kaydırma olarak bahsedilen durum “Stride” olarak adlandırılır ve her adımda filtrenin hareket adım boyutu olarak tanımlanır. Stride değeri arttıkça özellik haritası küçülür. Alt örnekleme sonucu elde edilen görüntü boyutu Denklem 2.6 de gösterilmiştir.

$$\text{Elde Edilen Görüntünün Boyutu} = W_2 \times H_2 \times D_2 \quad (2.6)$$

$$W_2 = (H_1 - F) / A + 1$$

$$H_2 = (H_1 - F) / A + 1$$

$$D_2 = D_1$$

G1= Giriş görüntü boyutunun genişlik değeri

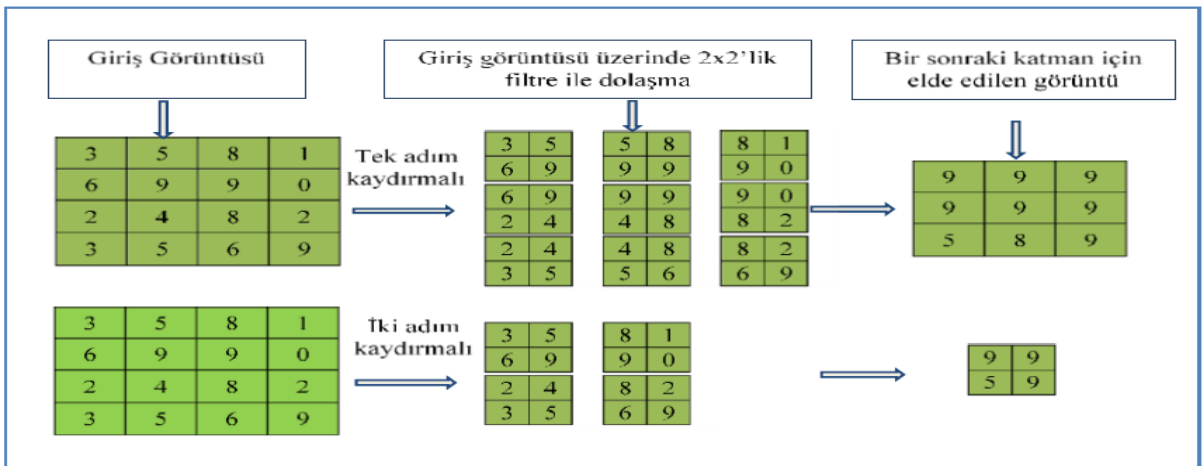
Y1= Giriş görüntü boyutunun yükseklik değeri

D1= Giriş görüntü boyutunun derinlik değeri

F=Filtre boyutu

A=Adım sayısı

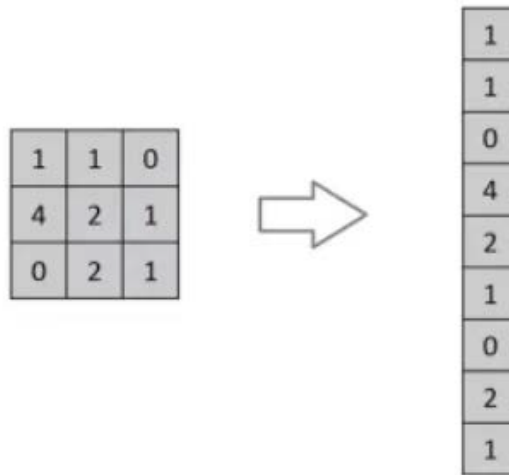
Pooling işleminde genellikle F=2 ve A=2 olarak seçilir.



Şekil 2.15. 5x5'lik giriş görüntüsüne 2x2 filtre ile bir ve iki adım kaymalı max pooling işleminin uygulanması (İnik ve Ülker, 2017)

2.3.3. Tam bağlantılı katman (Fully Connected Layer)

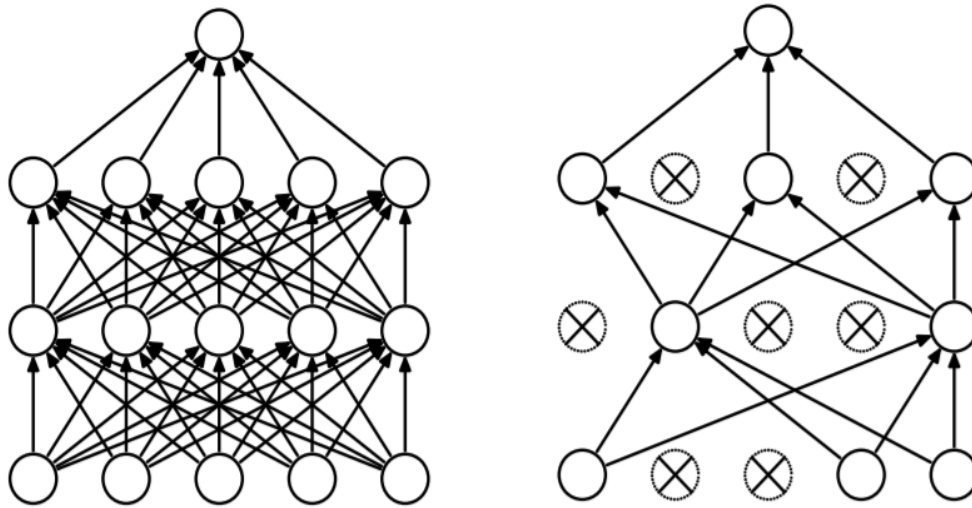
Konvolüsyon ve pooling katmanlarının ardından gelen ve “dense” olarak adlandırılan katmandır. Önceki katmanlardan elde edilen özelliklerden modelin sınıflandırma işlemi bu katmanda gerçekleşir. Tam bağlantılı katman, kendinden önceki katmandan özellik haritasını alır ve istenilen sayıda sınıf sayısı için işlenir. Ağın çıkışında elde edilecek değerler sayı değerleri olacağı için çok boyutlu veriler (matrisler) ile işlem yapıldıktan sonra bu katmanda boyut indirgeme yapılarak öznelik haritasından elde edilen veri tek boyutlu hale getirilir. Tek boyutlu veri sınıflandırıcıya girdi olarak verilir. Şekil 2.16 de tam bağlı katmanla tek boyuta düşürme işlemi gösterilmiştir (Kutlu, 2018).



Şekil 2.16. Tam bağlı katmanla tek boyuta düşürme işlemi (Kutlu, 2018)

2.3.4. Seyreltme katmanı (Dropout Layer)

Seyreltme katmanı yapay sinir ağları içerisinde verilerin ezberlenmesi probleminin önüne geçmek için kullanılan bir tekniktir (Srivastava vd., 2014). Dropout tekniği genelde tam bağlı katmanlarda sonra kullanılır. Dropout kullanılarak tam bağlı katmandaki bağlar koparılır. Dropout değeri probleme ve veri setine göre değişiklik göstermesine rağmen genelde 0,5 olarak alınır. Şekil 2.17 de normal bir yapay sinir ağı bağlantıları ve dropout uygulanmış ağ yapısı görülmektedir (Budhiraja, A., 2016).



Şekil 2.17. Yapay sinir ağına seyreltme tekniğinin uygulanması (Budhiraja, A., 2016)

2.3.5. Sınıflandırma katmanı (Classification Layer)

Tam bağlantılı katmandan sonra, sınıflandırma katmanı kullanılmaktadır. Sınıflandırma yapılacak olan nesne sayısı ile sınıflandırma katmanının çıkış değerinin aynı olması gerekmektedir (Uysal vd., 2018). Bu katmanda farklı sınıflandırıcılar kullanılmaktadır. Çoğunlukla başarısından dolayı softmax sınıflandırıcı tercih edilir (İnik ve Ülker, 2017). Softmax, çoklu sınıflandırmalar için sigmoid aktivasyon fonksiyonunun bir genellemesidir.

2.3.6. Eğitim dönemi sayısı (Epoch sayısı)

Bir modelin eğitimi sırasında, modelin ağırlıklarını güncellemek için tüm giriş verilerinin sinir ağı boyunca bir kere gidip gelmesidir. Her bir epoch da YSA öğrenir. Aynı girişleri çok fazla sayıda uygulamanın sonucu olarak, model kendini en az hata ile eğitebilir.

2.3.7. Mini-batch boyutu

Derin öğrenme çok büyük veri setlerinde gerçekleştirilir ve model eğitimi sırasında verilerin hepsi aynı anda eğitime katılmaz. Örneğin bir veri setinde 2000 tane örnek olduğunu varsayalım. Mini-batch boyutu 200 olursa, ilk 200 örnek alınır ve ağdan geçirilir.

Daha sonra ikinci 200 örnek alınır ve oda ağdan geçirilir. Bu işlem veri setindeki tüm örnekler ağdan geçirilinceye kadar devam eder. Bu şekilde veri setinin gruplar halinde işlenmesine mini-batch denir. Eğer mini-batch değeri çok küçük tutulursa doğruluk oranı düşer. Mini-batch boyutu 1 ile eğitim setinin boyutu arasında bir değer belirlenmelidir. Ayrıca bu değerın GPU belleğine sığması gerekiyor (Çarkacı, 2018). Bu nedenle 2 ve 2'nin katları şeklinde belirlenmelidir.

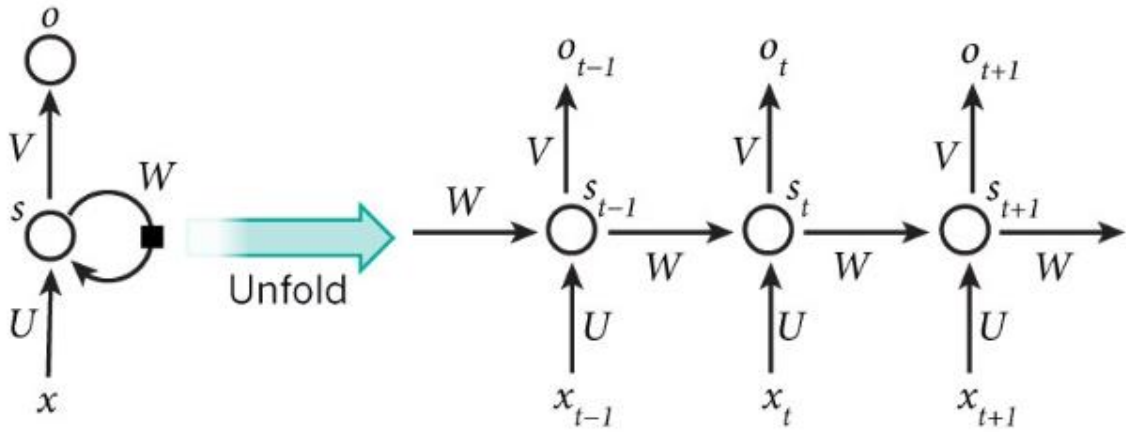
2.3.8. Öğrenme katsayısı (Learning Rate)

Öğrenme katsayısı, bir öğrenme işleminin hızı ve doğruluğu ile orantılı olup bunları kontrol eden bir değişkendir. Bu değişken ağırlıkların güncellenmesinde kullanılmaktadır. Derin öğrenmede ağırlıkların güncellenmesi geriye yayılım işlemi ile gerçekleşir. Geriye yayılım da ağırlıkların güncellenmesi işleminde, geriye doğru türev olarak fark bulunur. Daha sonra çıkan fark değeri öğrenme katsayısıyla çarpılır ve bulunan sonuç ağırlık değerlerinden çıkarılarak yeni ağırlık değeri hesaplanır.

Öğrenme katsayısı çok büyük seçilirse hata yüzeyinde büyük atlamalar meydana gelir ve bu yüzden öğrenmenin gerçekleşeceği dar alanlar atlanabilir. Çok küçük seçilmesi durumunda ise küçük adımlarla ilerleneceğinden öğrenme süresi zaman alır. Yapılan çalışmalarda öğrenme katsayısının 0.01 ile 0.9 arasında seçildiğinde iyi sonuç verdiği gözlenmiştir (Arı ve Berberler, 2017).

2.4. Tekrarlayan Sinir Ağları (RNN)

Tekrarlayan Sinir Ağları (RNN) önceki adımdan elde edilen çıkışın mevcut duruma girdi olarak beslendiği bir sinir ağıdır. Geleneksel sinir ağlarında, bütün girdiler ve çıktılar birbirinden bağımsızdır. Ancak bir cümlenin bir sonraki kelimesini tahmin etmek durumunda olduğu gibi örneklerde, önceki kelimeler zorunludur. Bu nedenle önceki kelimeleri hatırlama ihtiyacı vardır. Bu mantıkla RNN modeli ortaya çıkmıştır. Şekil 2.18 de RNN modelinin tam bir ağa açılması gösterilmiştir (Anonim,2017).



Şekil 2.18. Tekrarlayan sinir ağları (Anonim,2017)

X: t anındaki girdi

S: t anındaki gizli konum

U,V,W: ağırlık parametreleridir. Bu parametreler her zaman adımı için aynıdır.

RNN zaman serileri ve sıralı olan verileri modellemek için tasarlanabilirler. Örneğin, 5 3 2 5 3 2 5 3 gibi bir girdi olabilir. Girdi arka arkaya gelen 3 karakter olsun. Amaç ise 4. karakteri bulmaktır. Bu veri modelinde RNN eğitilebilir ve 5 3 2 den sonra 4. karakter olarak hangi sayının geleceğini tahmin etme üzerine modellenebilir.

RNN düğümler arasındaki bağlantıların geçici bir dizi boyunca yönlendirilmiş grafik oluşturduğu YSA sınıfıdır. İleri beslemeli ağlardan farklı olarak, RNN'ler girdi sınıflarını işlemek ve neyin hesaplandığı gibi tüm bilgileri hatırlayan bir belleğe sahiptirler. Çünkü belli bir düzende gelen giriş setinin çıktısı için bir anlamı vardır.

Çıktıyı üretmek için tüm girişlerde veya gizli katmanlarda aynı ağırlık değerlerini kullanırlar. Böylece her bir çıktının bir sonraki katmana girdi olarak verilmesiyle, bir önceki çıktının ezberlenmesi ve artan parametrelerin karmaşıklığı azaltılabilir.

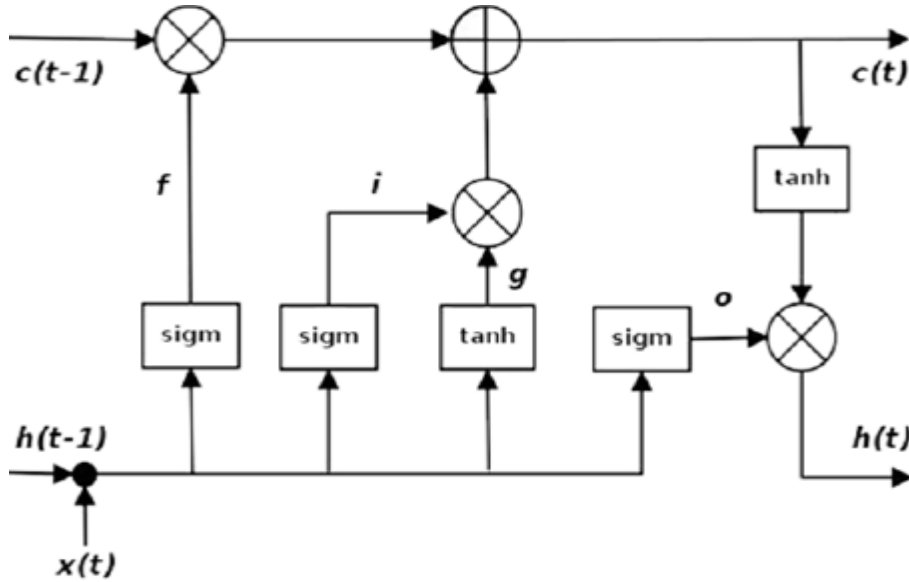
2.4.1. Uzun kısa vadeli hafıza ağları (LSTM)

Uzun Kısa Vadeli Hafıza Ağları (LSTM) derin öğrenme alanında kullanılan RNN mimarisidir. Yapay zekâ programlarının insan düşüncesini daha etkin bir şekilde taklit

etmesine yardımcı olabilecek belirli yapay bellek işlemlerinden faydalanmak için RNN kullanılır. RNN ise programın girdi alma ve çıktı oluşturma şekli için LSTM bloklarını kullanır.

RNN, belirli bir kelimeyi almak için de LSTM bloklarını kullanır. Ve hafızanın bu tip girdileri ayırmada ve kategorize etmede yararlı olabileceği bir dizgide değerlendirir. Genel olarak LSTM, RNN de öncü olarak kabul edilmiş ve yaygın bir kavramdır.

LSTM ağırlıklı girdiler, aktivasyon fonksiyonları, önceki bloklardan gelen girdiler ve nihai çıktılar gibi çeşitli bileşenlere sahip karmaşık bir ünedir. Şekil 2.19 da LSTM yapısı gösterilmiştir (Dev, 2017).

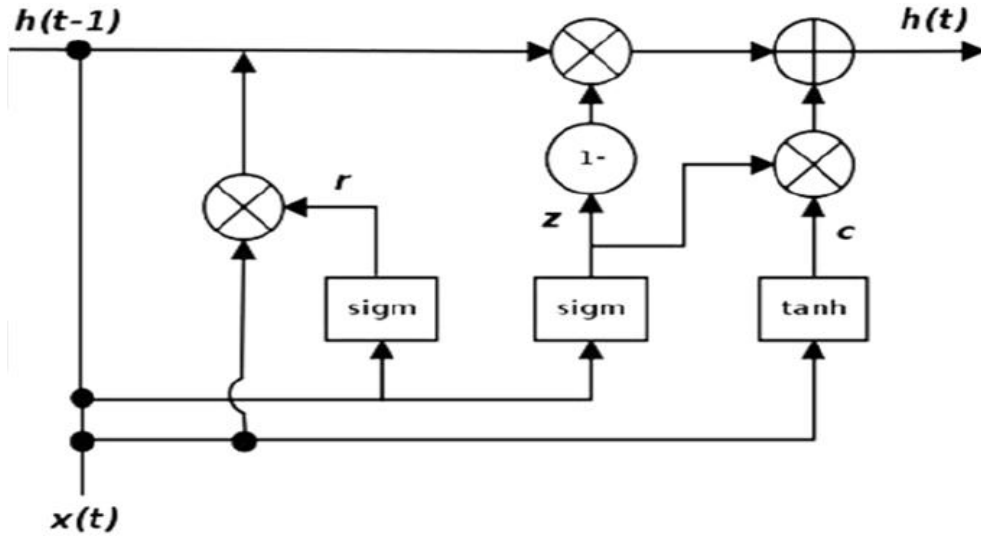


Şekil 2.19. LSTM yapay sinir ağı (Dev, 2017)

Standart bir RNN'de tekrarlayan modül tek bir katman içerir. LSTM de benzer bir yapıya sahiptir ancak tek bir katman yerine dört katmana sahiptir. Yukarıdaki şekilde her satır bir düğümün çıktısından, bir bütün vektörü başkalarının girişlerine taşır. Şekilde daireler vektör eklenmesi gibi noktasal işlemleri, dikdörtgenler ise sinir ağı katmanlarının öğrenmesini temsil eder.

2.4.2. Kapılı yinelenen birim (GRU)

Kapılı Yinelenen Birim (GRU), LSTM'nin geliştirilmiş bir versiyonudur. Her ikisi de benzer şekilde tasarlanır ve bazı durumlarda eşit derecede sonuçlar verir. Ancak GPU'nun içyapısı daha basittir ve bu sayede daha hızlı eğitilir. Şekil 2.20 de GPU yapısı gösterilmiştir (Dev, 2017).



Şekil 2.20. Kapılı yinelenen birim (Dev, 2017)







GPU, RNN'den farklı olarak güncelleme (z) ve sıfırlama (r) kapısı kullanırlar. Temel olarak bu kapılar, çıktıya hangi bilginin aktarılması gerektiğine karar veren iki vektördür. Güncelleme kapısı, önceki hafızanın ne kadar süre daha saklanacağını, sıfırlama kapısı ise yeni girdinin önceki hafıza ile nasıl birleştireceğini tanımlamaktadır.

3. MATERYAL VE YÖNTEMLER

3.1. Veri Seti

Bu tez çalışması için Kütahya Sağlık Bilimleri Üniversitesi Genel Cerrahi Anabilim Dalı Kütahya Evliya Çelebi Eğitim ve Araştırma Hastanesi Endoskopi Ünitesi 2018 yılı arşiv kayıtlarından polipli ve normal görüntüler alınarak veri tabanı oluşturuldu. Oluşturulan veri tabanında 54 kayıttan, 93 polipli ve 216 normal görüntü bulunmaktadır. Bu 54 kayıttan 12 si polipli görüntüye sahip olup, 93 polipli görüntünün 63 tanesi eğitim için 30 tanesi test için rastgele ayrılmıştır. Geriye kalan 42 kayda ait 216 normal görüntünün 144 tanesi eğitim için 71 tanesi de test için ayrılmıştır. Çizelge 3.1’de normal ve polipli görüntüler için örnekler verilmiştir.

Çizelge 3.1. Sağlıklı ve polipli görüntüler

Normal Görüntü	Polipli Görüntü
	
	
	

3.1.1. Veri artırma

Eğitime girecek olan verilere çeşitli bozma ve dönüştürme işlemlerinin uygulanmasıyla eğitimin başarı oranı artırılmaktadır. Bu işlemlerin gerçek zamanlı yapılması ImageDataGenerator sınıfıyla mümkündür. ImageDataGenerator kullanmak

eđitim başarısını olumlu yönde etkilerken eđitim süresini uzatmaktadır. Çizelge 3.2 de bu çalışma için kullanılan veri artırma teknikleri gösterilmiştir.

Çizelge 3.2. Veri Artırma Teknikleri

Veri Artırma Yöntemi	Açıklaması	Deđeri
Rescale	Yeniden ölçeklendirme deđeridir. Varsayılan deđer 'None' olarak ayarlanmıştır. Deđer olarak None veya 0 verilirse yeniden ölçeklendirme uygulanmaz. Aksi takdirde bütün işlemlerden önce girdi verisi, verilen deđerle çarpılır.	1./255
Shear_range	Saat yönünün tersi yönde uygulanan, derece cinsinden shear açısı.	0.2
Zoom_range	Rastgele yakınlaştırma aralığıdır.	0.2
Horizontal_flip	Girdileri rastgele yatayda çevirir	True

3.2. Hiper Parametreler

Çalışmada, veri setinin 2/3'ü eđitim için ve 1/3'ü test verisi olarak ayrılmıştır. Modellerin performansları, test verilerine ait doğruluk (vall_acc) ve hata deđerlerine (vall_loss) göre deđerlendirilmiştir. Çalışmada çizelge 3.3 de verilen nöron sayıları, epoc sayıları, aktivasyon fonksiyonları ve optimizasyon yöntemleri kullanılarak, 48 farklı model oluşturulmuştur.

Çizelge 3.3. Çalışmada oluşturulan modellere ait hiper parametreler

Modelde Kullanılan Parametreler	Parametre Değeri
Nöron Sayıları	32
	64
Epoc Sayıları	5
	10
	15
Aktivasyon Fonksiyonları	Relu
	Tanh
Optimizasyon Yöntemleri	Sgd
	Adagrad
	Adam
	Rmsprop

3.3. Kullanılan Teknolojiler

Verilerin işlenmesi, evrişimli sinir ağının oluşturulup eğitim aşamasının gerçekleştirilmesi ve sonuçların değerlendirilmesi için kullanılan yazılımsal kaynaklara ihtiyaç vardır.

3.3.1. Python

Python işleri hızlı, sisteme entegre ve etkin bir şekilde yazmayı sağlayan bir programlama dilidir. Python ile masaüstü uygulamaları, her ortamda çalışabilen web uygulamaları ve mobil uygulamalar tasarlanabilir. Python tasarlanırken diğer programlama dillerinin olumlu ve olumsuz yönleri dikkate alınmıştır. Bu sayede programdaki teknik hatalar minimum seviyeye indirgenmiştir. Python'un genel özelliklerini şöyle sıralanabilir.

- Nesne tabanlı ve yüksek seviyeli bir dildir.
- Yorumsaldır, derlenmeye ihtiyaç duyulmaz. Bu sayede program geliştirmek daha kolaydır.
- Açık kaynaklıdır.
- Windows, Linux, Mac, Unix gibi farklı platformlarda çalışabilir.

- Basit bir söz dizinine sahiptir. Bu sayede hem daha kolay yazılır ve öğrenilir hem de daha okunaklıdır.
- Program yazarken ihtiyaç duyulan veri yapıları ve fonksiyonları hazır olarak sunar.

Bilim ve teknoloji sanayisinde, uluslararası alanda, askeri savunmada ve birimsel hesaplamalarda kullanılabilir.

3.3.2. Pycharm

Jetbrains firması tarafından geliştirilen, birden fazla işletim sisteminde çalışabilen, bir tümleşik geliştirme ortamıdır. Projeleri uzaktaki bir makineye veya aynı makinede kurulu başka bir sanal makineye dağıtılmasına olanak sağlar. Pycharm ile programın ilerleyişini ve tanımlanan değişkenleri takip etmek kolaydır.

Auto-fix özelliği sayesinde otomatik olarak hataları düzeltebilir. Console sayesinde kendi üzerinde terminal açar. Versiyon kontrol sistemi özelliğiyle, var olan proje sistemlerle uyum sağlayarak versiyonlama işlemi yapabilmektedir.

3.3.3. Numpy ve Scipy

Python'un günümüzde bu kadar aktif kullanılmasının nedenlerinden biri de hesaplamalı birimlerdeki işlevselliğidir. Bu özelliği sayesinde, C ve C++ gibi diğer programlama dillerine oranla karmaşık verileri daha iyi bir performansla gerçekleştirir. Python bu özelliğini sahip olduğu kütüphanelere borçludur. Numpy ve Scipy bu kütüphanelerdendir.

Numpy, çok boyutlu dizilerde gerçekleştirilen matematiksel işlemleri daha hızlı yapmamızı sağlayan Python kütüphanesidir. Bu kütüphane kullanılarak matris oluşturma, toplama ve çarpma gibi işlemler daha kolay yapılabilmektedir. Numpy dizileri, sabit boyutlu oluşturulur. Dizi boyutu değiştirildiğinde yeni bir dizi oluşturulur ve var olan dizi silinir. Ayrıca dizi elemanlarının aynı veri türünde olması gerekir.

Scipy ise, optimizasyon, sayısal integral, diferansiyel denklemler, sinyal işleme, Fourier analizi, lineer cebir gibi sayısal işlemler için hazır fonksiyonlar sağlayan, Numpy uzantısına dayanan bir kütüphanedir.

3.3.4. Tensorflow

Tensorflow, veri akış grafikleri kullanarak sayısal hesaplama için Google'ın açık kaynak kodlu sunduğu makine öğrenmesi kütüphanesidir. Tensorflow, oluşturulan yapay sinir ağı modelini birçok sunucu üzerinde çalıştırma imkânı sağlar.

TensorFlow iki terimden oluşur. Bunlar tensör ve flow (akış terimi). Tensör terimi, n boyutlu dizi olarak tanımlanırken, akış terimi tensor üzerinde gerçekleştirilen işlem serileri olarak tanımlanır. Tensorflow, tensördeki işlevleri tanımlamak ve türevlerini otomatik olarak hesaplamak için ilkeler sağlar.

3.3.5. Keras

Keras, Theano ya da Tensor kütüphaneleri üzerinde çalışan, derin öğrenme için yazılmış bir Python kütüphanesidir.

Theano ve Tensorflow kütüphanelerine göre daha kolay uygulama geliştirilebilir ama CPU ya da GPU üzerinde çalışmasını da bu temel kütüphaneler üzerinden sağlar. Keras kullanıcılara kendi aktivasyon, kayıp ve öğrenme algoritmalarını geliştirip çalışmalarına olanak sağlayan kütüphane desteği sunmaktadır.

3.4. Değerlendirme Kriterleri

Sınıflanlandırma algoritmaları kullanan modellere sınıflandırıcı denir. Sınıflandırıcı başarımı birçok farklı yöntemle ölçülebilir. Sınıflandırıcı birçok örnekleme doğru değerlendirirken, bazı örneklemleri doğruyken yanlış, bazı örneklemleri yanlışken doğru belirleyebilir. Bu durumlar göz önüne alınarak, karmaşıklık matrisi ve sınıflandırma doğruluk oranları oluşturulur.

3.4.1. Karmaşıklık matrisi

Karmaşıklık matrisi, farklı tahminleri ve test sonuçlarını özetleyen ve bunları gerçek dünya değerleri ile karşılaştıran bir tablodur. Karmaşıklık matrisleri, istatistiksel verilerin analizini daha hızlı yapmak ve sonuçların veri görselleştirilmesi yoluyla okunmasını kolaylaştırmak için kullanılır.

Karmaşıklık matrisi, sınıflandırma modelinin ne kadar doğru olduğu ve ne tür hatalar yapıldığı konusunda daha iyi fikirler verebilir. Karmaşıklık matrisi istatistiklerde, veri madenciliğinde, makine öğrenmesi modellerinde ve diğer yapay zekâ uygulamalarında kullanılmaktadır.

Bir modelde sınıflandırma doğruluğundaki temel sorun, sınıflandırma performansını daha iyi anlamak için duyulan ayrıntının gizlenmesidir ve bu durumda karmaşıklık matrisi kullanılarak gizlenen ayrıntılar ayrılabilir. Sınıflandırma doğruluğunda ki bu sorun iki durumda meydana gelebilir (Brownlee, 2016). Birincisi sınıf sayısı iki veya daha fazla veri setine sahip modellerde %80 lik bir sınıflandırma doğruluğu elde edilebilir. Bunun nedeni, tüm sınıfların eşit şekilde tahmin edilip edilmediğini ya da bir veya iki sınıfın model tarafından ihmal edilip edilmediğinin bilinmemesindedir. İkincisi ise veriler çift sayıda sınıfa sahip olduğunda %90 veya daha fazla doğruluk elde edilebilir. Ancak 100 veri için 90 veri bir sınıfa aitse ve skoru her zaman en yaygın sınıf değerini tahmin ederek elde edilirse bu iyi bir sonuç olmaz.

Karmaşıklık matrisini hesaplamak için uygulanan işlemler şöyledir:

1. Bir test veri setine veya beklenen sonuç değerine sahip bir doğrulama veri setine sahip olmak.
2. Test veri kümesindeki her satır için tahmin yapılır.
3. Beklenen sonuç ve tahminler sayılır.
 1. Her satır için doğru tahmin sayısı
 2. Tahmin edilen sınıf tarafından düzenlenen her sınıf için yanlış tahmin sayısı

Bir sınıf için toplam doğru tahmin sayısı, o sınıf için beklenen satıra ve o sınıf için öngörülen sütuna gider. Aynı şekilde bir sınıf için toplam yanlış tahmin sayısı o sınıf değeri

için beklenen satıra ve o sınıf değeri için öngörülen sütuna gider. Bu sayılar daha sonra çizelge 3.4 gibi bir tablo ya da matris halinde düzenlenir.

Çizelge 3.4. Karmaşıklık matrisi

		TAHMİNİ DEĞER	
		POZİTİF	NEGATİF
GERÇEK DEĞER	POZİTİF	TP	FN
	NEGATİF	FP	TN

Çizelge 3.4. de gösterilen karmaşıklık matrisinde eğitim veri kümesinde belirlenmiş gerçek durumlar ile modelinin tahmin ettiği durumlar gösterilmiştir. Burada 4 ayrı durum vardır:

- Gerçek Pozitif (True Positive - TP): Doğru tahmin edilen pozitif değerlerdir, yani gerçek sınıfın değeri pozitif, öngörülen sınıfın değeri de pozitiftir.
- Gerçek Negatif (True Negative - TN): Doğru tahmin edilen negatif değerlerdir, yani gerçek sınıfın değeri negatif ve öngörülen sınıfın değeri de negatiftir.
- Yanlış Pozitif (False Positive - FP): Gerçek değer negatif, öngörülen değer pozitif olduğu durumlardır.
- Yanlış Negatif (False Negative - FN): Gerçek değer pozitif, öngörülen değer negatif olduğu durumlardır.

3.4.2. Sınıflandırma doğruluk oranları

Doğruluk (Accuracy) : Doğru tahminli gözlemin, toplam gözlemlere oranıdır.

Sınıflayıcının ne sıklıkta doğru tahmin ettiğinin bir ölçüsüdür. Yanlış pozitif ve yanlış negatif değerlerin neredeyse aynı olduğu simetrik veri kümeleri olduğunda doğruluk

en iyi sonucu verir. Doğruluk değeri ne kadar yüksek çıkarsa, model o kadar iyi çalışıyor demektir.

$$\text{Doğruluk} = \frac{TP+TN}{TP+TN+FP+FN} \quad (3.1)$$

Kesinlik (Precision) : Kesinlik, doğru tahmin edilen pozitif gözlemlerin toplam öngörülen pozitif gözlemlere oranıdır. Örneğin e-posta spam tespitinde, yanlış pozitif değeri spam olmayan (gerçek negatif) bir e-postanın önceden belirlenmiş spam olarak tanımlandığı anlamına gelir. Spam algılama modeli için kesinlik değeri yüksek değilse e-posta kullanıcısı önemli e-postaları kaybedebilir (Shung, 2018).

$$\text{Kesinlik} = \frac{TP}{TP+FP} \quad (3.2)$$

Hassasiyet (Recall) : Tüm pozitif sınıflardan ne kadarının doğru tahmin edildiğidir. Doğru tahmin edilmiş pozitif gözlemlerin gerçek sınıftaki tüm pozitif gözlemlere oranıdır.

$$\text{Hassasiyet} = \frac{TP}{TP+FN} \quad (3.3)$$

F1 Skoru : F1 skoru kesinlik ve hassasiyet değerlerinin ağırlıklı ortalamasıdır. Bu nedenle bu puan hem yanlış pozitifleri hem de yanlış negatifleri hesaba katar. Sezgisel olarak doğruluk kadar anlaşılması kolay değildir ancak F1 özellikle düzensiz bir sınıf dağılımı varsa genellikle doğruluktan daha kullanışlıdır (Joshi, 2016).

$$\text{F1 Skoru} = \frac{2 \times \text{kesinlik} \times \text{hassasiyet}}{\text{kesinlik} + \text{hassasiyet}} \quad (3.4)$$

4. BULGULAR VE TARTIŞMA

Bölüm 3.2 de verilen hiper parametreler ile hazırlanan ağ modelleri; 5,10 ve 15 epoch sayıları ve her epoch içerisinde 50 iterasyon olacak şekilde örnek veri seti ile eğitim sürecine uygulanmıştır. Modellerin doğruluk (accuracy), kayıp (loss), test doğruluk (vall_acc) ve test hata (vall_loss) değerleri çizelge 4.1’de gösterilmiştir.

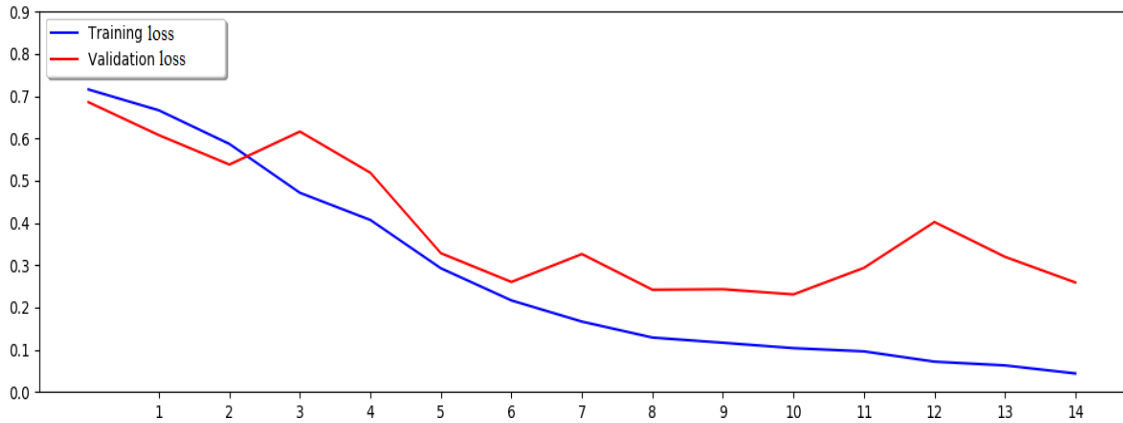
Çizelge 4.1 Çalışmada kullanılan modellere ait sonuçlar

Model No	Nöron Sayısı	Aktivasyon Fonksiyonu	Epoc Sayısı	Optimizasyon Fonksiyonu	Doğruluk	Kayıp	Test Hata	Test Doğruluk
1	32	Relu	5	Sgd	0.71	0.59	0.58	0.69
2				Adagrad	0.92	0.21	0.27	0.90
3				Adam	0.92	0.22	0.40	0.87
4				RmsProp	0.92	0.24	0.33	0.89
5			10	Sgd	0.83	0.41	0.48	0.76
6				Adagrad	0.95	0.13	0.40	0.89
7				Adam	0.94	0.15	0.35	0.88
8				RmsProp	0.95	0.13	0.38	0.90
9			15	Sgd	0.85	0.36	0.36	0.87
10				Adagrad	0.97	0.07	0.44	0.98
11				Adam	0.94	0.14	0.43	0.89
12				RmsProp	0.97	0.07	0.58	0.89
13	64	Relu	5	Sgd	0.76	0.54	0.53	0.77
14				Adagrad	0.92	0.20	0.27	0.89
15				Adam	0.93	0.19	0.27	0.88
16				RmsProp	0.93	0.19	0.37	0.89
17			10	Sgd	0.82	0.42	0.38	0.85
18				Adagrad	0.94	0.13	0.31	0.90

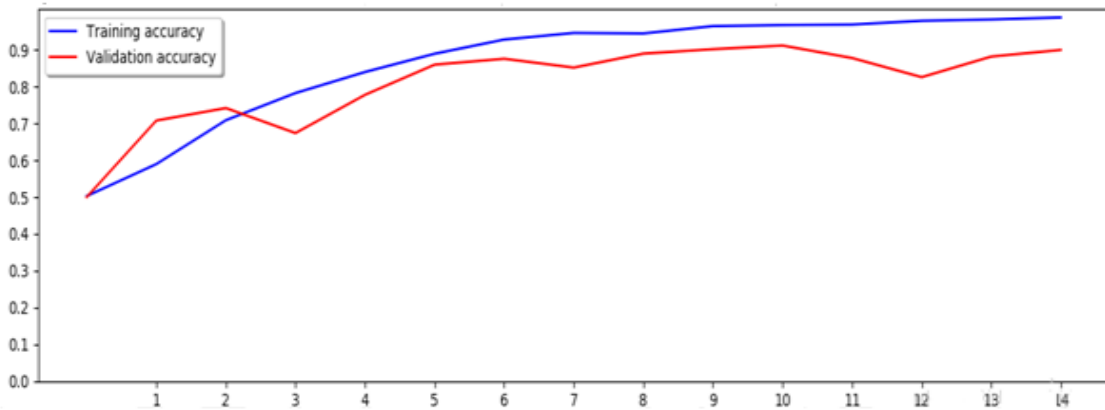
19	64	Relu	10	Adam	0.96	0.09	0.40	0.87		
20				RmsProp	0.95	0.14	0.36	0.90		
21			15	Sgd	0.88	0.33	0.39	0.82		
22				Adagrad	0.98	0.02	0.75	0.89		
23				Adam	0.98	0.04	0.57	0.87		
24				RmsProp	0.98	0.04	1.02	0.88		
25	32	Tanh	5	Sgd	0.82	0.44	0.42	0.85		
26				Adagrad	0.68	0.62	0.61	0.70		
27				Adam	0.93	0.18	0.26	0.90		
28				RmsProp	0.92	0.21	0.32	0.88		
29			10	Sgd	0.96	0.09	0.43	0.88		
30				Adagrad	0.69	0.61	0.60	0.70		
31				Adam	0.97	0.08	0.45	0.89		
32				RmsProp	0.96	0.10	0.29	0.92		
33			15	Sgd	0.89	0.30	0.36	0.87		
34				Adagrad	0.98	0.04	0.36	0.88		
35				Adam	0.98	0.05	0.59	0.85		
36				RmsProp	0.97	0.07	0.43	0.89		
37			64	Tanh	5	Sgd	0.82	0.42	0.60	0.74
38						Adagrad	0.69	0.63	0.60	0.70
39	Adam	0.94				0.16	0.35	0.89		
40	RmsProp	0.93				0.19	0.31	0.88		
41	10	Sgd			0.87	0.33	0.35	0.86		
42		Adagrad			0.92	0.23	0.31	0.89		
43		Adam			0.97	0.06	0.49	0.89		
44		RmsProp			0.97	0.07	0.44	0.88		
45	15	Sgd			0.89	0.30	0.34	0.87		
46		Adagrad			0.99	0.04	0.37	0.89		
47		Adam			0.97	0.04	0.60	0.89		
48		RmsProp			0.98	0.04	0.72	0.89		

Çizelge 4.1'e göre 10'nolu model; gizli katmanında 32 nöron bulunan, Relu aktivasyon fonksiyonlu, 15 epoch sayılı ve Adagrad optimizasyon tekniği kullanılarak %98 doğruluk oranıyla en başarılı sınıflandırma yapan model olmuştur. 1'nolu model %69 doğruluk oranı ile 48 model arasından en düşük başarı oranıyla sınıflandırma yapmıştır. Test verilerinin doğruluk değerlerine bakıldığında, modellerin sınıflandırma aralığı (0.69-0.98) arasında değişmektedir. Çizelge 5.1 incelendiğinde Adagrad, Adam ve Rmsprop optimizasyon yöntemleri ile en başarılı modeller oluşturulurken, en düşük başarıya sahip model Sgd optimizasyon yönteminden elde edilmiştir.

Şekil 4.1'de 10'nolu modele ait hata grafiği, Şekil 4.2' de ise doğruluk grafiği verilmiştir. Bu şekiller incelendiğinde 10'nolu modelin 10'uncu epokta en yüksek performansa ulaştığı ve eğitimin ortalama olarak bu şekilde devam ettiği görülmektedir.



Şekil 4.1. 10'nolu modele ait hata grafiği



Şekil 4.2. 10'nolu modele ait doğruluk grafiği

Çizelge 4.2. En başarılı beş modele ait sonuçlar

Model no	Nöron Sayısı	Aktivasyon Fonksiyonu	Epoch Sayısı	Optimizasyon Yöntemi	Test Hata	Test Doğruluk
10	32	Relu	15	Adagrad	0.44	0.98
32	32	Tanh	10	Rmsprop	0.29	0.92
27	32	Tanh	5	Adam	0.26	0.90
2	32	Relu	5	Adagrad	0.27	0.90
18	64	Relu	10	Adagrad	0.31	0.90

Çizelge 4.3. En düşük başarıya sahip beş modellere ait sonuçlar

Model no	Nöron Sayısı	Aktivasyon Fonksiyonu	Epoch Sayısı	Optimizasyon Yöntemi	Test Hata	Test Doğruluk
37	64	Tanh	5	Sgd	0.60	0.74
30	32	Tanh	10	Adagrad	0.60	0.70
26	32	Tanh	5	Adagrad	0.61	0.70
38	64	Tanh	5	Adagrad	0.60	0.70
1	32	Relu	5	Sgd	0.58	0.69

Çizelge 4.1, 4.2 ve 4.3 incelendiğinde Adagrad optimizasyon tekniğinin Relu aktivasyon fonksiyonuyla, Sgd optimizasyon tekniğinin ise Tanh aktivasyon fonksiyonuyla kullanıldığında daha başarılı sonuçlar elde edilmiştir. Adagrad optimizasyon algoritmasının Relu aktivasyon fonksiyonu ile kullanıldığında 32 nöronlu algoritmanın, 64 nöronlu algoritmaya göre daha başarılı olduğu belirlenmiştir ($0.98 > 0.89$; model 10, model 22). Ayrıca Adam optimizasyon algoritmasının Relu aktivasyon fonksiyonuyla kullanıldığında

32 nöronlu modelin daha başarılı olduğu ($0.89 > 0.87$; model 11, model 23), Tanh aktivasyon fonksiyonu ile kullanıldığında ise 64 nöronlu modelin daha başarılı olduğu belirlenmiştir ($0.89 > 0.85$; model 35, model 47).

Bu çalışmada endoskopi görüntüleri incelenmiş, normal ve polipli görüntülerin sınıflandırılması için farklı aktivasyon ve optimizasyon yöntemleriyle oluşturulan modellerin karşılaştırılması yapılmıştır. Elde edilen verilere göre modellerin sonuçlarının parametrelere göre farklılık gösterdiği gözlenmiştir.

Son yıllarda mühendislik alanında popüler olan derin öğrenme algoritmalarından biri olan CNN yöntemi bu çalışmada kullanılarak hekimlere destek sisteminin tasarımını ortaya koymuştur. Farklı kanserlerin teşhisinde ve biyometrik tanıma üzerine yapılan çalışmalarda da derin öğrenme yöntemleri kullanılmıştır. Ancak daha önceki çalışmalarda endoskopi görüntüleri kullanılarak aktivasyon fonksiyonu, optimizasyon yöntemi, nöron ve epoc sayılarında değişiklik yapıp bunların karşılaştırılması yapılmamıştır. Elde ettiğimiz sonuçlar, tıbbi karar vermeyi destekler yönündedir.

Shin ve Balasingham (2017) çalışmalarında, SVM ve CNN sınıflandırma yöntemlerini karşılaştırmak için 978 görüntü kullanmışlardır. Bu çalışma için üç tane konvolüsyon katmanı ve adamax optimizasyon fonksiyonu kullanılmıştır. Çalışma sonunda SVM ile %84, CNN ile %91 doğruluk elde edilmiştir. Bu çalışmada ise 309 görüntü, iki tane konvolüsyon katmanı ve en iyi sonucu elde ettiğimiz modelde adagrad optimizasyon fonksiyonu kullanılmıştır. Çalışma sonunda %98 doğruluk elde edilmiştir. Çalışmamızda kullanılan veri kümesi ve konvolüsyon katman sayısı daha küçük olmasına rağmen daha yüksek başarı elde edilmesi farklı optimizasyon fonksiyonu kullanılmasından dolayı olabilir.

Zou vd. (2015) çalışmalarında, DCNN modelini kullanarak kapsül endoskopisi görüntülerindeki sindirim organlarının sınıflandırma problemini incelemişlerdir ve bunun için bir milyon görüntü kullanmışlardır. Öğrenme oranı 0.01 olarak belirlenmiş ve tam bağlı katmanların her biri 64 nörona sahiptir. CNN ile %95.52 doğruluk elde edilmiştir. Çalışmamızda ise öğrenme oranı 0.5 olarak belirlenmiş ve en iyi sonucu elde ettiğimiz modelde tam bağlı katmanların her biri 32 nörona sahiptir. Çalışmamızda daha küçük veri

kümesi ve nöron sayısının daha küçük olmasına rağmen daha büyük başarı elde edilmesi öğrenme oranı arasındaki farktan kaynaklı olabilir.

5. SONUÇ VE ÖNERİLER

Derin öğrenmedeki en iyi sınıflandırma modelini bulmak için farklı aktivasyon ve optimizasyon yöntemi, farklı nöron ve epoc sayıları kullanılmıştır. Modellerin sonuçları parametrelere göre farklılık göstermiştir.

Optimizasyon yöntemleri, aktivasyon fonksiyonları, nöron ve epoc sayıları kullanılarak 48 farklı derin sinir ağı modeli oluşturulmuştur. Adagrad, Adam ve Rmsprop optimizasyon yöntemleri ile en başarılı modeller oluşturulurken, başarı oranı en düşük yöntemin Sgd olduğu belirlenmiştir. Çalışma sonuçları incelendiğinde genel olarak Relu aktivasyon fonksiyonunu Tanh'a aktivasyon fonksiyonuna göre daha başarılı sonuçlar ortaya koyarken, Sgd optimizasyon yönteminin Tanh aktivasyon fonksiyonu ile daha başarılı olduğu görülmektedir.

Derin sinir ağları ile yapılan çalışmalarda; kullanılan veri setinin yapısına, boyutuna, optimizasyon yöntemlerine ve aktivasyon fonksiyonlarına göre modellerin performanslarının değişiklik gösterdiği bilinmektedir. Ayrıca optimizasyon algoritmalarının performansı, parametlerin seçimine ve sinir ağının nasıl oluşturulacağına bağlı olarak farklılık göstermektedir.

Derin öğrenme uygulamalarında optimizasyon algoritmalarının seçim ölçütleri tam olarak bilinmemekle beraber, problemin yapısına ve parametlere bağlı olarak algoritmalar farklı performanslar göstermektedir. Dolayısıyla, derin öğrenme uygulamalarında model oluşturulurken optimizasyon algoritmaları, aktivasyon fonksiyonları, nöron ve epoc sayılarının farklı kombinasyonlarının oluşturulması durumunda, veri setine daha uygun mimari elde edilecektir.

Mevcut çalışmanın değerlendirilmesinde temel olarak sınıflandırma doğruluk oranı kullanılmaktadır. Gelecekteki çalışmalarda sınıflandırma hızı ve maliyeti gibi kriterlere de odaklanılacaktır. Ayrıca aynı parametre ve değerleri kullanılarak derin öğrenme algoritmaları (CNN, LSTM, GRU, RNN) karşılaştırılarak analizi yapılacaktır.

KAYNAKLAR DİZİNİ

- Abdel-Zaher, A., Eldeib, A., 2016, Breast cancer classification using deep belief networks, Expert Systems with Applications, p. 139–144
- Anonim, 2017, Olasılıksal dereceli azalma, <https://veribilimcisi.com/2017/07/19/olasiliksal-dereceli-azalma-stochastgra-di-ent-descent-sgd/> , erişim tarihi: 13.02.2019
- Anonim, 2017, Tekrarlayan sinir ağları, <https://veribilimcisi.com/2017/09/26/tekrarlayan-sinir-aglari-recurrent-neural-network/>, erişim tarihi: 11.03.2019
- Anonim, 2018, Endoskopi, <http://endohem.com/endoskopi>, erişim tarihi: 03.01.2019
- Anonim, 2018, Max-pooling, https://computersciencewiki.org/index.php/Max-pooling/_/Pooling, erişim tarihi: 01.03.2019
- Anwer, A., 2017, Yüksek Lisans tezi, Derin öğrenme yöntemleri ile göğüs kanseri teşhisi, Türk Hava Kurumu Üniversitesi, 104 s.
- Arı, A., Berberler, M. E., 2017, Yapay sinir ağları ile tahmin ve problemlerinin çözümü için arayüz tasarımı, Acta Infologica, 1,2, s.55-73
- Bengio, Y., 2008, Learning deep architectures for AI, Foundations and Trends in Machine Learning, 2, 1, p. 1– 127.
- Bengio, Y., Le Cun, Y., 2007, Scaling learning algorithms towards AI, Large Scale Kernel Machines, 34, 5 , 1-41
- Brownlee, J., 2016, Machine learning project in Python, <https://machinelearningmastery.com/machine-learning-in-python-step-by-step/>, erişim tarihi: 25.01.2019
- Brownlee, J., 2016, What is a confusion matrix in machine learning, <https://machinelearningmastery.com/confusion-matrix-machine-learning/>, erişim tarihi: 15.03.2019

KAYNAKLAR DİZİNİ (devam)

- Budhiraja, A., 2016, Dropout in (deep) machine learning, <https://medium.com/@amarbudhiraja/https-medium-com-amarbudhiraja-learning-less-to-learn-better-dropout-in-deep-machine-learning-74334da4bfc5>, erişim tarihi: 05.03.2019
- Castelluccio, M., Poggi, G., Sansone, C., Verdoliva, L., 2015, Land use classification in remote sensing images by convolutional neural networks, arXiv preprint arXiv:1508.00092.
- Çarkacı, N., 2018, Derin öğrenme uygulamalarında en sık kullanılan hiper parametreler, <https://medium.com/deep-learning-turkiye/derin-ogrenme-uygulamalarinda-en-sik-kullanilan-hiper-parametreler-ece8e9125c4>, erişim tarihi: 02.02.2019
- Çayıroğlu, İ., 2019, İleri algoritma analizi, <http://www.ibrahimcayiroglu.com/dokumanlar/ilerialgoritmaanalizi/ilerialgoritmaanalizi-5.hafta-yapaysiniraglari.pdf>, erişim tarihi: 23.01.2019
- Çınar, A. C., 2017, Matlab ile derin öğrenmeye giriş, <http://ahmetcevahircinar.com.tr/2017/08/09/matlab-ile-derin-ogrenmeye-giris-introducing-deep-learning-with-matlab/>, erişim tarihi: 17.02.2017
- Çoşar, K., 2018, Derin öğrenme, <http://kubracosar.blogspot.com/2018/01/deep-learning-derin-ogrenme.html>, erişim tarihi: 26.02.2019
- Dev, D., 2017, Deep learning with hadoop, e-kitap, 71-150, https://books.google.com.tr/books?hl=tr&lr=&id=K1QoDwAAQBAJ&oi=fnd&pg=PP1&dq=+D.+Dev,+Deep+Learning+with+Hadoop,+Packt+Publishing,+2017.&ots=UHC2YWOsF5&sig=fjcZhDqEsuNwMvAyyX8nQDxT98M&redir_esc=y#v=onepage&q=amazon&f=false, erişim tarihi: 20.02.2019
- Dhungel, N., Carneiro, G., Bradley. A., 2015, Automated mass detection in mammograms using cascaded deep learning and random forests, International Conference on Digital Image Computing: Techniques and Applications, p. 1-8
- Doğan, F., Türkoğlu, İ., 2018, Derin öğrenme algoritmalarının yaprak sınıflandırma başarımlarının karşılaştırılması, Sakarya University Journal of Computer and Information Sciences, 1, 1, s.10-21
- Duchi, J., Hazan, E., Singer, Y., 2011, "Adaptive subgradient methods for online learning and stochastic optimization, Journal of Machine Learning Research, p. 2121-2159

KAYNAKLAR DİZİNİ (devam)

- İnik, Ö., Ülker, E., 2017, Derin öğrenme ve görüntü analizinde kullanılan derin öğrenme modelleri, Gaziosmanpaşa Bilimsel Araştırma Dergisi, 6, 3, 85-104
- İstanbulu, H., 2018, Endoskopi nedir? Neden ve Nasıl yapılır? Riskleri nelerdir?, <https://www.medikalakademi.com.tr/endoskopi-nedir-neden-ve-nasil-yapilir-riski-nedir/>, erişim tarihi: 05.01.2019
- Kaçmaz, R., Yılmaz, B., 2018, Detection of ulcerative colitis from colonoscopy images, Medical Technologies National Congress, IEEE, p. 1-4
- Kızrak, A., 2019, Derin öğrenme için aktivasyon fonksiyonlarının karşılaştırılması, <https://mc.ai/derin-ogrenme-icin-aktivasyon-fonksiyonlarinin-karsilastirilmesi/>, erişim tarihi: 03.03.2019
- Kingma, D. P., Ba, J., 2014, Adam a method for stochastic optimization, arXiv preprint arXiv:1412.6980
- Krizhevsky, A., Sutskever, I., Hinton, G. E., 2012, Imagenet classification with deep convolutional neural networks, In Advances in neural information processing systems p. 1097-1105
- Kutlu, H., 2018, Biyoistatistik temelli bilimsel araştırmalarda derin öğrenme uygulamaları, KKTC, 66 s.
- LeCun, Y., Kavukcuoglu K., Farabet C., 2010, Convolutional networks and applications in vision, Proceedings of 2010 IEEE International Symposium on Circuits and Systems., p. 253–256
- Liu, L., Shen, C., Van den Hengel, A., 2015 The treasure beneath convolutional layers: Cross-convolutional-layer pooling for image classification, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition p. 4749-4757
- Millington, I., Funge, J., 2009, Artificial intelligence for games, CRC Press, p. 647
- Pratap, KS., 2013, Artificial Neural Network (Ann) Inspired From Biological Nervous System. International Journal of Application or Innovation in Engineering Management, 2(1), 227-231.

KAYNAKLAR DİZİNİ (devam)

- Ribeiro, E., Uhl, A., Hafner, M., 2016, Colonic polyp classification with convolutional neural networks, 2016 IEEE 29th International Symposium on Computer-Based Medical Systems, p.253-258
- Ruder, S., 2016, An overview of gradient descent optimization algorithms, arXiv preprint arXiv:1609.04747
- Sarraf, S., Tofighi, G., 2016, Deep learning-based pipeline to recognize Alzheimer's disease using fMRI data, Future Technologies Conference, IEEE, p. 816-820
- Schmidhuber, J., 2015, Deep learning in neural networks: An overview, Neural Networks, 61, 85–117.
- Shen, D., Guoron, W., Heung-II, S., 2017, Deep learning in medical image analysis, Annual review of biomedical engineering 19, p 221-248.
- Shin, Y., Balasingham I., 2017, Comparison of hand-craft feature based SVM and CNN based deep learning framework for automatic polyp classification, 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 3277-3280
- Shung, K. P., 2018, Accuracy, precision, recall or F1, <https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9>, erişim tarihi: 19.03.2019
- Sivanandam, S.N., Deepa, S. N., 2006, Introduction to neural networks using Matlab 6.0, Hill Education, p.53
- Soytürk M, Akkaya S, Sarioğlu S, Sağol Ö, Derviş G, vd., 2012, Endoskopik incelemede saptanan mide poliplerinin yerleşimi, histopatolojik tipleri ve helicobacter pylori enfeksiyonu ile ilişkisi, Endoskopi Gastrointestinal, 20, 1, 1-4, <http://dergipark.gov.tr/download/article-file/63960>, erişim tarihi: 17.02.2019
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever I., Salakhutdinov R., 2014, Dropout: a simple way to prevent neural networks from overfitting, Journal of machine learning research, 15, 1, 1929- 1958.

KAYNAKLAR DİZİNİ (devam)

Suzuki, S., Zhang, X., Homma, N., Ichiji, K., Sugita, N., vd, 2016, Mass detection using deep convolutional neural network for mammographic computer-aided diagnosis, 55th Annual Conference of the Society of Instrument and Control Engineers of Japan, IEEE, p. 1382-1386

Şengür, A., Türkoğlu, İ., İnce, İ., 2010, Endoskopik görüntülerin değerlendirilmesinde görüntü işleme temelli akıllı bir karar destek sistemi, Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi, 15, 1, 33-42

Taviloğlu, K., 2017, Kapsül endoskopisi, <http://www.taviloglu.com/endoskopi/kapsul-endoskopisi.html>, erişim tarihi: 08.01.2019

Yang, X., Chen, S., Ding, V., Pang, B., 2016, A deep learning approach for tumor tissue image classification, Biomedical Engineering

Uysal, F., Hardalaç, F., Koç, M., 2018, Classification of T1 and T2 weighted magnetic resonance prostate images using convolutional neural networks, 2018 Medical Technologies National Congress, IEEE, p. 1-4

Yazan, E., Talu, M. F., 2017, Comparison of the stochastic gradient descent based optimization techniques, International Artificial Intelligence and Data Processing Symposium, IEEE, p. 1-5

Yixuan, Y., Meng, M., 2017, Deep learning for polyp recognition in wireless endoscopy images, Medical Physics, 44, 4, 1379-1389

Zou, Y., Li, L., Wang, Y., Yu, J., Li, Y., vd., 2015, Classifying digestive organs in wireless capsule endoscopy images based on deep convolutional neural network, IEEE International Conference on Digital Signal Processing, p. 1274-1278

EK AÇIKLAMALAR**Sayfa**

Ek Açıklama-A: Kütahya Sağlık Bilimleri Üniversitesi Girişimsel Olmayan Klinik Araştırmalar Etik Kurulu Raporu	54
---	----

EK AÇIKLAMALAR

Ek Açıklama-A: Kütahya Sağlık Bilimleri Üniversitesi Girişimsel Olmayan Klinik Araştırmalar Etik Kurulu Raporu

Evrak Tarih ve Sayısı: 17/07/2019-E.5269



T. C.
KÜTAHYA SAĞLIK BİLİMLERİ ÜNİVERSİTESİ
REKTÖRLÜĞÜ
Girişimsel Olmayan Klinik Araştırmalar Etik Kurulu
Başkanlığı

BD6513800922

Sayı : 41997688-402.03.01-
Konu : Başvuru Sonucu

Sayın Doç. Dr. Faik YAYLAK

Girişimsel Olmayan Klinik Araştırmalar Etik Kurulu tarafından gerçekleştirilen 27.06.2019 tarih ve 2019/07 sayılı toplantıda "Endoskopi raporlarının metin madenciliği algoritmaları kullanılarak incelenmesi" ve "Endoskopi görüntülerindeki poliplerin derin öğrenme algoritmaları kullanılarak incelenmesi" konulu çalışmaların araştırılması incelenmiş olup karar formu ekte tarafınıza sunulmaktadır. Gereğini bilgilerinize rica ederim.

e-imza

Prof. Dr. Duygu PERÇİN RENDERS
Başkan

EK :
Başvuru Sonucu (2 Sayfa)

Evliya Çelebi Yerleşkesi Tavşanlı Yolu 10. km KÜTAHYA

Telefon: 0 (274) 260 00 43 / 1139-Faks:
E-Posta: etik.gir.olmayan@ksbu.edu.tr

Ayrıntılı bilgi için irtibat: E.YAŞİ Veri Hazırlama ve Kontrol
İşletmeni
Elektronik ağ:

Bu belge 5070 sayılı Elektronik İmza Kanununun 5. Maddesi gereğince güvenli elektronik imza ile imzalanmıştır.

T.C
KÜTAHYA SAĞLIK BİLİMLERİ ÜNİVERSİTESİ REKTÖRLÜĞÜ
GİRİŞİMSEL OLMAYAN KLİNİK ARAŞTIRMALAR ETİK KURULU
KARAR FORMU

ARAŞTIRMANIN AÇIK ADI	“Endoskopi görüntülerindeki poliplerin derin öğrenme algoritmaları kullanılarak incelenmesi” adlı araştırma
-----------------------	---

ETİK KURUL BİLGİLERİ	ETİK	Kütahya Sağlık Bilimleri Üniversitesi
	KURULUN ADI	Girişimsel Olmayan Klinik Araştırmalar Etik Kurulu
	KURUL ADRESİ	Kütahya Sağlık Bilimleri Üniversitesi Evliya Çelebi Yerleşkesi Tavşanlı Yolu 10. Km KÜTAHYA
	TELEFON	(0 274) 260 00 43 / 1199
	FAKS	(0 274) 265 22 85
	E-POSTA	etik.gir.olmayan@ksbu.edu.tr

BAŞVURU BİLGİLERİ	KOORDİNATÖR/SORUMLU ARAŞTIRMACI UNVANI/ADI/SOYADI	Doç. Dr. Faik YAYLAK
	KOORDİNATÖR/SORUMLU ARAŞTIRMACININ UZMANLIK ALANI	Kütahya Sağlık Bilimleri Üniversitesi Genel Cerrahi Anabilim Dalı
	KOORDİNATÖR/SORUMLU ARAŞTIRMACININ BULUNDUĞU MERKEZ	KSBUE.A.H Tıp Fakültesi (Genel Cerrahi)
	YARDIMCI ARAŞTIRMACI VE BÖLÜMÜ	Prof. Dr. M. Cem ALGIN / Doç. Dr. Sezgin EREN / Emine CENGİZ / Eyyüp GÜLBANDILAR

KARAR BİLGİLERİ	Karar No : 2019/07-8	Tarih: 27.06.2019
	<p>“Endoskopi görüntülerindeki poliplerin derin öğrenme algoritmaları kullanılarak incelenmesi” adlı araştırma konulu dosyanız Girişimsel Olmayan Klinik Araştırmalar Etik Kurulu tarafından incelenmiş olup başvurunun etik açıdan uygun olduğuna katılanların oy birliği ile karar verilmiştir.</p>	