

DERİN ÖĞRENME YÖNTEMLERİYLE BİR UYGULAMA

Hazırlayan :

Öğrenci No :

Teslim Tarihi :

Ders :

Danışman :

**ALP DOĞAN FİDAN**

**090180320**

**05.06.2023**

**MAT 4901**

**PROF. DR. MUSTAFA NADAR**

# İçindekiler Tablosu

[***İçindekiler Tablosu 2***](#_Toc136486713)

[***Şekiller Tablosu 3***](#_Toc136486714)

[***1. Giriş 5***](#_Toc136486715)

[***2. Konvolüsyonel Sinir Ağları 6***](#_Toc136486716)

[***2.1. Makine Öğrenmesi 6***](#_Toc136486717)

[***2.1.1. Öğrenme Çeşitleri 7***](#_Toc136486718)

[***Denetimli Öğrenme (Supervised Learning) 7***](#_Toc136486719)

[***Denetimsiz Öğrenme (Unsupervised Learning) 8***](#_Toc136486720)

[***2.1.2. Modelin Öğrenmesi 8***](#_Toc136486721)

[***Kayıp Fonksiyonu (Loss Function) 8***](#_Toc136486722)

[***Optimizasyon Fonksiyonu (Optimization Function) 9***](#_Toc136486723)

[***2.1.3. Yetersiz Öğrenme (Underfitting) ve Aşırı Öğrenme (Overfitting) 11***](#_Toc136486724)

[***2.1.4. Yapay Sinir Ağları 12***](#_Toc136486725)

[***Tek Katmanlı Algılayıcı (Perceptron) 13***](#_Toc136486726)

[***Çok Katmanlı Algılayıcı (Multi Layer Perceptron) 15***](#_Toc136486727)

[***2.2. Derin Öğrenme 17***](#_Toc136486728)

[***2.3. Konvolüsyonel Sinir Ağları 18***](#_Toc136486729)

[***2.3.1. Konvolüsyonel Sinir Ağları Mimarisi 21***](#_Toc136486730)

[***Girdi Katmanı 21***](#_Toc136486731)

[***Konvolüsyon (Filtre) Katmanı 21***](#_Toc136486732)

[***Toplu Normalizasyon (Batch Normalization) Katmanı 25***](#_Toc136486733)

[***Rektifiye Edilmiş Doğrusal Birim Katmanı (Rectified Linear Units Layer) 25***](#_Toc136486734)

[***Ortaklama Katmanı (Pooling Layer) 27***](#_Toc136486736)

[***Düzleştirme Katmanı (Flattening) 28***](#_Toc136486737)

[***Tam Bağlı Katman 29***](#_Toc136486738)

[***Sönümleme (Dropout) Katmanı 29***](#_Toc136486739)

[***Sınıflandırma Katmanı 30***](#_Toc136486740)

[***Kayıp Fonksiyonu 30***](#_Toc136486742)

[***Kaynakça 31***](#_Toc136486744)

# Şekiller Tablosu

[Şekil 1: Yapay Zeka, Makine Öğrenmesi, Derin Öğrenme ve Yapay Sinir Ağlarına ilişkin şema (Carpenter, Cohen, Jarrell, & Huang, 2018) 5](#_Toc136821251)

[Şekil 2: Küçük, uygun ve aşırı büyük öğrenme oranlarının seçilmesi (Cayla, 2021) 9](#_Toc136821252)

[Şekil 3: Sırayla Underfitting, Düzgün Öğrenme ve Overfitting karşılaştırması (Forjan, 2021) 10](#_Toc136821253)

[Şekil 4: İnsan sinir hücresinin yapısı (Agatonovic-Kustrin & Beresford, 2000) 11](#_Toc136821254)

[Şekil 5: Biyolojik Sinir Ağları ve Yapay Sinir Ağları bileşenlerinin denklikleri (Artificial Neural Network Tutorial) 12](#_Toc136821255)

[Şekil 6: Tek Katmanlı Algılayıcı modeli (Güzel, 2018) 12](#_Toc136821256)

[Şekil 7: Çok Katmanlı Algılayıcı modeli (Lee, Kim, & Lee, 2020) 14](#_Toc136821257)

[Şekil 8: Geleneksel Makine Öğrenmesinde ve Derin Öğrenmede özellik seçimi karşılaştırması (Alzubaidi, ve diğerleri, 2021) 16](#_Toc136821258)

[Şekil 9: ILSVRC-2010 verisetine ait 8 adet test görüntüsü ve AlexNet modelinin görüntüler için sunduğu yüksekten küçüğe sıralanmış en olası 5 etiket tahmini (Krizhevsky, Sutskever, & Hinton, 2017) 17](#_Toc136821259)

[Şekil 10: İnsana kıyasla derin öğrenme performansı (Alzubaidi, ve diğerleri, 2021) 18](#_Toc136821260)

[Şekil 11: 3x3 kernel'e sahip 13 adet Konvolüsyon katmanı ile 3 adet Tam Bağlı Katman'dan oluşan VGG-16 mimarisi (Sharma, 2020) 18](#_Toc136821261)

[Şekil 12: Otonom sürüşlerde nesne tespiti (Barla, 2023) 19](#_Toc136821262)

[Şekil 13: CNN modelinin örnek gösterimi (Anding, Haar, Polte, Walz, & Notni, 2019) 20](#_Toc136821263)

[Şekil 14: Resim üzerinde filtre etkileri (Doğan, 2020) 21](#_Toc136821264)

[Şekil 15: CNN'de farklı katmanlarda meydana gelen nesne temsilleri (İnik & Ülker, 2017) 22](#_Toc136821265)

[Şekil 16: Receptive field ile Kernel'in nokta çarpımı (Singh, Meitei, & Majumder, 2020) 23](#_Toc136821266)

[Şekil 17: Örnek 96 Adet Konvolüsyon Kernel’i (Krizhevsky, Sutskever, & Hinton, 2017) 24](#_Toc136821267)

[Şekil 18: 4×4 boyutundaki matrisin ReLU fonksiyonuna girmeden önceki ve girdikten sonraki durumu (Taşhan, 2017) 25](#_Toc136821268)

[Şekil 19: Giriş görüntüsünün sırasıyla konvolüsyon ve ReLU katmanlarından geçtikten sonraki çıktı görüntüleri (İnik & Ülker, 2017) 25](#_Toc136821269)

[Şekil 20: Stride değeri 2 iken 2×2 boyutlu max pooling ve average pooling işlemleri (Raitoharju, 2022) 26](#_Toc136821270)

[Şekil 21: ReLU katmanından çıkan görüntünün Pooling katmanında işlem gördükten sonraki çıkan görüntüsü (İnik & Ülker, 2017) 27](#_Toc136821271)

[Şekil 22: Matrisin Düzleştirme (Flattening) Katmanı ile vektör haline getirilmesi (Keskin, 2022) 27](#_Toc136821272)

[Şekil 23: Basit bir Tam Bağlı Katman yapısı (Keskin, 2022) 28](#_Toc136821273)

[Şekil 24: Solda 2 gizli katmandan oluşan bir sinir ağı ve sağda Dropout katmanının uygulanmasından sonra oluşan ağ (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014) 28](#_Toc136821274)

# 1. Giriş

Bu giriş bölümünde, tasarımın tanımı, amacı ve kapsamı hakkında bilgi verilecek ve yapılan çalışmaların genel bir özetine yer verilecektir.

Günümüzde, makine öğrenmesi, yapay sinir ağları ve derin öğrenme gibi konseptler bilgisayarlı görü sektöründe önemli bir etkiye sahiptir. Bu bitirme çalışmasında; 2010’lu yılların başından itibaren obje tanıma, obje sınıflandırma ve örüntü tanıma gibi bilgisayarlı görü görevlerindeki göze çarpan başarısı nedeniyle derin öğrenme alanında önemli bir yere sahip olan Konvolüsyonel Sinir Ağları (CNN: Convulotional Neural Network) hakkında kapsamlı bir literatür araştırması yapılacaktır. Bu araştırmayla CNN’in tarihsel gelişimi ile birlikte teoriksel alt yapısının da tanıtılması ve bu bilgiler ışığında ortaya bir uygulama çıkarılması amaçlanmaktadır.

Bu dönemde, tasarım projesinin kapsamında mevcut bilimsel makaleleri, konferans bildirileri ve diğer kaynaklar incelenerek, basit bir CNN modelinin genel yapısı ve şu ana kadar oluşturulan çeşitli popüler state-of-the-art CNN modellerinin mimari özellikleri ve uygulama alanları araştırılmış ve tanıtılmıştır.

CNN’in kullanım alanları oldukça geniştir ve birçok sektörde birçok şirket tarafından da kullanılarak çözümler üretildiği bilinmektedir. Başlıca kullanım alanları tarımda fitopatoloji (Toda & Okura, 2019) ve entomoloji (Høye, ve diğerleri, 2021); medikalde radyoloji (Raitoharju, 2022); otomotiv sektöründe otonom sürüş; diğer sektörlerde ise uzaktan algılama (Kattenborn, Leitloff, Schiefer, & Hinz, 2021), yaban hayatında hayvanların tespiti ve şüpheli kişilerin yüz tanıması olarak verilebilir (Karimi, 2021).

2021 yılında gerçekleştirilen bir çalışmada, 100 Çinli vatandaştan oluşan bir grup yardımıyla 15 çeşit Çin rakam birimini içeren toplam 15000 adet 64×64 boyutunda görüntüden oluşan bir veri seti hazırlanmıştır. Diğer yandan 256, 128 ve 64 adet 3×3’lük filtre içeren 3 adet konvolüsyon katmanından oluşan CNN mimarisi tasarlanmıştır. Bu konvolüsyon katmanlarından sonra ise aktivasyon katmanı olarak ReLU, pooling katmanı olarak ise max pooling kullanılmıştır. Tasarlanan mimari toplamda 8,792,783 parametre içermektedir. Sonuç olarak ilgili veriseti ise bu CNN mimarisi kullanılarak eğitilmiş olup test verisinde %97 doğruluk oranı ile %13 hata oranı elde edilmiştir. (Kayalı & Omurca, 2021).

CNN’in başarısı farklı görüntü boyutlarında ve yeterli-yetersiz koşullara göre değişebilmektedir. Yine 2021 yılında yayımlanan başka bir makalede, aynı koşullar altında fakat farklı nitelikte veri setleri üzerinde geleneksel makine öğrenmesi algoritması olan Support Vector Machine (SVM) ile CNN modelleri eğitilip karşılaştırılmıştır. Veri setindeki görüntülerin boyutu sırayla 64×64, 128×128 ve 256×256 alındığında SVM modelinin sınıflandırma başarısı yaklaşık olarak %61-%64 arasında değişirken CNN’in sınıflandırma doğruluğunun %71’den %95’e ulaştığı tespit edilmiştir. Yine aynı çalışmada COREL1000 veri setinin küçük bir örneği ile modeller eğitildiğinde SVM %86 doğruluk sağlarken CNN %83 doğruluk sağlamıştır. MNIST’den alınan büyük veri seti üzerinde ise SVM ve CNN modelleri sırayla %88 ve %98 doğruluğa sahip olmuştur (Wanga, Fan, & Wang, 2021).

Bir sonraki dönem, bu tasarım çalışması kapsamında bilgisayarlı görü alanında bir sınıflandırma problemi tanıtılıp bu problem için Python programlama dili kullanılarak bir uygulama tasarlanması ve başarının ölçülmesi planlanmaktadır.

# 2. Konvolüsyonel Sinir Ağları

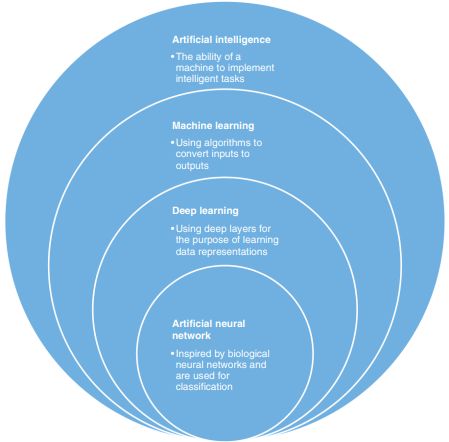
Bu bölümde Makine Öğrenmesi, Yapay Sinir Ağları ve Derin Öğrenme kavramlarına kısaca değinilip ardından sırasıyla CNN’in kısa tarihi, CNN’in ana mimarisi ve son olarak CNN’den türeyen başarısı kanıtlanmış mimarilerden bahsedilecektir.

## Makine Öğrenmesi

Yapay zeka, insanların sahip olduğu mantık yürütebilme yeteneğinin makinelere kazandırılarak makinelerin adeta bir insan gibi davranmasını ve insan gibi düşünerek karar vermesini sağlamaya çalışan bir disiplindir.

1950 yılında "Makineler Düşünebilir mi?" sorusunu soran Alan Turing, kendisinin geliştirdiği Turing Test’i ile camiada büyük bir yankı uyandırmıştır. Turing Test'i, Yapay Oyun (The Imitation Game) olarak da bilinir ve makinenin insanı kandırabilme yeteneğini ölçer. Eğer bir makine veya bir yazılım insan gibi davranarak başka bir insanı kandırabiliyorsa test başarılı anlamına gelmektedir (French, 2000).

Şekil 1’de şemada Yapay Zeka, Makine Öğrenmesi, Derin Öğrenme ve Yapay Sinir Ağları kavramlarının hiyerarşisi ve ilişkisi gösterilmektedir.



Şekil : Yapay Zeka, Makine Öğrenmesi, Derin Öğrenme ve Yapay Sinir Ağlarına ilişkin şema (Carpenter, Cohen, Jarrell, & Huang, 2018)

Teknolojinin gelişmesi ile birlikte artan internet ve sosyal medya kullanımları sonucunda her gün sayısız miktarda veri meydana gelmektedir. Bu veriler, yapay zekanın bir alt dalı olan makine öğrenmesi alanında kullanılarak anlamlı desenler ve özellikler keşfedilebilir. Makine öğrenmesi disiplini yüksek matematik, istatistik ve programlama bilgisi gerektirmektedir.

Bir makine öğrenmesi modeli, veri kümeleri üzerinden eğitilerek anlamlı desenler ve özellikler keşfeder. Eğitilen modele gelecekte yeni bir girdi verildiğinde daha önce analizini gerçekleştirerek anlamlı desenler ve özelliklerin çıkarımını yaptığı veri kümesine dayalı bir tahmin çıktısı ortaya koyar.

Modeller oluşturulurken seçilen veri kümesine ve problemin türüne özel yaklaşımlar dikkate alınmaktadır. Bu başlık altında bazı bilinen problem türleri ve birkaç öğrenme yöntemi ele alınmıştır.

### Öğrenme Çeşitleri

Makine öğrenmesi dünyasında tarih boyunca sınıflandırma, regresyon ve kümeleme gibi başka birçok probleme çözüm getirebilmek adına farklı öğrenme yöntemleri geliştirilmiştir. Bu teknikler sayesinde veri setlerindeki neden-sonuç ilişkilerinin anlamlandırılması ve makine öğrenmesi modellerine tahmin etme kapasitesi kazandırılması sağlanmıştır. Bu öğrenme stratejilerinden en popüler olanları denetimli öğrenme ve denetimsiz öğrenmedir.

#### Denetimli Öğrenme (Supervised Learning)

Denetimli öğrenme tekniğindeki amaç makine öğrenmesi modellerini, etiketlenmiş veriler üzerinden eğitmektir. Veri kümesi, eğitim (train) ve test olmak üzere iki kümeye ayrılır ve iki küme de etiketlenmiş verilerden oluşmaktadır. Model, eğitim sürecinde train kümesindeki öznitelik değişkenleri ile hedef (target) değişkenlerin arasındaki ilişkiyi öğrenir. Modelin başarısı train ve test kümesine göre verdiği tahminler üzerinde çeşitli metrikler aracılığıyla hesaplanır. Bu öğrenme yöntemi regresyon ve sınıflandırma problemlerinde kullanılmaktadır. Regresyon, sürekli (continuous) çıktıyı tahmin etme problemidir. Sınıflandırma ise ayrık (discrete) kategorik çıktıların tahmin edilmesi problemidir. Aslında sınıflandırma da bir regresyon çeşididir; çıktı (response) eğer kategorikse bu tip regresyon, lojistik regresyon olarak adlandırılır ve regresyonla elde edilen değer kategorik değere yuvarlanır.

Denetimli öğrenme metodu kullanılarak literatürde birçok alanda çalışmalar yürütülmüştür ve yürütülmeye de devam edilmektedir. Bu yöntemin başlıca kullanıldığı alanlar aşağıdaki maddelerde belirtilmiştir (Das, Dey, Pal, & Roy, 2015):

* E-mail spam filtreleme,
* El yazısı tanıma,
* Yüz tanıma,
* Konuşma tanıma,
* Doğal dil işleme,
* Anomali tespiti,
* Saldırı tespit sistemi,
* Epilepsi nöbet sistemi.

#### Denetimsiz Öğrenme (Unsupervised Learning)

Denetimsiz öğrenme stratejisinde modeller bir rehber yardımı olmadan etiketsiz veriler üzerinde eğitilir. Hedef bilgisi, bu öğrenmede kullanılan veri setinde bulunmaz. Verinin kendisinden anlam çıkarılır. Benzer özelliklere sahip veriler gruplandırılır. Kümeleme (clustering) problemlerinde denetimsiz öğrenme tercih edilmektedir.

Denetimsiz öğrenme metodunun başlıca kullanıldığı alanlar aşağıdaki maddelerde belirtilmiştir (Das, Dey, Pal, & Roy, 2015):

* DNA sınıflandırma,
* Sosyal medya analizi,
* Piyasa bölünmesi (Market Segmentation)
* Astronomik veri analizi,
* Kanser teşhisi,
* Konuşma etki tespiti.

### Modelin Öğrenmesi

Makine öğrenmesi, verilerden öğrenme yoluyla modele karar verme yeteneği kazandırmayı amaçlayan bir alandır. Modeller bir veya daha fazla parametrelerden oluşmaktadır. Modelin başarısını bu parametreler belirler ve bu parametrelerin düzgün ayarlanması modelin performansını oldukça yükseltir. Modelin öğrenmesinde en önemli etkenlerden ikisi kayıp fonksiyonları ve optimizasyondur.

#### Kayıp Fonksiyonu (Loss Function)

Kayıp fonksiyonu, modelin tahmin ettiği değerler ile gerçekte beklenen değerler arasındaki hata farkının hesaplayarak uygun optimizasyon yöntemi ile modelin iyileştirilmesini sağlar. Maliyet (Cost Function) olarak da bilinir. Bilinen kayıp fonksiyonlarından biri olan ortalama hata karesi (MSE), Denklem ***(1)***’de gösterilmiştir.

|  |  |  |
| --- | --- | --- |
|  |  | **(****1)** |

Denklemde modelin tahmin değerini, gerçek değeri, N gözlem miktarını ifade etmektedir.

#### Optimizasyon Fonksiyonu (Optimization Function)

Optimizasyon, bir fonksiyonun parametrelerini en iyilemeyi sağlayan bir süreçtir. Makine öğrenmesinde temel optimizasyon yöntemleri, birinci dereceden optimizasyon yöntemleri (First-Order Methods), yüksek dereceden optimizasyon yöntemleri (High-Order Methods) ve türevsiz optimizasyon yöntemleri (Derivative-Free optimization methods) olarak üç gruba ayrılabilir. Birinci dereceden alınan türev metotları optimizasyon süreçlerinde ise en yaygın kullanılan metotlardandır (Sun, Cao, Zhu, & Zhao, 2019).

Dereceli azalma (gradient descent) birinci dereceden türeve dayanan, minimizasyon problemlerinde kullanılan bir optimizasyon algoritmasıdır. Literatürde bulunan en popüler dereceli azalma çeşitlerinden 2 tanesi şunlardır:

* Toplu Dereceli Azalma (Batch Gradient Descent),
* Rastgele Dereceli Azalma (Stochastic Gradient Descent).

Batch gradient descent yönteminin fikri, değişkenlerin kayıp fonksiyonunun gradyanlarının (tersi) yönünde iteratif olarak güncellenmesidir. Güncelleme, amaç fonksiyonunun optimal değerine adım adım yavaşça yaklaşmak için yapılır (Sun, Cao, Zhu, & Zhao, 2019).

Diyelim ki öğrenme oranımız, kayıp fonksiyonumuz ve iyileştirilecek parametreli fonksiyonumuz olsun. Fonksiyonlar sırasıyla Denklem (2) ile Denklem (3)’de verilmiştir.

|  |  |  |
| --- | --- | --- |
|  |  | **(****2)** |

|  |  |  |
| --- | --- | --- |
|  |  | **(****3)** |

Burada eğitimdeki gözlem sayısını, girdi öznitelikleri, olmak üzere bağımsız değişkeni ve hedef çıktıyı temsil etmektedir. Dereceli azalma yöntemi Denklem ***(4)*** ve Denklem ***(5)***’de verilen 2 adımı yerel optimuma yakınsayana kadar devam etmektedir (Sun, Cao, Zhu, & Zhao, 2019).

İlk adımda fonksiyonun ’e göre 1.dereceden kısmi türevi alınır:

|  |  |  |
| --- | --- | --- |
|  |  | **(****4)** |

İkinci adımda yeni değer olmak üzere her bir parametresi, gradyenin negatif yönünde maliyet fonksiyonunun minimize edilmesi için güncellenir:

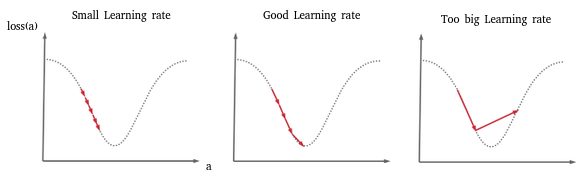
|  |  |  |
| --- | --- | --- |
|  |  | (5) |

Batch Gradient Descent’nin çalışma karmaşıklığı her iterasyon için ’dir. Hacimli verilerle uğraşırken yüksek matematiksel hesaplamalar gerektirmesi yüzünden çalışma karmaşıklığı yine her iterasyon için olan Stochastic Gradient Descent (SGD)’in doğmasına sebep olmuştur. SGD, veri kümesindeki örnekleri rastgele seçer. Bir önceki yönteme göre daha hızlı olsa da gürültü yaratabilmektedir. SGD, Denklem ***(6)***’de verilmiştir (Sun, Cao, Zhu, & Zhao, 2019).

|  |  |  |
| --- | --- | --- |
|  |  | **(****6)** |

Bu başlık altında dikkat edilmesi gereken en önemli hususlardan bir tanesi, öğrenme katsayısı dediğimiz hiperparametrenin uygun seçilmesidir. 'nin küçük seçilmesi lokal minimuma küçük adımlarla gidilmesine ve böylece modelin çok yavaş öğrenmesine neden olur. Tam tersi, öğrenme katsayısının yüksek seçilmesi ise genellikle lokal minimumdan sapmaya neden olur (Ruder, 2017).

Şekil 2’de öğrenme oranına göre lokal minimuma yaklaşma biçimleri gösterilmektedir. Optimum öğrenme oranında lokal minimuma başarıyla yaklaştığı, düşük öğrenme oranında adımların çok yavaş ilerlendiği, yüksek öğrenme oranında ise lokal minimumdan sapabileceği anlatılmaktadır.

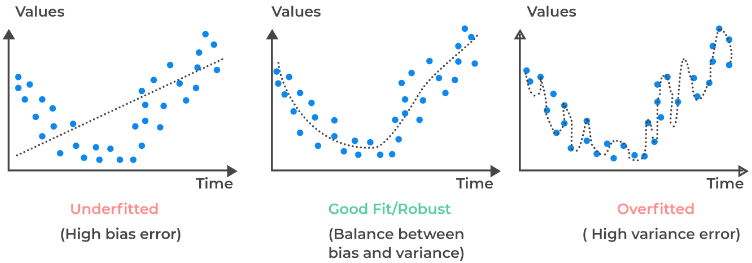


Şekil : Küçük, uygun ve aşırı büyük öğrenme oranlarının seçilmesi (Cayla, 2021)

Yapay sinir ağlarında gradient descent yöntemleri kullanılarak ileri yönlü yayılım işleminin sonucunda elde edilen kayıp fonksiyonunun 1. dereceden türevinin alınıp rastgele adımlarla lokal minimuma yakınsaması sağlanır. Ağırlıklar bu işlem sayesinde güncellenmektedir (Ruder, 2017).

### Yetersiz Öğrenme (Underfitting) ve Aşırı Öğrenme (Overfitting)

Modelin düzgün öğrenebilmesi için hem düşük varyans hem de düşük taraflılığa sahip olması gerekmektedir. Genel olarak makine öğrenmesinde modeller bazen yetersiz bazen de aşırı öğrenebilmektedir. Yetersiz öğrenme yüksek yanlılık (bias) sonucunda gerçekleşir. Aşırı öğrenme ise yüksek varyans sebebiyle yaşanır. Bahsedilen iki durum da modelin başarısız performans göstermesine neden olur. Örneğin, aşırı öğrenmiş bir makine öğrenmesi modeli gürültüleri de barındırmasından ötürü daha önce gözlemlenmemiş bir veri için çıktıyı yanlış tahmin etme eğilimindedir. Aşırı öğrenme durumu, eğitim setinin doğruluğunun test setinin doğruluğundan bir hayli yüksek çıkmasıyla farkedilebilir. Şekil 3’te regresyon problemini içeren bir örnek üzerinden underfitting, overfitting ve düzgün öğrenme farkı gösterilmiştir.

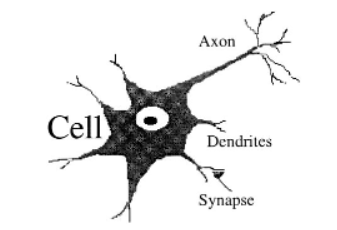


Şekil : Sırayla Underfitting, Düzgün Öğrenme ve Overfitting karşılaştırması (Forjan, 2021)

### Yapay Sinir Ağları

Yapay Sinir Ağları (ANN: Artificial Neural Network), gelişmiş varlıklar olan insanlar ve hayvanların bilgi işleme şeklini taklit etmek amacıyla sinir sisteminin önemli parçası olan beyin organının biyolojik yapısından esinlenerek matematiksel modellenmesi ile ortaya çıkmış bir bilgisayar kavramıdır. Tıpkı insanlarda olduğu gibi deneyim yoluyla öğrenirler. ANN’leri iyi anlamak için ilk olarak beyindeki işleyiş hakkında bir bilgi sahibi olmak gerekmektedir.

Beyin, örüntü tanıma, tahmin gibi konularda son derece etkileyici bir araçtır ve sonsuz çeşitlilikteki girdi örüntülerini öğrenip tanıyabilir. Beyinde ortalama olarak 100 milyar tane nöron adı verilen sinir hücreleri bulunmaktadır. Her bir nöronun da diğer nöronlarla sayısı yaklaşık olarak 1000 ile 10000 arasında değişen bağlantılarla birbirlerine bağlı olduğu düşünülmektedir (Agatonovic-Kustrin & Beresford, 2000). Bilgi ve deneyimin tümü, nöronların aralarındaki bağlantılar tarafından kodlandığına inanılır, ancak bu konu hakkında hala tartışmalar sürmektedir (Noriega, 2005). Bir biyolojik nöronun yapısı Şekil 4’deki modelde gösterilmiştir.



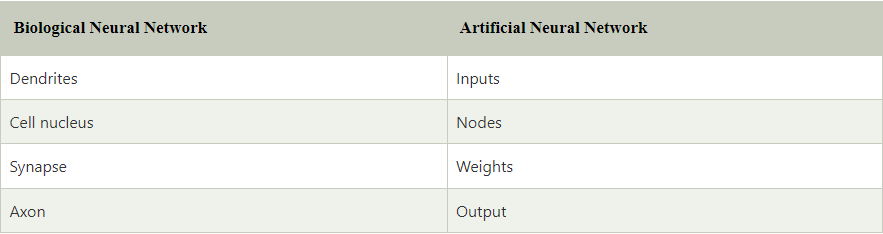
Şekil : İnsan sinir hücresinin yapısı (Agatonovic-Kustrin & Beresford, 2000)

Beynin temel yapı taşları olan nöronlar, hücre aktivitesini kontrol eden çekirdek içeren bir hücre gövdesi, bilgiyi hücreye taşıyan birçok ince tel olan dendritler ve sinyali uzaklaştıran daha uzun bir tel olan aksondan oluşmaktadır. Bir nöron diğer nöronlarla sinaps adı verilen bir bağlantı noktasında birleşir. Reseptörlerden herhangi bir uyarılma sonucunda oluşan elektro-kimyasal sinyaller (impuls), bir sinir hücresine dentritlerden girer. Dentritlerden gelen tüm sinyaller hücre gövdesinde toplanır. Eylem potansiyeli akson boyunca sinapslara ulaştırılır. Hücrede gerçekleştirilen eşik kontrolü sonucunda bu sinyaller diğer nöronlara ya tamamen iletilebilir ya da hiçbir şekilde iletilemez.

Nöronlar tamamen bağlı (fully-connected) bir ağda düzenlenir ve uyarıları alıp göndererek adeta bir mesajcı gibi davranırlar. Sinir ağlarının bağlantısının sonucunda öğrenme, tahmin ve tanıma gibi yeteneği olan zeki bir beyin ortaya çıkarır (Agatonovic-Kustrin & Beresford, 2000).

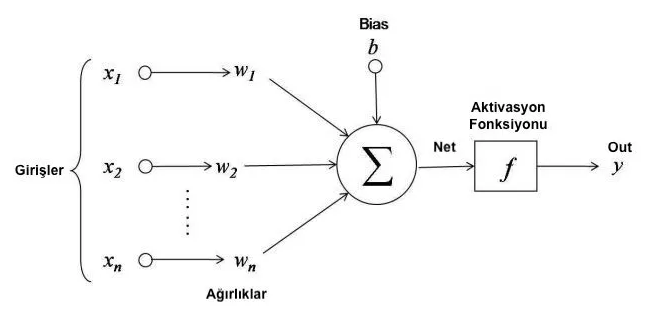
#### Tek Katmanlı Algılayıcı (Perceptron)

Yapay sinir ağlarının temel yapıtaşı yapay nöron hücreleridir. 1943 yılında nörobilimci Warren S. McCulloch ve mantık bilimci Walter Pitts, birçok sinir hücresinin bir araya bağlanmasıyla beynin nasıl karmaşık desenler üretebileceğini anlamaya çalışmışlar ve sundukları MCP nöron modeli ile yapay sinir ağlarının gelişimine önemli katkıda bulunmuşlardır (Marsalli). Daha sonra 1957 yılında Frank Rosenblatt tarafından yapay nöron modeli olarak bilinen tek katmanlı Perceptron tanıtılmıştır (Rosenblatt, 2017). Perceptron, insan ve hayvan beynindeki gerçek nöronlara benzer şekilde girdi sinyallerini alır, bu sinyalleri çarpar ve toplar, ardından da aktivasyon işlemlerine tabi tutar. Tek katmanlı perceptron yapısı Şekil 5’de gösterilmiştir.



Şekil : Biyolojik Sinir Ağları ve Yapay Sinir Ağları bileşenlerinin denklikleri (Artificial Neural Network Tutorial)

Yapay nöron bileşenlerinin biyolojik nörondaki hangi bileşenlere karşılık geldiği de Şekil 6’de gösterilmiştir.



Şekil : Tek Katmanlı Algılayıcı modeli (Güzel, 2018)

Algılayıcıya giren girdi değerleri olsun, her bir girdi değerine denk düşen ağırlık bulunmaktadır. Girdiler ile ağırlıkların çarpımına skalar bir değer olan yanlılık değerinin de eklenmesiyle Denklem ***(7)***’de verilen net değer oluşmaktadır.

|  |  |  |
| --- | --- | --- |
|  |  | **(****7)** |

Denklem ***(7)***’den elde edilen değeri bir aktivasyon fonksiyonuna sokulur ve çıkan sonuca göre nöronun ateşlenip ateşlenmeyeceğinin kararı verilir. Aktivasyon fonksiyonlarından biri olan adım fonksiyonu değerini girdi alacak şekilde Denklem ***(8)***’de ifade edilmektedir. Aktivasyon fonksiyonun sonucu olan değeri 0 ise nöron ateşlenmez, 1 ise nöron ateşlenir.

|  |  |  |
| --- | --- | --- |
|  |  | **(****8)** |

Perception’ın öğrenme sürecinde yaygın olarak delta kuralı kullanılır. Delta kuralında, gerçekleşen değer ve beklenen değerin hata kareler yöntemine göre türevinden faydalanılarak ağırlıklar güncellenir. Ağırlıklar, hatanın türevine göre hatayı minimize etmeye çalışan gradient descent yöntemine göre iyileştirilir. Modele verilen girdilere göre, modelin tahmin ettiği değer , gerçek değer ve modele tanımlanan öğrenme oranı olsun. Denklem ***(9)***’de delta kuralı tanıtılmıştır (Noriega, 2005).

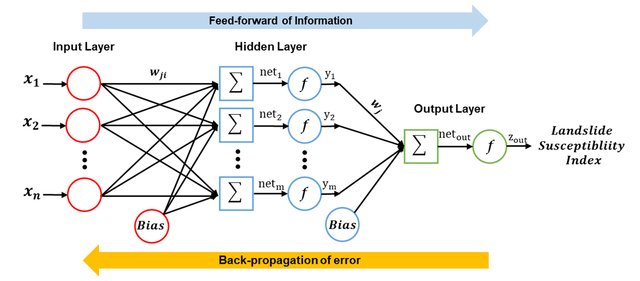
|  |  |  |
| --- | --- | --- |
|  |  | **(****9)** |

Model, Denklem ***(7)***, Denklem ***(8)*** ve Denklem ***(9)*** iterasyonlarca tekrar edilerek hatanın minimize edilmesi sağlanana kadar eğitilir.

Tek katmanlı perceptronlar ile doğrusal ayrılabilir (linearly seperable) olan mantıksal OR kapısı ve mantıksal AND kapısı gibi sınıflandırma problemleri başarıyla çözülebilmiştir. Fakat bu tür bir sistem XOR gibi doğrusal ayrılabilir olmayan sınıflara çözüm bulmakta sınıfta kalmaktadır (Noriega, 2005).

#### Çok Katmanlı Algılayıcı (Multi Layer Perceptron)

Çok katmanlı algılayıcı birden fazla algılayıcının bir araya gelmesiyle oluşan yapay sinir ağı modelidir. Tek katmanlı algılayıcıda girdi katmanı ve çıktı katmanı bulunurken bu sistemde girdi katmanı, gizli katman ve çıktı katmanı olmak üzere 3 katman bulunmaktadır. Gizli katman, bir veya daha fazla katman içerebilmektedir. Lineer olmayan problemlere tahmin üretebilmek amacıyla tanıtılmıştır. Tek katmanlı algılayıcıyla çözülemeyen doğrusal olmayan XOR kapısı problemi, çok katmanlı algılayıcı modeli ile çözülebilmiştir (Noriega, 2005). Şekil 7’de modelin yapısı gösterilmektedir. Ayrıca bu modelde aynı katmanlarda bulunan düğümler arasında bir bağlantı bulunmamaktadır.



Şekil : Çok Katmanlı Algılayıcı modeli (Lee, Kim, & Lee, 2020)

Girdi katmanından dış dünyadan alınan veriler sisteme girer. Girdi değerler, sırasıyla gizli katman ve çıktı katmanlarda bulunan algılayıcılarda çarpım ile toplama operasyonlarına dahil olarak net değerler hesaplanır. Hesaplanan net değerler üzerinde de lineer veya lineer olmayan aktivasyon fonksiyonları uygulanarak çıktı değer(ler) üretilmiş olur. Bu akışa ileri yönlü yayılım adı verilir.

|  |  |  |
| --- | --- | --- |
|  |  | **(****10)** |

|  |  |  |
| --- | --- | --- |
|  |  | **(****11)** |

Denklem ***(10)*** ve Denklem ***(11)***’de hesaplamaların yapıldığı fonksiyonlar verilmiştir. Burada girdi değerini, ve sırayla gizli ve çıktı katmanından çıktı değerlerini, ve sinaptık ağırlıkları, ve yanlılık değerlerini, n ve m değerleri sırayla girdi katmanındaki ve gizli katmandaki nöron sayılarını, ve operatörü ise aktivasyon fonksiyonunu temsil etmektedirler (Lee, Kim, & Lee, 2020).

Modelin öğrenme süreci 2 aşamadan oluşur:

* İleri yönlü yayılım (Forward Propagation),
* Geri yönlü yayılım (Backward Propagation).

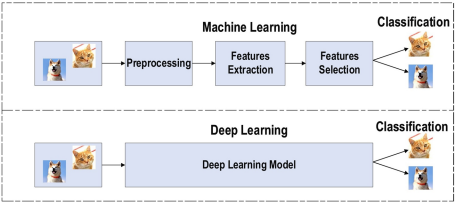
Süreçte ileri yönlü yayılım akışıyla girdi katmanından başlanılarak çıktı katmanına gidilir ve akış sonra erer. Sonrasında geri yönlü yayılım başlar. Bu algoritma, başlangıçta rastgele atanan ağırlıkları (Dongare, Kharde, & Kachare, 2012) ağın tahmin ettiği değerler ile gerçek değerlerinin kullanılmasıyla hesaplanan MSE kayıp fonksiyonu üzerinden gradient descent yöntemi ile güncellenir. Ağırlıkların güncellenmesi işlemi, MSE hatası belli bir eşiğe ulaşana devam eder (Lee, Kim, & Lee, 2020).

Bishop'a (1995) göre geriye doğru yönelim algoritması adımları aşağıdaki gibidir (Gardner & Dorling, 1998):

1. Ağ ağırlıkları başlatılır (initialization),
2. Eğitim verilerinden giriş vektörü ağa girdi olarak sağlanır,
3. Giriş vektörü ağ üzerinden ileriye doğru iletilerek çıktı tahmini elde edilir,
4. Tahmin ile olması beklenen çıktının hata sinyali hesaplanır,
5. Hata sinyali ağ üzerinden geriye doğru iletilir,
6. Toplam hatanın minimize etmek için ağırlıklar ayarlanır,
7. Toplam hata kabul edilebilir düzeye düşene kadar 2’den 7’ye kadar olan adımlar tekrarlanır.

## Derin Öğrenme

Makine Öğrenmesi alanında verideki özelliklerin seçilmesi modellerin başarısını önemli ölçüde etkileyen bir adımdır. Makine Öğrenmesinde genellikle uzman bilgisine göre analiz edilerek el yordamıyla özellikler seçilirken Derin öğrenmede eğitim sürecinde özellikler otomatik olarak tespit edilmektedir (Aslan, 2018). Derin öğrenme bu sayede hem manuel özellik seçme esnasında meydana gelebilecek hatalardan kurtarır hem de alan uzmanlığı gerektirmeden otomatik özellik seçimi sağlar. Şekil 8’de geleneksel makine öğrenmesin ve derin öğrenmedeki işleyiş farklılığı gözler önüne serilmiştir.



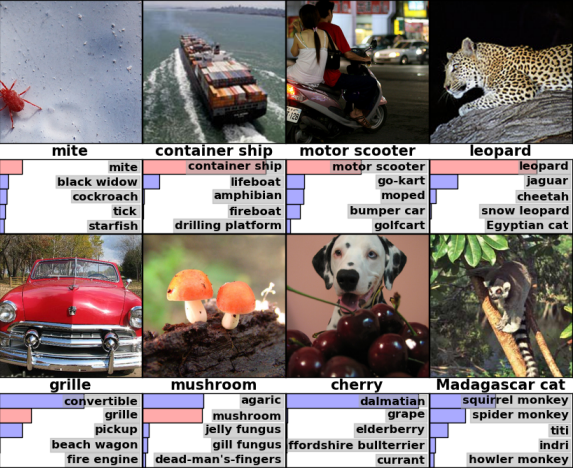
Şekil : Geleneksel Makine Öğrenmesinde ve Derin Öğrenmede özellik seçimi karşılaştırması (Alzubaidi, ve diğerleri, 2021)

Derin öğrenme algoritmaları, özellikle çok katmanlı sinir ağları, genel olarak makine öğrenmesine görece eğitim sürecinde daha fazla veriye ihtiyaç duymaktadır ve ayrıca gerek gerçekleştirilen kompleks hesaplamalar, gerekse veri tipinin çoğunlukla çok boyutta olması eğitim sürelerinin uzun sürmesine sebebiyet verebilmektedir.

## Konvolüsyonel Sinir Ağları

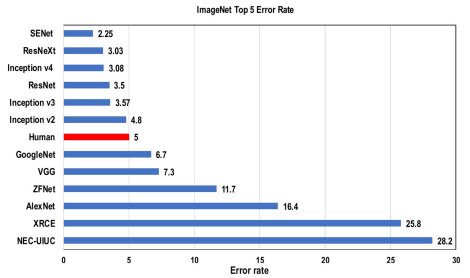
Tarihte CNN kavramı ilk olarak LeCun tarafından ortaya atılmıştır (1989). LeCun, 1998’de görüntü işleme kapsamında MNIST el yazısı rakamlarını tanımayı amaçladığı çalışmayla LeNet adını verdiği CNN mimarisini kullanarak yüksek bir başarı elde etmeyi başarmıştır. Yine de o dönemlerde büyük verilerin neredeyse hiç bulunmadığı ve günümüzde yaygınca kullanılan hesaplama kapasitesi yüksek GPU’ların henüz geliştirilmemiş olması nedeniyle pek kullanılan bir yöntem değildi. Sürekli gelişen teknoloji ile beraber veri kaynaklarının her saniye artması ve sürekli daha da iyileştirilen GPU’lar sayesinde bu engeller ortadan kalkmıştır.

Pascal Visual Object Challenge kapsamında ilk olarak 2010’da başlayan ve yıllık olarak gerçekleştirilmeye devam eden ImageNet Büyük Ölçekli Görsel Tanıma yarışmasının 2012 senesinde düzenlenen karşılaşmasında, ILSVRC-2012 veri seti üzerinde daha önceden %26,1 olan Top-5 hatasının bir CNN modeli olan AlexNet ile birlikte %15.3’e düşürülmesiyle performansını gözler önüne sermiştir. Şekil 9’de ILSVRC-2010 verisetinden alınan 8 adet test görüntüsü ve aynı model tarafından bu görüntüler için belirlenen en olası 5 etiket gösterilmektedir (Krizhevsky, Sutskever, & Hinton, 2017).



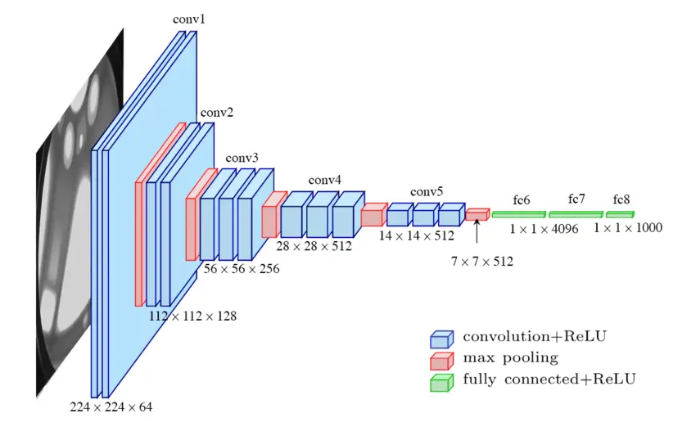
Şekil : ILSVRC-2010 verisetine ait 8 adet test görüntüsü ve AlexNet modelinin görüntüler için sunduğu yüksekten küçüğe sıralanmış en olası 5 etiket tahmini (Krizhevsky, Sutskever, & Hinton, 2017)

Dahası, yine farklı bir yılda düzenlenen ImageNet yarışmasına ait ILSVRC 2015 veri kümesine göre insanlar %5.1 Top-5 hata skoru elde ederken (Russakovsky, ve diğerleri, 2015), ResNet modeli %3.57 Top-5 hata skorunu yakalayarak bir başarıya imza atmıştır (He, Zhang, Ren, & Sun, 2016). Şekil 10’de başarısı kanıtlanmış (state-of-art) mimarilerle insan başarısının karşılaştırılması göz önüne konulmuştur.



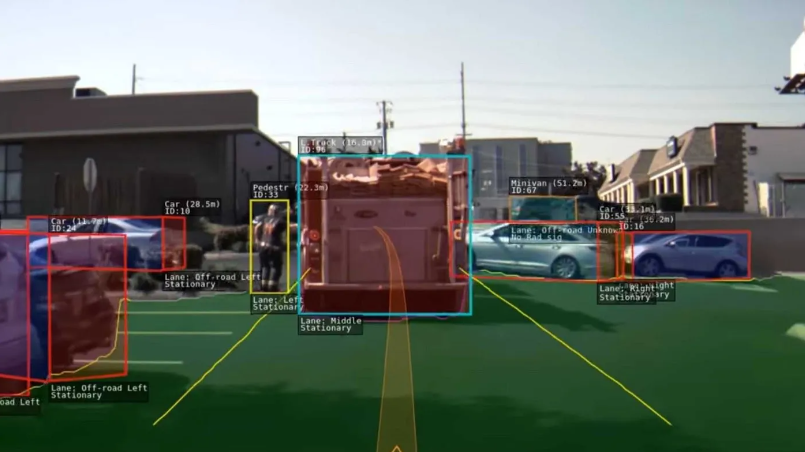
Şekil : İnsana kıyasla derin öğrenme performansı (Alzubaidi, ve diğerleri, 2021)

Burada AlexNet, ZFNet, GoogleNet, ResNet mimarileri sırayla 2012, 2013, 2014 ve 2015 ImageNet şampiyonlarıdır. Ayrıca 2014 yılında GoogleNet’in ardından 2.liği elde eden VGG (VGG-16) modelinin mimarisi Şekil 11’de verilmiştir (Sharma, 2020).



Şekil : 3x3 kernel'e sahip 13 adet Konvolüsyon katmanı ile 3 adet Tam Bağlı Katman'dan oluşan VGG-16 mimarisi (Sharma, 2020)

CNN, günümüzde popüler olan Google, Facebook, Pinterest ve Instagram gibi sosyal medya platformları tarafından yaygın bir biçimde kullanılmaktadır. CNN’ler genel olarak görsel tanıma amacıyla medikal, otomatik sürüş, biyometrik doğrulama, doküman analizi gibi bir çok farklı sektörde yer bulmuştur. Kalp hastalıkları, MR ile beyin tümörü ve kanser tespiti, kişinin yüzünden kimliğini tanıma ve araba destek uygulamaları (yolun, engellerin, arabaların, trafik levhalarının belirlenebilmesi) bu çalışmalara örnek olarak verilebilir (Gandharv, 2022). Otonom sürüş sektöründe Tesla’nın HydraNet’inde, Google Waymo’nun ChauffeurNet’inde ve Nvidia’nın otonom sürüş arabalarında CNN tekniklerinden faydalanılmaktadır (Barla, 2023). Otonom sürüşlerde nesnelerin tespitini belirten örnek bir görüntü Şekil 12’de gösterilmektedir.



Şekil : Otonom sürüşlerde nesne tespiti (Barla, 2023)

### Konvolüsyonel Sinir Ağları Mimarisi

Basit bir CNN mimarisi genel haliyle Girdi Katmanı, Konvolüsyon Katmanı, Havuzlama Katmanı, Tam Bağlı Katman ve Sınıflandırma Katmanından oluşmaktadır. Bu katmanlara ek olarak Batch Normalization, Rektifiye Edilmiş Doğrusal Birim Katmanı ve Dropout Katmanı gibi bir çok başka katmanlar da eklenip çıkarılarak birbirinden farklı başarılı mimariler elde edilebilmiştir. Şekil 13’de örnek bir CNN mimarisi gösterilmiştir.

diyagram içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil : CNN modelinin örnek gösterimi (Anding, Haar, Polte, Walz, & Notni, 2019)

#### Girdi Katmanı

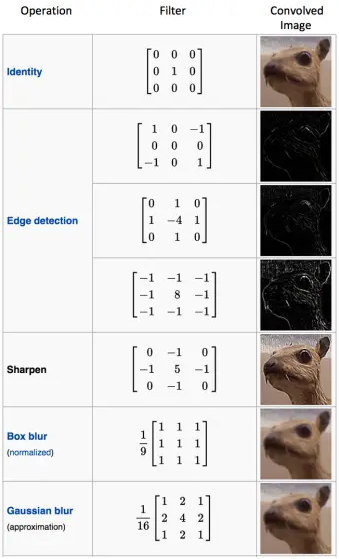
CNN’ler başta görüntü olmak üzere video, ses ve diğer sinyalleri girdi olarak alabilmektedir. Girdi boyutunun büyük olması işlem hacminin de artmasına sebep olmaktadır.

#### Konvolüsyon (Filtre) Katmanı

Geleneksel görüntü işleme tekniklerinde görüntüler çarpraz korelasyon (cross-correlation) ve konvolüsyon gibi bazı operasyonlarla manipüle edilebilmektedirler. Konvolüsyon katmanı, adından da anlaşılacağı üzere Konvolüsyonel Sinir Ağları’na ismini veren katmandır. Konvolüsyon katmanının ismi konvolüsyon olsa da tam tersine aslında çarpraz korelasyon operasyonu uygulanmaktadır (Rosebrock, 2021).

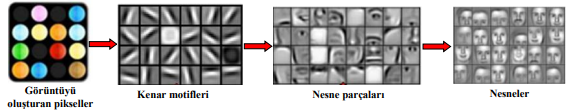
Bu operasyon için çekirdek(kernel) denilen filtre matrisi kullanılmaktadır. Filtre matrislerinin boyutu genellikle 3×3, 5×5, 7×7, 9×9 veya 11×11 olmaktadır. Bu matris yardımıyla girdi olarak verilen verideki öznitelik matrisleri (activation map) çıkarılır. Filtrelerdeki ağırlık değerleri, CNN algoritmasının öğrenme sürecinde güncellenmektedir. Bu değerler güncellendikçe kenarların ve köşelerin tespit edilme kalitesi de artmaktadır. Filtrelerin ağırlık değerleri rastgele veya önceden eğitilmiş modellere göre başlatılabilmektedir.

Filtre matrisinin resim üzerindeki etkisi aşağıdaki Şekil 14’de vurgulanmıştır. Şekilde örnek bir hayvan görseli ile blurlama, keskinleştirme gibi işlemler için kullanılan bazı bilinen filtre matris çeşitleri, ve aynı görselin her bir filtre matrisi tarafından konvolüsyona uğramış son hali gösterilmektedir.



Şekil : Resim üzerinde filtre etkileri (Doğan, 2020)

Konvolüsyon işleminin her uygulanışında farklı özellikler ortaya çıkarılmaktadır. CNN’de kullanılan ilk evrişimli katmanlar, görüntülerin düşük seviye özelliklerini (örneğin, kenarlar) yakalarken sonraki katmanlar yüksek seviye özellikleri (örneğin, şekiller ve belirli nesneler) çıkarırlar (Mostafa & Wu, 2021). Şekil 15’de motifler, nesne parçaları ve nesnelerin ilişkisi gösterilmiştir.



Şekil : CNN'de farklı katmanlarda meydana gelen nesne temsilleri (İnik & Ülker, 2017)

Kernel, görüntü üzerinde soldan sağa ve yukarıdan aşağı doğru görüntü boyunca hareket etmektedir. Kesikli cross-correlation operasyonunun 3 kanallı Red Green Blue (RGB) görüntüsü için matematiksel gösterimine Denklem(***13***)’de değinilmiştir (Raitoharju, 2022):

|  |  |  |
| --- | --- | --- |
|  |  | **(****13)** |

Burada C kanal sayısına karşılık gelirken s’inci kanaldaki Kernel ağırlığını, matrisi girdi matrisini veya 0 dolgulanmış halini, ise ortaya çıkan özellik matrisini temsil etmektedir.

Normal filtre sonucunda çıktı matrisinin girdi matrisinden daha küçük boyutta olması beklenir. Padding parametresi, girdi matrisinin etrafı pixeller ile doldurarak çıktı matrisinin boyutunu korumayı sağlar. Doldurulan pixelleri belirlemekte kullanılan bilinen iki adet padding tekniği vardır: Sıfır(zero) ve yansıma (reflection). Zero padding çevreyi 0 pixelleri ile doldurarak keskin bir geçiş yaratır. Reflection padding tekniği ise kenardaki pixellerin yansımaları eklenerek yumuşak geçiş uygular.

Bir diğer parametre ise adım (stride) sayısıdır. Filtreler, girdi olarak verilen görüntü boyunca belirlenmiş adım sayısı kadar kaydırılarak gezdirilir ve böylece öznitelik matrisi (activation map) ede edilir.

|  |  |  |
| --- | --- | --- |
|  |  | **(****14)** |

|  |  |  |
| --- | --- | --- |
|  |  | **(****15)** |

Çıktı matrisinin yüksekliği ve genişliği sırayla Denklem ***(14)*** ve Denklem ***(15)***’de verilmiştir. Girdi matrisinin yüksekliği H, genişliği W ve filtre matrisinin büyüklüğü F, stride değeri S, dolgulama değeri ile P temsil edilmektedir.

Kernel, görüntü matrisi üzerinde gezerken görüntü matrisinin izdüşümündeki matrise alıcı alan (receptive field) denir. Alıcı alan ve kernelin boyutu aynıdır. Bu iki matrisin noktasal çarpımı sonucunda öznitelik matrisinin bir elemanının değeri elde edilir.

|  |  |  |
| --- | --- | --- |
|  |  | **(****16)** |

|  |  |  |
| --- | --- | --- |
|  |  | **(****17)** |

Denklem ***(16)***’deki A ve Denklem ***(17)***’deki B, 3×3 boyutunda matrisler olsun. Noktasal çarpımları Denklem ***(18)***’de gösterilmiştir:

|  |  |  |
| --- | --- | --- |
|  |  | **(****18)** |

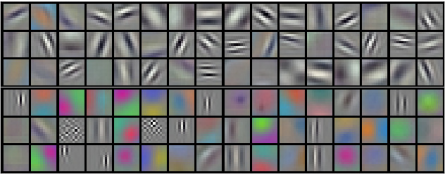
Şekil 16’de görüldüğü üzere 3×3 boyutundaki Kernel, 8×8 boyutundaki girdi matrisi üzerinde gezerek alıcı alan ile eleman bazlı çarpım uygulanmıştır ve 1x1 + 1x1 + 1x0 + 1x0 + 0x1 + 0x0 + 1x1 + 0x0 + 0x1 = 3 sonucu elde edilmiştir. Tüm bu işlemlerin sonucunda 8 – 3 + 1 = 6, 6×6 boyutunda öznitelik matrisi oluşmuştur.

diyagram içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil : Receptive field ile Kernel'in nokta çarpımı (Singh, Meitei, & Majumder, 2020)

Örneğin bir konvolüsyon katmanında k adet filtre kullanılırsa aynı katmanda çıktı olarak k adet bağımsız öznitelik matrisi oluşmaktadır. İlk konvolüsyon katmanında 224×224×3 boyutundaki girdi görüntülerinden elde edilen 96 adet 11×11×3 boyutundaki konvolüsyon kernel’leri Şekil 17’de gösterilmiştir (Krizhevsky, Sutskever, & Hinton, 2017).



Şekil : Örnek 96 Adet Konvolüsyon Kernel’i (Krizhevsky, Sutskever, & Hinton, 2017)

#### Toplu Normalizasyon (Batch Normalization) Katmanı

Bilindiği üzere, CNN ve diğer derin öğrenme mimarilerinin başarılı bir şekilde eğitebilmesi için çok miktarda veriye ihtiyaç duyulmaktadır. Bu modellere resimlerin tek tek sokulduğu düşünülürse her bir resim için ağırlıkların güncellenmesi öğrenme sürecini oldukça fazla uzatacaktır.

Geriye doğru yönelim yöntemiyle çıktıdan girdiye doğru bir yol izlerken katmanlarda bulunan ağırlıklar katman katman güncellenir. Yani bir katmandaki ağırlıkların güncellenmesi için önce diğer katmandaki ağırlıkların güncellenmiş olması gerekmektedir. Bu durum bazen Derin Öğrenme’de yaygın olarak ortaya çıkan İçsel Kovaryans Kayması’na (Internal Covariate Shift) neden olabilmektedir (Brownlee, 2019). Kullanılması opsiyonel olan Batch Normalization katmanı ise veriyi gruplar halinde normalleştirme yaparak eğitimi hızlandırır (Keskin, 2022). Internal Covariate Shift olgusunun önüne geçer ve eğitim döngüsü sayısını da (epoch) dramatik bir şekilde azaltır (Brownlee, 2019).

#### Rektifiye Edilmiş Doğrusal Birim Katmanı (ReLU: Rectified Linear Units Layer)

Rektifiye Edilmiş Doğrusal Birim, bir aktivasyon fonksiyonu çeşididir. ReLU katmanı, evrişim katmanında gerçekleşen matematiksel işlemlerin ardından doğrusallaşan ağı doğrusal olmayan ağa dönüştürür ve aynı zamanda ağın hızını arttırır (İnik & Ülker, 2017). ReLU aktivasyon katmanı, ilk defa bir CNN mimarisi olan AlexNet’te kullanılmıştır (Abdelmalek, Ahmed, & Amine, 2019). Denklem ***(19)***’de ReLU aktivasyon fonksiyonunun matematiksel denklemi verilmiştir.

|  |  |  |
| --- | --- | --- |
|  |  | **(****19)** |

Şekil 18’de, sol tarafta 4×4 boyutunda bir girdi matrisi, orta kısımda ReLU fonksiyonunun koordinat düzlemindeki grafiği ve sağ tarafta ise fonksiyon sonunda oluşan çıktı matrisi verilmiştir. Negatif değerlerin 0 değerini aldığı gözlenmektedir.

diyagram içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil : 4×4 boyutundaki matrisin ReLU fonksiyonuna girmeden önceki ve girdikten sonraki durumu (Taşhan, 2017)

ReLU fonksiyonunun konvolüsyon katmanından gelen görüntü verisi üzerindeki etkisi Şekil 19’de gösterilmiştir.

web sitesi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil : Giriş görüntüsünün sırasıyla konvolüsyon ve ReLU katmanlarından geçtikten sonraki çıktı görüntüleri (İnik & Ülker, 2017)

Bazen bu katman Konvolüsyon katmanının içinde sayıldığından nihai çıktının eşitliği Denklem ***(20)***’de ifade edilmektedir. Daha düzgün ifadeyle, Denklem ***(20)***’de uygulanan \* operasyonu sonrası elde edilen değerine yanlılık değeri eklenip aktivasyon fonksiyonuna sokulduğunda öznitelik matrisi elde etmiş oluruz.

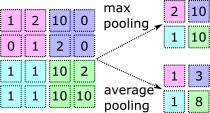
|  |  |  |
| --- | --- | --- |
|  |  | **(****20)** |

#### Ortaklama Katmanı (Pooling Layer)

Ortaklama katmanı, verinin boyutunu düşürürken verideki özellik kaybının mümkün olduğunca az olmasını sağlar. Literatürde max pooling, average pooling ve minimum pooling gibi birçok pooling çeşidi bulunsa da CNN algoritmalarında en yaygın kullanılanı max pooling metodu olmuştur.

Max pooling işleminde bazı özelliklerin silinmesine karşın iyi eşleşmiş özelliklerin bilgisi korunmaktadır ve ayrıca görüntü boyutunu düşürmesi ağın hesaplama süresini oldukça azaltmaktadır (İnik & Ülker, 2017). Konvolüsyonel sinir ağları, pooling katmanları sayesinde translation invariant özelliği gösterir. Translation invariant özelliği, girdi üzerinde dönüşüm işlemleri (kaydırma, döndürme vb.) uygulansa dahi bu değişikliklere karşı direnç gösterilerek özelliğin yine de tespit edilebilmesidir (Goodfellow, Bengio, & Courville, 2016).

Ortaklama katmanına giren görüntünün yüksekliği ve genişliği konvolüsyon katmanındakine benzer şekilde Denklem ***(14)*** ve Denklem ***(15)***’ye göre değişmektedir. Şekil 20’de pooling katmanına giren özellik matrisi üzerinde 2×2 boyutlu stride değeri 2 olan max pooling ve average pooling işlemlerinin uygulanışı gösterilmiştir.



Şekil : Stride değeri 2 iken 2×2 boyutlu max pooling ve average pooling işlemleri (Raitoharju, 2022)

Girdi matrisi 4×4 boyutunda olup filtre matrisi ise 2×2 boyutundadır. Burada stride değeri 2 olduğundan her iki pooling işlemi sonucunda meydana gelen matrisin yükseklik ve derinliği sırasıyla Denklem ***(14)*** ve Denklem ***(15)***’ye göre 2×2 olmaktadır.

ReLU katmanından çıkan görüntünün havuzlama katmanından geçmesi sonucunda oluşan görüntüsü Şekil 21‘de gösterilmiştir. Örneğin, soldaki görüntüde yollar bembeyaz gözükürken sağdaki görüntüde yolların daha belirginleştiği anlaşılmaktadır.

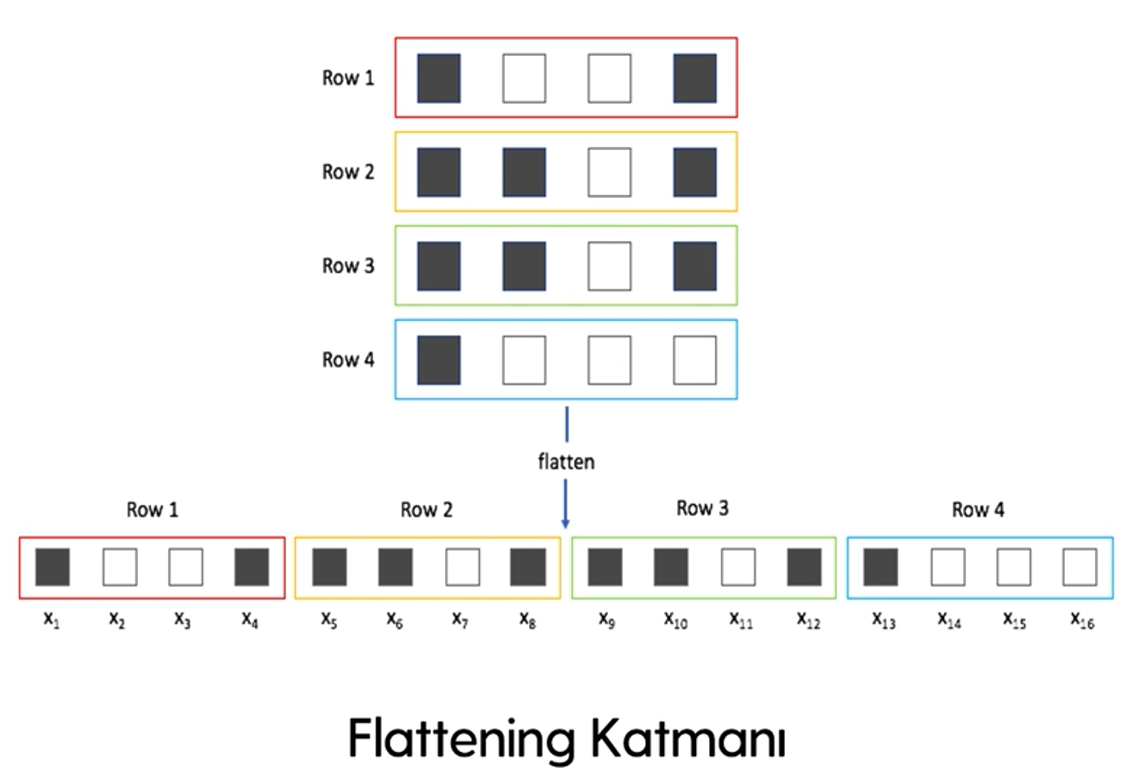
metin, ekran, görüntüleme, hava taşıtı, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil : ReLU katmanından çıkan görüntünün Pooling katmanında işlem gördükten sonraki çıkan görüntüsü (İnik & Ülker, 2017)

#### Düzleştirme Katmanı (Flattening)

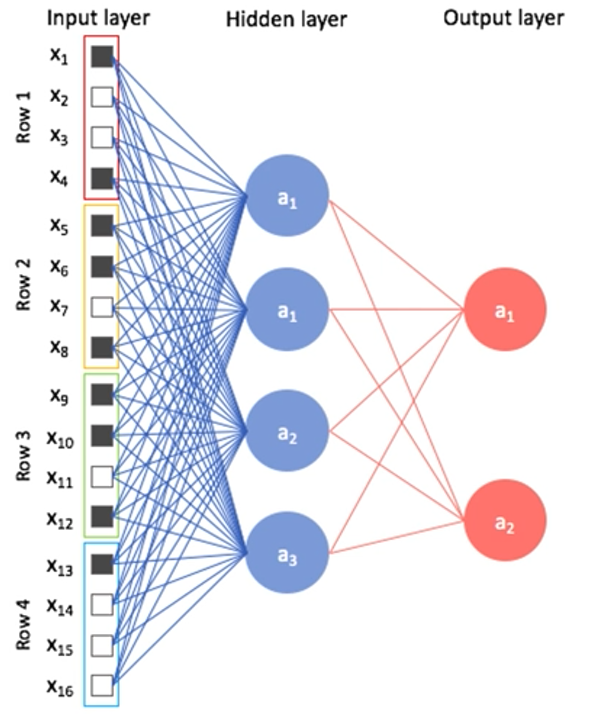
Önceki katmanlarda işlemlerden geçerek son hale gelen çok boyutlu veri, tam bağlı katmana girmeden önce bu katmanda tek boyuta yani vektöre indirgenir. Bunun sebebi tam bağlı katmandaki sinir ağlarının girdi değerlerini vektör olarak kabul etmesidir (Keskin, 2022). Şekil 22’de flattening katmanında örnek bir 4×4 boyutuna sahip matrisin 1x8 boyutundaki vektöre indirgendiği ifade edilmektedir.



Şekil : Matrisin Düzleştirme (Flattening) Katmanı ile vektör haline getirilmesi (Keskin, 2022)

#### Tam Bağlı Katman

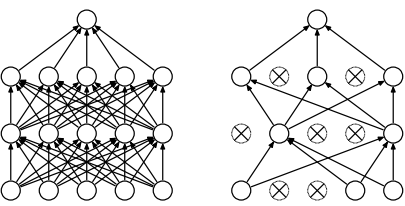
Tam Bağlı (Fully Connected) katmanlarında ise daha önceden tek boyuta indirgenmiş veri burada sinir ağları yardımıyla sınıflandırma işlemine tabii tutulur. ANN’lerdeki işlemler sonucunda sınıflandırmak üzere her sınıf için skor üretilir. Sınıf sayısı ile çıktı sayısı aynı miktardadır (Keskin, 2022). Şekil 23’de örnek basit bir tam bağlı katmanın yapısı gösterilmektedir.



Şekil : Basit bir Tam Bağlı Katman yapısı (Keskin, 2022)

#### Sönümleme (Dropout) Katmanı

Yapay sinir ağları, veriyi çok fazla öğrenmesi sebebiyle yaşanan aşırı öğrenme sorununu Dropout katmanı azaltabilmektedir. Tamamen bağlı sinir ağlarını oluşturan nöronların belli oranda rastgele sönümlenmesi ile bilinçli olarak meydana gelen hata rastgelelik sağlar. Böylece aşırı öğrenmenin önüne geçilir ve model başarısı artmış olur (Keskin, 2022). Şekil 24’de Dropout katmanının CNN üzerindeki etkisi gösterilmiştir.



Şekil : Solda 2 gizli katmandan oluşan bir sinir ağı ve sağda Dropout katmanının uygulanmasından sonra oluşan ağ (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014)

#### Sınıflandırma Katmanı

Çok katmanlı ANN’lerde tahmin skorunun olasılık dağılımı elde edilen katmandır. Tek etiketli çoklu sınıflandırma problemlerinde Softmax genellikle tercih edilen popüler bir aktivasyon fonksiyonudur. Denklem ***(21)***’de Softmax fonksiyonu verilmiştir:

|  |  |  |
| --- | --- | --- |
|  |  | **(****21)** |

Burada nöronlardan çıktı olarak elde edilen vektörün i. elemanını, C ise çıktı sınıf sayısını temsil etmektedir. Tüm elemanlar 0 ile 1 arasında değerlenir ve tüm elemanların toplamı 1 değerini verir yani bir anlamda her sınıfın olasılıkları elde edilmiş olur (Raitoharju, 2022).

#### Kayıp Fonksiyonu

CNN’ler bir çok amaçla kullanılabilmektedir ve mutlaka ilgili amaca uygun bir kayıp fonksiyonu seçilmelidir. Regresyon problemlerinde MSE tercih edilirken ikili veya çoklu sınıflandırma problemlerinde Binary Cross Entropy kullanılır. Denklem ***(22)***’de Binary Cross Entropy eşitliği verilmiştir.

|  |  |  |
| --- | --- | --- |
|  |  | **(****22)** |

Burada ikili one-hot encoding vektörünü, N ise sınıf sayısını temsil etmektedir (Raitoharju, 2022).

### Konvolüsyonel Sinir Ağları’nda Yaygın Karşılaşılan Zorluklar

CNN'lerde yaşanan zorluklar başlıca overfitting, milyonlarca parametreyle eğitilmesi nedeniyle eğitim süresinin uzunluğu, eğitim süresinin uzunluğu sonucunda güncelleme kilitlemesi (update locking), küçük gradyanların üstel olarak azalmasıyla 0’a yaklaşarak gradyanların kaybolması veya büyük gradyanların üstel olarak artmasıyla kararsız hale gelerek gradyanların patlaması, ve İçsel Kovaryans Kayması olarak özetlenebilir. Bu tür sıkıntıların giderilerek modelin başarısının arttırılması amacıyla sıkça başvurulan aşağıdaki teknikler kullanılır ve bu teknikler birden fazla soruna çözüm olabilmektedir (Raitoharju, 2022):

* Veri Artırımı (Data Augmentation) kullanılarak çeşitlilik yaratılması,
* Dropout tekniği kullanılarak ağda rastgelelik sağlanması,
* Aktivasyon fonksiyonu olarak ReLU kullanılması,
* Transfer Learning yönteminden yararlanarak daha önceden benzer ve büyük veri setleri ile eğitilmiş CNN katmanlarının kullanımı,
* Ağırlık Düzenlileştirilmesiyle modelin karmaşıklığının azaltılması,
* Normalizasyon yöntemiyle alt katmanlarda gerçekleşen parametre güncellenmesinin sonraki katmanlarda yaşanan girdi dağılımının değişiminin önlenmesi.

# Kaynakça

Abdelmalek, B., Ahmed, K., & Amine, T. M. (2019). A Survey on Lightweight CNN-Based Object Detection Algorithms for Platforms with Limited Computational Resources. *International Journal of Informatics and Applied Mathematics*, 28-44.

Agatonovic-Kustrin, S., & Beresford, R. (2000). Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research. *Journal of Pharmaceutical and Biomedical Analysis*, 717-727.

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, 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*, 1-74.

Anding, K., Haar, L., Polte, G., Walz, J., & Notni, G. (2019). Comparison of the performance of innovative deep learning and classical methods of machine learning to solve industrial recognition tasks. *Joint TC1 - TC2 International Symposium on Photonics and Education in Measurement Science 2019.* Jena: Spie.

*Artificial Neural Network Tutorial*. (tarih yok). Javatpoint: https://www.javatpoint.com/artificial-neural-network adresinden alındı

Aslan, Z. (2018). ON THE USE OF DEEP LEARNING METHODS ON MEDICAL IMAGES. *The International Journal of Energy & Engineering Sciences*, 1-15.

Barla, N. (2023, Nisan 25). *Self-Driving Cars With Convolutional Neural Networks (CNN)*. Neptune: https://neptune.ai/blog/self-driving-cars-with-convolutional-neural-networks-cnn adresinden alındı

Brownlee, J. (2019, January 16). *A Gentle Introduction to Batch Normalization for Deep Neural Networks*. Machine Learning Mastery: https://machinelearningmastery.com/batch-normalization-for-training-of-deep-neural-networks/ adresinden alındı

Carpenter, K. A., Cohen, D. S., Jarrell, J., & Huang, X. (2018). Deep learning and virtual drug screening. *Future Medical Chemistry*, 2557–2567.

Cayla, B. (2021, Mart 7). *The Stochastic Gradient Descent (SGD) & Learning Rate*. Aishelf: http://aishelf.org/sgd-learning-rate/ adresinden alındı

Das, S., Dey, A., Pal, A., & Roy, N. (2015). Applications of Artificial Intelligence in Machine Learning: Review and Prospect. *International Journal of Computer Applications*, 31-41.

Doğan, Ö. (2020, Kasım 26). *CNN (Convolutional Neural Networks) Nedir?* Teknoloji: https://teknoloji.org/cnn-convolutional-neural-networks-nedir/ adresinden alındı

Dongare, A., Kharde, R., & Kachare, A. D. (2012). Introduction to Artificial Neural Network. *International Journal of Engineering and Innovative Technology*, 189-194.

Forjan, J. (2021, Mart 6). *Overfitting and Methods of Addressing it*. AnalystPrep: https://analystprep.com/study-notes/cfa-level-2/quantitative-method/overfitting-methods-addressing/ adresinden alındı

French, R. M. (2000). The Turing Test: the first 50 years. *Trends in Cognitive Sciences*, 115-122.

Gandharv, K. (2022, Haziran 29). *Top 5 applications of Convolution Neural Network*. Indiaai: https://indiaai.gov.in/article/top-5-applications-of-convolution-neural-network adresinden alındı

Gardner, M. W., & Dorling, S. R. (1998). Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric Environment*, 2627-2636.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning.* MIT Press.

Güzel, K. (2018, 7 20). *Geri Yayılımlı Çok Katmanlı Yapay Sinir Ağları-2*. Medium: https://kadirguzel.medium.com/geri-yayılımlı-çok-katmanlı-yapay-sinir-ağları-2-6a47b4f3a6c adresinden alındı

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.

Høye, T. T., Ärje, J., Bjerge, K., Hansen, O. L., Iosifidis, A., Leese, F., Mann, H. M. R., Meissner, K., Melvad, C., Raitoharju, J. (2021). Deep learning and computer vision will transform entomology. *Proceedings of the National Academy of Sciences*, 1-10. doi:https://doi.org/10.1073/pnas.2002545117

İnik, Ö., & Ülker, E. (2017). Derin Öğrenme ve Görüntü Analizinde Kullanılan Derin Öğrenme Modelleri. *Gaziosmanpaşa Bilimsel Araştırma Dergisi*, 85-104.

Karimi, G. (2021, Nisan 15). *Introduction to YOLO Algorithm for Object Detection*. Section: https://www.section.io/engineering-education/introduction-to-yolo-algorithm-for-object-detection/ adresinden alındı

Kattenborn, T., Leitloff, J., Schiefer, F., & Hinz, S. (2021). Review on Convolutional Neural Networks (CNN) in vegetation remote sensing. *ISPRS Journal of Photogrammetry and Remote Sensing*, 24-49. doi:https://doi.org/10.1016/j.isprsjprs.2020.12.010

Kayalı, N. Z., & Omurca, S. İ. (2021). Konvolüsyonel Sinir Ağları (CNN) ile Çin Sayı Örüntülerinin Sınıflandırılması. *Journal of Computer Science*, 184-191. doi:https://doi.org/10.53070/bbd.989668

Keskin, M. V. (2022, Eylül 23). *Evrişimli Sinir Ağları*. Geleceği Yazanlar: https://gelecegiyazanlar.turkcell.com.tr/konu/egitim/derin-ogrenme-cnn/evrisimli-sinir-aglari adresinden alındı

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet Classification with Deep Convolutional Neural Networks. *Communications of the ACM*, 84-90.

Lecun, Y. (1989). Backpropagation applied to handwritten zip code. *Neural computation*, 541-551.

Lee, D.-H., Kim, Y.-T., & Lee, S.-R. (2020). Shallow Landslide Susceptibility Models Based on Artificial Neural Networks Considering the Factor Selection Method and Various Non-Linear Activation Functions. *Remote Sensing*, 1-28.

Marsalli, M. (tarih yok). *McCulloch-Pitts Neurons*. The Mindproject: https://mind.ilstu.edu/curriculum/mcp\_neurons/index.html adresinden alındı

Mostafa, S., & Wu, F.-X. (2021). Chapter 3 - Diagnosis of autism spectrum disorder with convolutional autoencoder and structural MRI images. *Neural Engineering Techniques for Autism Spectrum Disorder*, 23-38.

Noriega, L. (2005). Multilayer Perceptron Tutorial. 1-12.

Raitoharju, J. (2022). Chapter 3 - Convolutional neural networks. *Deep Learning for Robot Perception and Cognition*, 35-69.

Rosebrock, A. (2021, Mayıs 14). *Convolution and cross-correlation in neural networks*. Pyimagesearch: https://pyimagesearch.com/2021/05/14/convolution-and-cross-correlation-in-neural-networks/ adresinden alındı

Rosenblatt, F. (2017). *The Perceptron A Perceiving and Recognizing Automation.* Buffalo: Cornell Aeronautical Laboratory, Inc.

Ruder, S. (2017). An overview of gradient descent optimization algorithms. *Arxiv*, 1-14.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. 1-43. doi:https://doi.org/10.48550/arXiv.1409.0575

Sharma, P. (2020, Temmuz 27). *7 Popular Image Classification Models in ImageNet Challenge (ILSVRC) Competition History*. Machine Learning Knowledge: https://machinelearningknowledge.ai/popular-image-classification-models-in-imagenet-challenge-ilsvrc-competition-history/ adresinden alındı

Singh, S. A., Meitei, T. G., & Majumder, S. (2020). Short PCG classification based on deep learning. *Deep Learning Techniques for Biomedical and Health Informatics*, 141-164.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. (Y. Bengio, Dü.) *Journal of Machine Learning Research, 15*, 1929-1958.

Sun, S., Cao, Z., Zhu, H., & Zhao, J. (2019). A Survey of Optimization Methods from a Machine Learning Perspective. *Arxiv*, 1-30.

Toda, Y., & Okura, F. (2019). How Convolutional Neural Networks Diagnose Plant Disease. *Plant Phenomics*, 1-14. doi:https://doi.org/10.34133/2019/9237136

Wanga, P., Fan, E., & Wang, P. (2021). Comparative analysis of image classification algorithms based on traditional machine learning and deep learning. *Pattern Recognition Letters*, 61-67. doi:https://doi.org/10.1016/j.patrec.2020.07.042

Taşhan, B. (2017) Road lane detection system with convolutional neural network (Tez No. 467573) [Yüksek lisans tezi, Bahçeşehir Üniversitesi]. YÖK Tez Merkezi. https://tez.yok.gov.tr/UlusalTezMerkezi/tezDetay.jsp?id=r\_HEN\_uwwz9qfeOkPFkiZw&no=1ECK3\_7sRvgkMOExfHr7cw