

Derin Öğrenme – Deep Learning

Dr. Fatih KALEMKUŞ

Kafkas Üniversitesi

Yazılımcı Ne Yapar?

- Algoritma ve matematiksel model geliştirir.
- Değişkenlere bakarak yorum yapacak, öngörude bulanacak, karar verecek algoritmalar ve matematiksel modeller geliştirir.
- Matematik değil aritmetiği çok iyi bilir.
- Aritmetik: Toplama, çıkarma, çarpma, bölme; logaritma, kompleks sayı sistemi, üssel işlemler, oran, yüzde, türev,...

Yazılımcı Ne Yapar?

ML
Veri için
- Karar Verme
- Yorum
- Analiz
- Öngörme.

ML
Sınıflandırma $f(t)$, $\{f_1(t), f_2(t), \dots\}$
Regrasyon - Matematiksel ifade ya da dünden
Kümeleme - Katsayılarının belirlenmesi.
Benzerlerin özelliklerinin ayrıştırılması.
(Vücut-organları ile)

DL
1- Uyumsuz kalıp → Anomali
2- Filtreleme → Sapma
3- Ağırlık katsayılarının {Araştırılması} → Gövde. (Gürültüler, hatadan, eksik anomali değerlerinde baskın.)
ANN- Yapay Sinir Ağları.

Mantıksal devre tasarımı
OR, AND, NOT, XOR, XNOR
Temel aritmetik, binary numbering system, bit: %
Aritmetik:
(Toplama) $\begin{array}{r} 0 \\ +0 \\ \hline 0 \end{array}$ $\begin{array}{r} 0 \\ +1 \\ \hline 1 \end{array}$ $\begin{array}{r} 1 \\ +0 \\ \hline 1 \end{array}$ $\begin{array}{r} 1 \\ +1 \\ \hline 10 \end{array}$
Çarpma/Bölme → bitleri $\begin{array}{r} 1 \\ \times 1 \\ \hline 1 \end{array}$ $\begin{array}{r} 1 \\ \times 0 \\ \hline 0 \end{array}$
Sola Çarpım. Sağa-Bölme.

Mantık Devre: OR, AND, NOT, XOR, XNOR
NOT NAND
Sonuç: 0/1
AND: $0 \times 0 = 0$, $0 \times 1 = 0$, $1 \times 0 = 0$, $1 \times 1 = 1$
NOT: $0 \rightarrow 1$, $1 \rightarrow 0$
OR: $0 + 0 = 0$, $0 + 1 = 1$, $1 + 0 = 1$, $1 + 1 = 1$

⇒ DL.
↓
Durum diyagramları
Network.

*f(t): bağımlı.
t: bağımsız*

Derin Öğrenme

Derin öğrenme modeli, verinin yapısına göre uzamsal davranış kalıbını belirlenmesi ya da hangi parametrelere ne ağırlık verileceğini kendisinin keşfetmesidir. Verilecek ağırlıklar filtrelenir, değerler belirli aralıklarda sınırlanır ya da zarflanır (Genişleyip daralan yollara).

Derin öğrenme yöntemi insan beyninin öğrenme ve karar verme yöntemlerinden esinlenerek geliştiriliyor, yani nöronların hareketleri taklit ediliyor.

Bu da "Yapay Sinir Ağları" (Artificial Neural Networks) sayesinde oluyor.

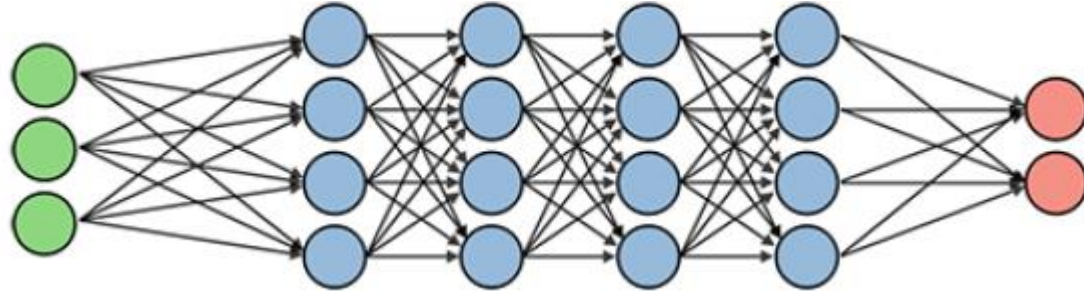
Derin öğrenme (deep learning) makine öğrenmesinin alt kümesidir.

Derin Öğrenme Nasıl Çalışır

- Bir derin öğrenme modeli, insanın sonuç çıkarma biçimine benzer bir mantık yapısıyla verileri sürekli olarak **analiz etmek** için tasarlanmıştır.
- Bu süreci gerçekleştirmek için **yapay sinir ağı (Artificial Neural Network)** adı verilen **katmanlı bir algoritma yapısı** kullanılır.
- Yapay sinir ağlarının tasarımı, **insan beynindeki biyolojik sinir ağlarından** esinlenmiştir.
- Bu ağlar, **standart makine öğrenimi modellerine göre çok daha gelişmiş bir öğrenme süreci** sağlar.
- **Makine öğrenimi** modellerinde genellikle **birkaç katman** bulunurken, **derin öğrenme** modellerinde **yüzlerce hatta binlerce katman** yer alabilir.
- Bu çok katmanlı yapı, **büyük miktarda veriyi işleyebilmek** için gereklidir.
- **Yeterli sayıda katman veya nöron** olmadığında, model **karmaşık sorunları çözmede yetersiz** kalabilir.

Derin Öğrenme

- Kullanılacak yöntem olarak mimari yapıda
 - etkinleştirme fonksiyonu,
 - geri yayılım, parametre
 - indirgeme ya da düşürme
- Evrişimsel katman, küme normalleştirilmesi, kapı çeşitleri
- Markov karar süreci, Bellman denklemi, Q-öğrenimi



Derin Öğrenmenin Temel Bileşenleri

- Katman (Layer) Yapısı: Girdi, Gizli ve Çıkış Katmanları
- Tensör (Tensor) Kavramı ve Veri Temsili
- Kayıp (Loss) ve Optimizasyon Fonksiyonları
- Öğrenme Oranı (Learning Rate) ve Aşırı Uyum (Overfitting)

Layers (Katmanlar)

Bir yapay sinir ağı (Artificial Neural Network), veriyi işlemek için birbirine bağlı katmanlardan (layers) oluşur. Bu katmanlar, bilgiyi girdiden sonuca doğru adım adım işler.

1. ● Girdi Katmanı (Input Layer)

- Modelin veriyi ilk aldığı katmandır.
- Her bir nöron, giriş verisindeki bir özelliği (feature) temsil eder.
- Girdi katmanı veriyi doğrudan işlemez, sadece bir sonraki katmana (gizli katmana) aktarır.
- Örnek:
 - Görüntü tanıma modelinde her piksel bir giriş nöronudur.
 - 28x28 boyutunda bir resim → 784 giriş nöronu.

Layers (Katmanlar)

2. ● Gizli Katman(lar) (Hidden Layers)

- Girdi katmanından gelen veriyi **işleyen** katmanlardır.
- “**Gizli**” denir çünkü bu katmanlardaki işlemler doğrudan kullanıcı tarafından görülmez.
- Her gizli katmandaki nöronlar:
 - Önceki katmandan gelen değerleri **ağırlıklar (weights)** ve **bias** ile çarpar,
 - Ardından **aktivasyon fonksiyonu** ile yeni çıktılar oluşturur.
- Derin öğrenme modelinin gücü, genellikle **gizli katman sayısı ve nöron yoğunluğuna** bağlıdır.
- **Örnek:**
 - Görüntüdeki kenarları, desenleri, şekilleri ve nesneleri tanımak gizli katmanlarda gerçekleşir.

Layers (Katmanlar)

3. ● Çıkış Katmanı (Output Layer)

- Modelin nihai tahmin veya sonucu ürettiği katmandır.
- Çıkış katmanının nöron sayısı, **çözölmek istenen probleme göre deęişir**:
 - **Sınıflandırma (classification)** → Her sınıf için 1 nöron.
 - **Regresyon (sürekli deęer tahmini)** → Tek nöron.
- **Aktivasyon fonksiyonu**, çıktı türüne göre seçilir:
 - Sınıflandırma için **Softmax veya Sigmoid**,
 - Regresyon için **Linear (Doęrusal)** fonksiyon kullanılır.

Layers (Katmanlar)

Katman Türü	Görevi	Örnek
Girdi Katmanı	Veriyi modele almak	Görsel pikseller, sensör verisi
Gizli Katman(lar)	Veriyi işlemek, özellikleri öğrenmek	Kenar, desen, nesne tespiti
Çıkış Katmanı	Sonucu üretmek	Sınıf tahmini (örneğin "kedi" veya "köpek")

Tensör Kavramı ve Veri Temsili

Tensör (Tensor) Kavramı

- Tensör, verilerin matematiksel olarak temsil edildiği çok boyutlu bir dizidir.
- Skaler, vektör ve matris kavramlarının **genelleştirilmiş hâli** olarak düşünülebilir.
- Derin öğrenme modellerinde, tüm veriler (görseller, metinler, sesler, vb.) **tensörler biçiminde temsil edilir.**

Tensör = Derin öğrenmede tüm verilerin matematiksel temsili.
Her şey (girdi, ağırlık, çıktı) tensör olarak ifade edilir.

Veri Temsili (Data Representation)

- Derin öğrenmede tüm veriler **sayısal forma dönüştürülerek** tensör hâline getirilir.
- Bu sayısal temsiller, modelin matematiksel hesaplamaları yapmasını sağlar.

Örnekler:

- Görseller → Piksel değerlerinden oluşan 3B tensörler (yükseklik × genişlik × renk),
- Metinler → Kelimeler veya karakterler sayısal vektörlerle temsil edilir,
- Ses verileri → Zaman serisi tensörleri olarak kaydedilir.

Tensör Kavramı ve Veri Temsili

Tensörlerin Boyut (Rank) Türleri

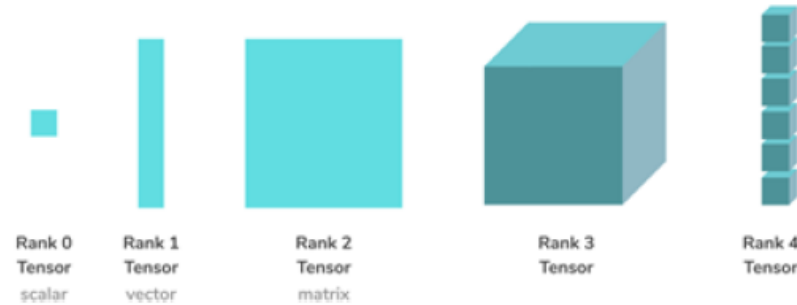
Tensör Türü	Açıklama	Örnek
Skaler (0-Boyutlu Tensör)	Tek bir sayıdır.	$x = 5$
Vektör (1-Boyutlu Tensör)	Sayılardan oluşan bir listedir.	$[3, 7, 2]$
Matris (2-Boyutlu Tensör)	Satır ve sütunlardan oluşur.	$\begin{bmatrix} 1, 2, 3, 4 \end{bmatrix}$
3-Boyutlu Tensör	Matrislerin bir araya gelmesidir. (örneğin, renkli bir görüntü)	RGB görüntü: $28 \times 28 \times 3$
n-Boyutlu Tensör	Daha karmaşık veri kümelerini temsil eder.	Videolar, metin dizileri, sensör verileri

Tensör Kavramı ve Veri Temsili

Tensörün Özellikleri

- Her tensörün **şekli (shape)** vardır → boyutlar ve her boyuttaki eleman sayısını gösterir.
 - Örnek: $(3, 4)$ → 3 satır, 4 sütunlu matris.
- Tüm elemanlar **aynı veri tipindedir** (örneğin `float32`, `int64`).
- **Boyut sayısı (rank)**, tensörün karmaşıklığını belirler.
- Tensörler, derin öğrenme modeline **girdi (input)** olarak verilir ve **çıktı (output)** olarak da üretilir.

A tensor is an N-dimensional array of data



Kayıp (Loss) ve Optimizasyon Fonksiyonları

- Kayıp fonksiyonu, modelin tahminlerinin doğruluğunu ölçen bir matematiksel ifadedir.
- Gerçek değer (ground truth) ile modelin tahmini arasındaki farkı sayısal olarak hesaplar.
- Amaç: Kayıp değerini minimize ederek modelin doğruluğunu artırmak.

Örnekler:

Problem Türü	Kayıp Fonksiyonu
Sınıflandırma (Classification)	Cross-Entropy Loss
Regresyon (Continuous Prediction)	Mean Squared Error (MSE), Mean Absolute Error (MAE)

- Optimizasyon fonksiyonu, kayıp fonksiyonunu en aza indirmek için modelin ağırlıklarını günceller.
- Modelin öğrendiği değerler (weights ve biases) optimizatör sayesinde iyileştirilir.

Yaygın Optimizatörler:

- **Stochastic Gradient Descent (SGD)**: Temel ve klasik optimizasyon yöntemi.
- **Adam (Adaptive Moment Estimation)**: Daha hızlı ve stabil öğrenme sağlar.
- **RMSprop**: Özellikle tekrarlayan sinir ağlarında (RNN) etkilidir.

Kayıp (Loss) ve Optimizasyon Fonksiyonları

Kayıp ve Optimizasyon İlişkisi

1. Model, bir **girdi** alır ve **tahmin** üretir.
2. Kayıp fonksiyonu, tahmin ile gerçek değer arasındaki farkı ölçer.
3. Optimizasyon fonksiyonu, **geri yayılım (backpropagation)** yöntemiyle ağırlıkları günceller.
4. Bu adımlar **tekrar tekrar** uygulanır ve kayıp değeri zamanla azalır.

Özet Görsel Mantık:

Girdi → Model → Tahmin → Kayıp Hesapla → Optimizatör ile Ağırlıkları Güncelle → Tekrar

Önemli Noktalar

- Kayıp fonksiyonu: "Ne kadar hata yaptık?" sorusuna yanıt verir.
- Optimizasyon fonksiyonu: "Hatanın azaltılması için ağırlıkları nasıl değiştirelim?" sorusuna yanıt verir.
- Bu ikisi birlikte, **derin öğrenme modelinin öğrenme sürecini yönlendirir.**

Öğrenme Oranı (Learning Rate) ve Aşırı Uyum (Overfitting)

Öğrenme Oranı (Learning Rate)

- Öğrenme oranı, **modelin ağırlıklarını güncellerken atacağı adım büyüklüğünü** belirler.
- Kısaca: "Ne kadar hızlı öğrenelim?" sorusuna cevap verir.
- **Özellikler:**
 - **Yüksek Öğrenme Oranı:**
 - Ağırlıklar çok hızlı güncellenir.
 - Model, optimum noktayı atlayabilir veya kararsız öğrenebilir.
 - **Düşük Öğrenme Oranı:**
 - Ağırlıklar yavaş güncellenir.
 - Öğrenme yavaş olur, uzun süre eğitim gerekebilir.
- **Amaç:**
 - Öğrenme oranını doğru seçmek, modelin **hızlı ve doğru şekilde öğrenmesini** sağlar.
 - Genellikle **0.001 – 0.01** aralığında başlamak yaygındır, ancak model ve veri tipine göre ayarlanır.

Öğrenme Oranı (Learning Rate) ve Aşırı Uyum (Overfitting)

Aşırı Uyum (Overfitting)

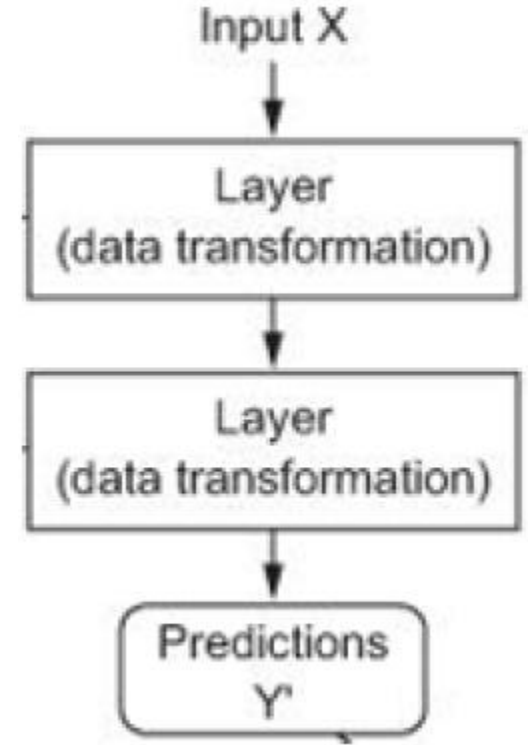
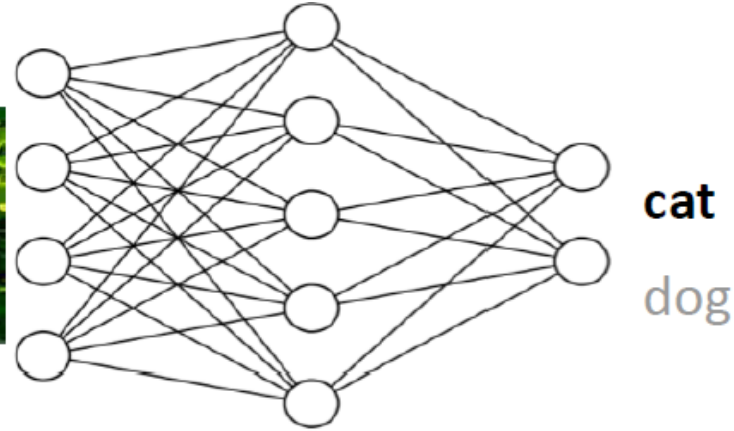
- Aşırı uyum, modelin eğitim verisine fazla odaklanması ve genelleme yeteneğini kaybetmesi durumudur.
- Sonuç: Model, eğitim verisinde çok iyi performans gösterirken yeni veya test verisinde başarısız olur.
- Nedenleri:
 - Model çok karmaşık (çok fazla katman veya nöron) olduğunda
 - Eğitim verisi çok az olduğunda
 - Eğitim çok uzun sürdüğünde
- Çözüm Yöntemleri:
 - Daha fazla veri toplamak
 - Dropout veya regularization gibi teknikler kullanmak
 - Modelin boyutunu veya karmaşıklığını azaltmak
 - Erken durdurma (Early Stopping) uygulamak

Öğrenme Oranı (Learning Rate) ve Aşırı Uyum (Overfitting)

Kavram	Tanım	Önem
Öğrenme Oranı (Learning Rate)	Model ağırlıklarını güncellerken adım büyüklüğü	Hızlı ve kararlı öğrenme için kritik
Aşırı Uyum (Overfitting)	Modelin eğitim verisine fazla bağlılığı	Genelleme yeteneğini düşürür, test verisinde başarısız olur

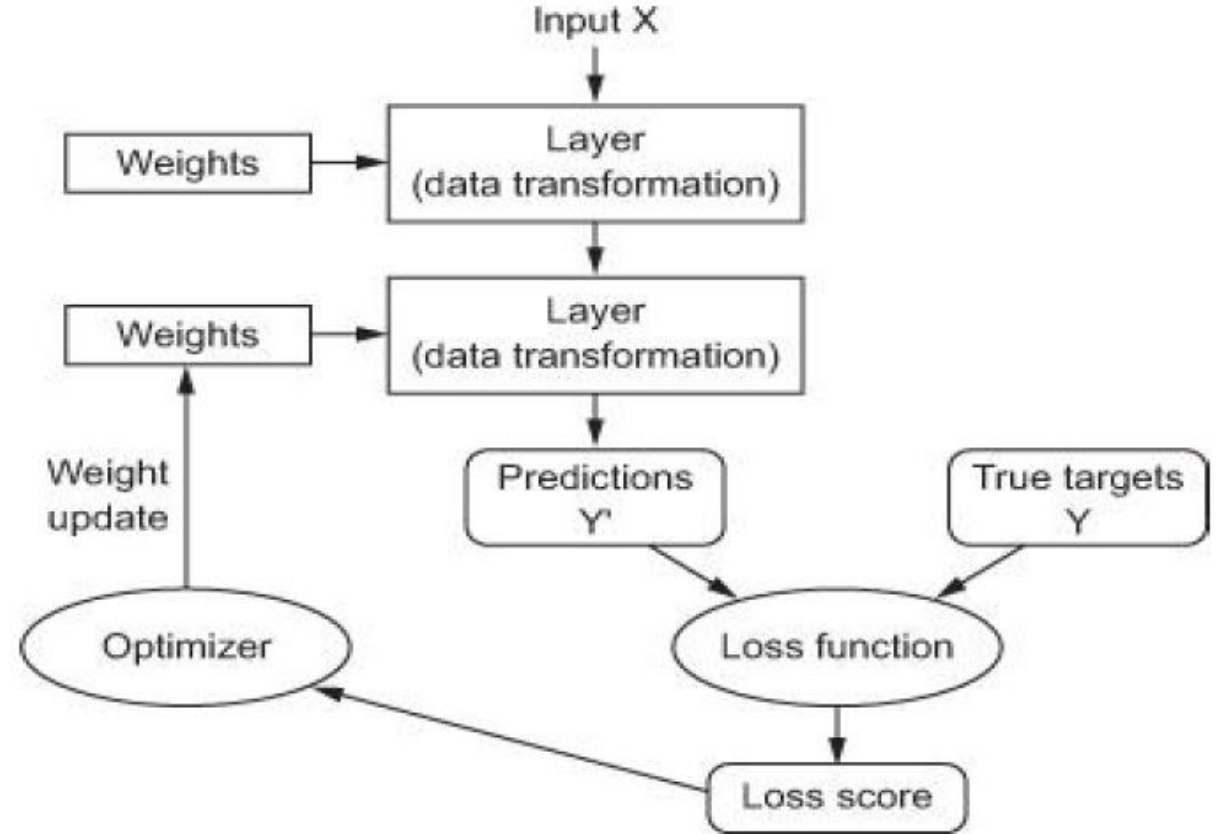
Veri Girişleri ve Hedefler

- Sinir Ağı, X giriş verisini Y' tahminleriyle eşleştirir
- Eğitim sırasında, Y' tahminleri, kayıp fonksiyonu kullanılarak gerçek Y hedefleri ile karşılaştırılır.
- Hatalı karar vermede geri dönülerek; katsayılar güncellenerek doğru karar verecek katsayılar belirlenmeye çalışılır.



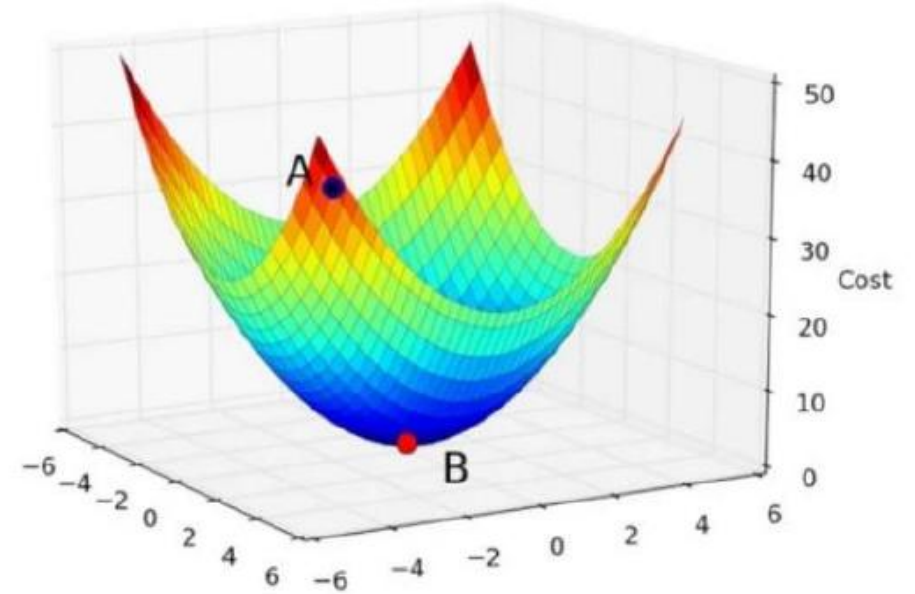
Derin Bir Sinir Ağının Anatomisi

- Layers
- Input data and targets
- Loss function
- Optimizer
- X: giriş dizisi
- W_i : ağırlık katsayıları



Gradyan İniş Optimize Edici

- Optimize edicilerin (En iyileyici: parametrede en iyi değeri bulmak) amacı, maliyet fonksiyonunun (Bellek işgali, veri işleme süresi, hızlı sonuca ulaşma adımları, hata oranlarının minimize edilmesi, sapma ya da kaotik davranış sergilememesi) mümkün olan en az değere ulaştığı küresel minimuma ulaşmaktır.



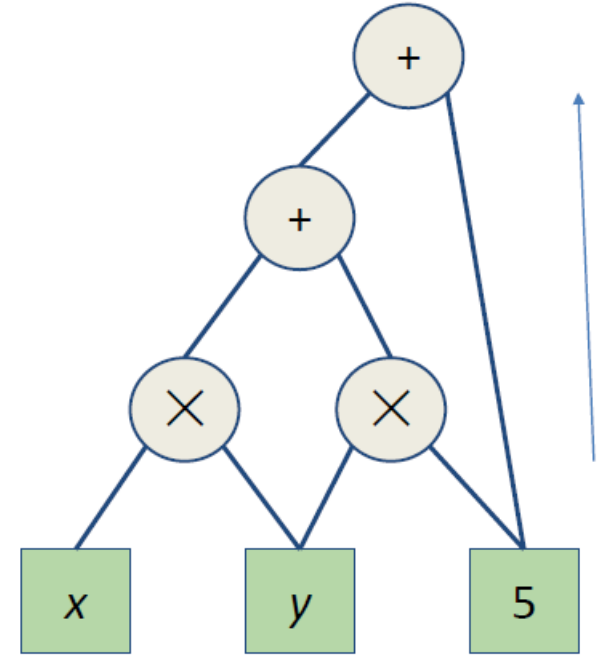
Stokastik Gradyan İniş Algoritması

Verilerdeki her örnek için;

- Siner ağı tarafından tahmin edilen deęerler bulunur.
- Kayıp fonksiyonunu deęeri hesaplanır.
- Kayıp fonksiyonunun kısmi türevlerini bulunur, bu kısmi türevler gradyanlar üretir.
- Ağırlık karsayılarının ve önyargıların(bias) deęerlerini güncellemek için gradyanlar kullanılır.

Derin Öğrenme Çerçevesi

- Aslında statik veya dinamik genel amaçlı hesaplamalı grafikleri tanımlamaya yönelik araçlar
- Otomatik farklılaşma
- GPU: Günümüzde yapay zeka uygulamalarını geliştirmek için üretilen grafiksel arayüzlere sahip mikroişlemcidir.
- Seamless CPU / GPU usage
 - multi-GPU, distributed
- Python/numpy or R interfaces
 - instead of C, C++, CUDA or HIP
- Open source



$$xy + 5y + 5$$

Derin Öğrenme ile Makine Öğrenimi Farkı

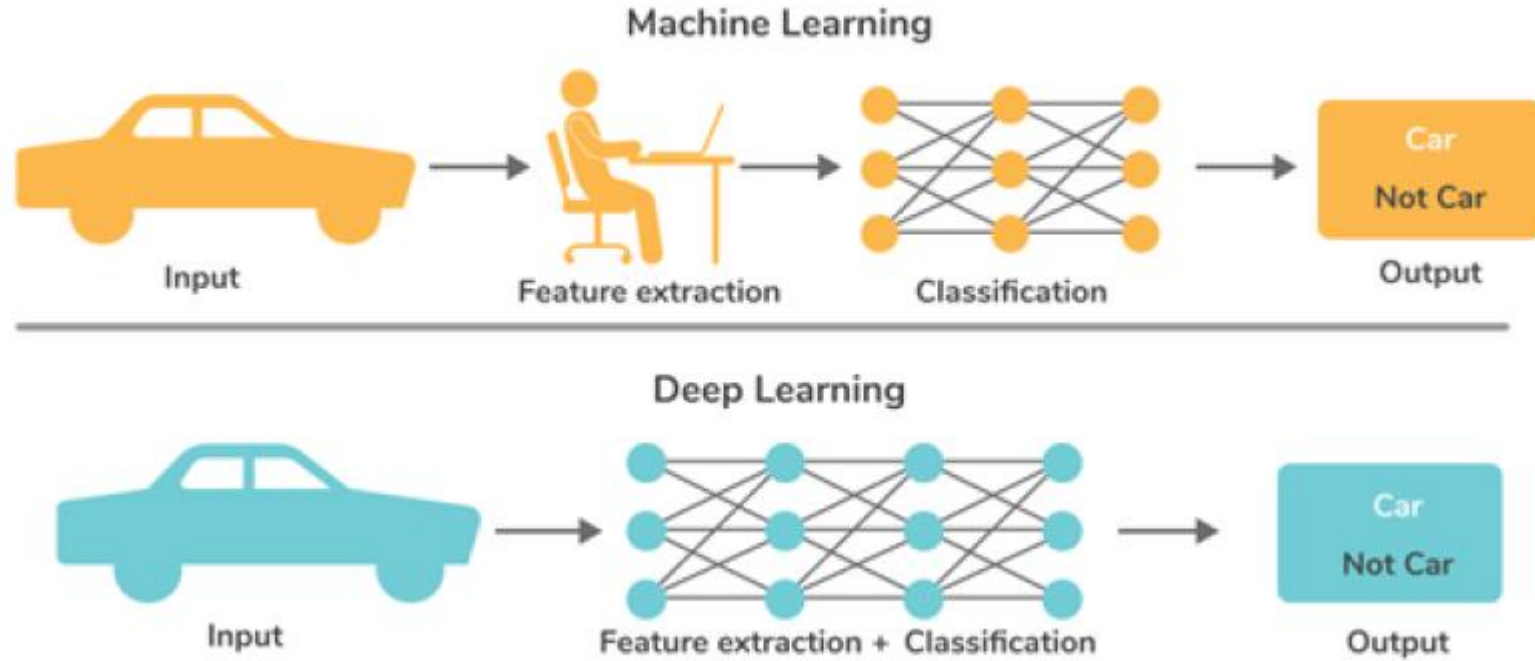
✿ Derin Öğrenme (Deep Learning)

- Derin Öğrenme, **Makine Öğrenimi'nin (Machine Learning)** bir alt dalıdır.
- **Yapay sinir ağları (Artificial Neural Networks)** ve **tekrarlayan sinir ağları (Recurrent Neural Networks)** üzerine kuruludur.
- Algoritmalar, makine öğreniminde olduğu gibi tasarlanır; ancak **daha fazla katmandan (derinlikten)** oluşur.
- Yapay sinir ağı, algoritmadaki tüm ağların bir araya gelmesini ifade eder.
- Derin öğrenme, **insan beynindeki sinir ağlarını taklit ederek** öğrenme sürecini simüle eder.
- **Karmaşık problemleri çözmek için gelişmiş algoritmalar ve teknikler** kullanır.

🤖 Makine Öğrenimi (Machine Learning)

- Makine öğrenimi, **Yapay Zeka'nın (AI)** bir alt kümesidir.
- Sistemlerin, **açıkça programlanmadan** deneyimlerinden öğrenmesini ve kendini geliştirmesini sağlar.
- **Verilerden öğrenir** ve doğru sonuçlar elde etmek için bu verileri kullanır.
- Algoritmalar, **daha fazla veri ile performanslarını artırabilir.**
- Günümüzde pek çok alanda kullanılmaktadır, örneğin:
 - Sürücüsüz araçlar,
 - Siber dolandırıcılık tespiti,
 - Yüz tanıma sistemleri,
 - Facebook arkadaş önerileri.

Derin Öğrenme ile Makine Öğrenimi Farkı

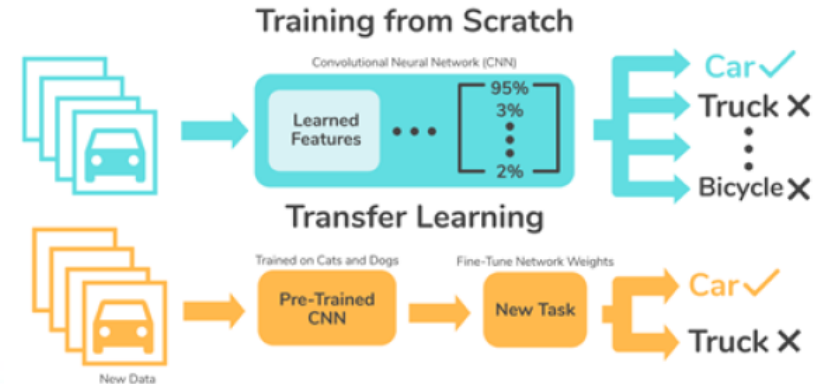


Derin Öğrenme Uygulamaları

- Örüntü tanıma ve doğal dil işleme.
- Görüntülerin tanınması ve işlenmesi.
- Otomatik çeviri.
- Duygu analizi.
- Soruları yanıtlamak için sistem.
- Nesnelerin Sınıflandırılması ve Tespiti.
- Makineye Göre El Yazısı Oluşturma.
- Otomatik metin oluşturma.
- Siyah Beyaz görüntülerin renklendirilmesi.

Derin Öğrenme Bağlamında Transfer Öğrenme

- Aktarım öğrenimi, veri bilimcilerinin daha önce benzer bir görev için eğitilmiş bir makine öğrenimi modelinden öğrendiklerini kullanmalarına olanak tanıyan bir tekniktir.
- Bu yaklaşım, insanların bilgilerini yeni durumlara aktarma yeteneğinden esinlenmiştir.
- Örneğin, bisiklete binmeyi öğrenen biri, başka iki tekerlekli araçları (örneğin motosiklet veya scooter) kullanmayı daha kolay öğrenir.
- Benzer şekilde, otonom otomobil sürüşü için eğitilmiş bir model, otonom kamyon sürüşü için de temel alınabilir.
- Bu durumda, modelin özellikleri (features) ve ağırlıkları (weights) yeniden kullanılabilir.
- Aktarım öğrenimi, veri miktarının sınırlı olduğu durumlarda yeni bir modeli hızlı ve etkili bir şekilde eğitmeye yardımcı olur.
- Bu sayede, zaman ve kaynak tasarrufu sağlanır.



Derin Öğrenmede Türleri

1. Denetimli Öğrenme (Supervised Learning)

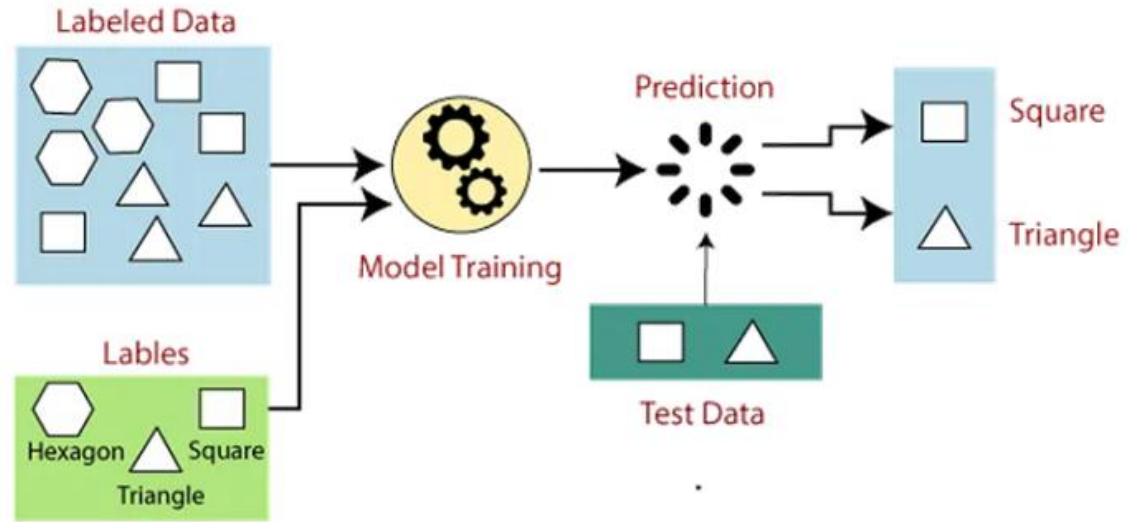
2. Denetimsiz Öğrenme (Unsupervised Learning)

3. Yarı Denetimli Öğrenme (Semi-Supervised Learning)

4. Pekiştirmeli Öğrenme (Reinforcement Learning)

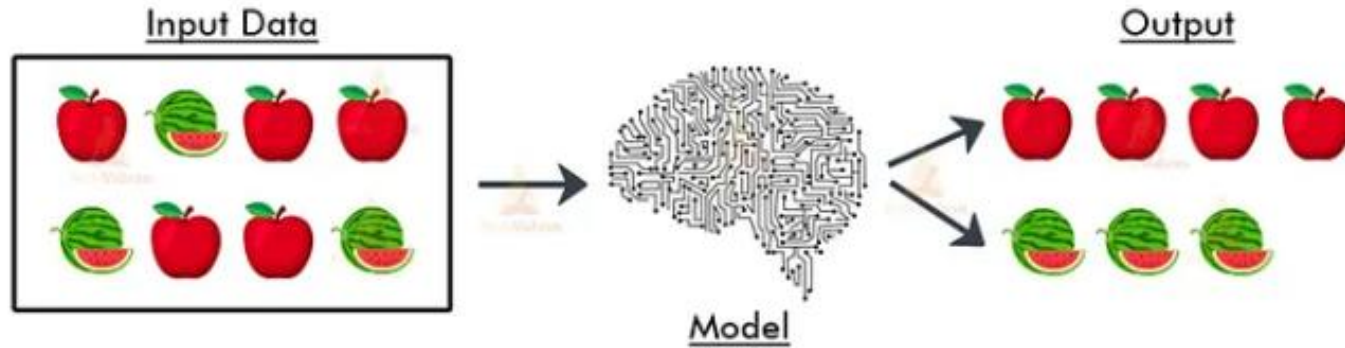
Denetimli Öğrenme

- Model, **etiketli (önceden sınıflandırılmış)** verilerle eğitilir.
- Girdi verisi ile doğru çıktı arasındaki ilişkiyi öğrenir.
- Amaç: Yeni verilere doğru tahminler yapmak.
- **Örnek:** Görüntü sınıflandırma (örneğin "kedi" – "köpek"), e-posta spam tespiti.



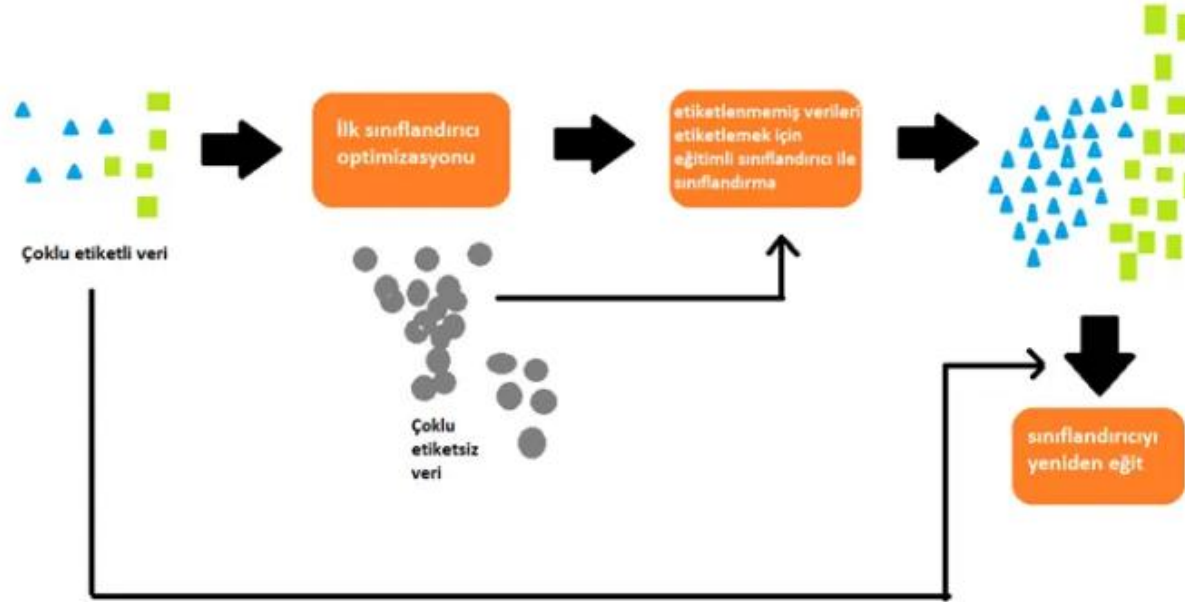
Denetimsiz Öğrenme

- Model, etiketsiz verilerle eğitilir.
- Verideki gizli örüntüleri, benzerlikleri ve grupları bulmaya çalışır.
- Örnek: Müşteri segmentasyonu, kümeleme (clustering), boyut indirgeme (PCA).



Yarı Denetimli Öğrenme

- Eğitim verisinin küçük bir kısmı **etiketli**, büyük kısmı **etiketsizdir**.
- Model, etiketli verilerden öğrendiği bilgiyi etiketsiz veriye **genelleştirerek öğrenmeyi geliştirir**.
- **Örnek:** Tıbbi görüntülerde, sadece bazı örnekler etiketlenmişken hastalık tespiti modeli oluşturma.



Pekiştirmeli Öğrenme

- Model, **deneme-yanılma** yoluyla öğrenir.
- Her doğru adımda **ödül (reward)** alır, yanlışta **ceza (penalty)** alır.
- **Amaç:** Belirli bir görevi en iyi şekilde tamamlamayı öğrenmek.
- **Örnek:** Oyun oynayan yapay zekâlar (örneğin AlphaGo), robotik hareket kontrolü, otonom araçlar.



Derin Öğrenmede Kullanılan Ağ Türleri

- **Yapay Sinir Ağları (ANN)** → Genel amaçlı modeller.
- **Evrişimli Sinir Ağları (CNN)** → Görüntü işleme ve nesne tanıma.
- **Tekrarlayan Sinir Ağları (RNN, LSTM, GRU)** → Zaman serisi, konuşma, metin işleme.
- **Generative Adversarial Networks (GANs)** → Yeni veri üretimi (örneğin sahte görseller, sanat üretimi).
- **Transformer Tabanlı Modeller** → Dil işleme, çeviri, sohbet botları (örneğin ChatGPT).

Yaygın Derin Öğrenme Mimarileri

- Yapay Sinir Ağları (ANN)
- Evrişimli Sinir Ağları (CNN)
- Tekrarlayan Sinir Ağları (RNN, LSTM, GRU)
- Otokodlayıcılar (Autoencoders)
- Üretici Çekişmeli Ağlar (GANs)
- Transformer Tabanlı Modeller

Yaygın Derin Öğrenme Mimarileri

1. Yapay Sinir Ağları (Artificial Neural Networks – ANN)

- **Tanım:** En temel derin öğrenme mimarisidir.
- **Yapısı:** Girdi katmanı, bir veya daha fazla gizli katman ve çıkış katmanı içerir.
- **Kullanım Alanları:** Basit sınıflandırma ve regresyon problemleri.
- **Örnek:** Ev fiyat tahmini, kredi skorlama.

2. Evrişimli Sinir Ağları (Convolutional Neural Networks – CNN)

- **Tanım:** Görüntü ve video işleme için özel olarak tasarlanmış sinir ağlarıdır.
- **Özellikler:**
 - **Evrişim katmanları (Convolution Layers):** Görüntüdeki kenar, renk ve desenleri tespit eder.
 - **Havuzlama katmanları (Pooling Layers):** Boyut indirgeme ve öznitelik çıkarımı yapar.
- **Kullanım Alanları:** Görüntü sınıflandırma, nesne tespiti, yüz tanıma.
- **Örnek:** MNIST el yazısı tanıma, ImageNet yarışması.

Yaygın Derin Öğrenme Mimarileri

3. Tekrarlayan Sinir Ağları (Recurrent Neural Networks – RNN)

- **Tanım:** Zaman serisi veya sıralı veriler için uygundur.
- **Özellikler:**
 - Önceki adımların bilgisini **hafızasında tutar** (state).
 - Uzun bağımlılıklar için **LSTM (Long Short-Term Memory)** ve **GRU (Gated Recurrent Unit)** varyantları kullanılır.
- **Kullanım Alanları:** Metin, konuşma ve müzik üretimi, zaman serisi tahminleri.
- **Örnek:** Dil modelleme, otomatik çeviri.

4. Otokodlayıcılar (Autoencoders)

- **Tanım:** Veriyi sıkıştırıp yeniden oluşturmayı öğrenen ağlardır.
- **Yapısı:** Encoder (sıkıştırma) ve Decoder (geri açma) katmanlarından oluşur.
- **Kullanım Alanları:** Boyut indirgeme, gürültü giderme (denoising), anomali tespiti.
- **Örnek:** Yüz görüntülerinden sıkıştırılmış özellik çıkarımı.

Yaygın Derin Öğrenme Mimarileri

5. Üretici Çekişmeli Ağlar (Generative Adversarial Networks – GANs)

- **Tanım:** İki ağtan oluşur: Üretici (Generator) ve Ayırt Edici (Discriminator).
- **Özellikler:**
 - Üretici, gerçek gibi görünen veri üretir.
 - Ayırt edici, üretilen veriyi gerçek veriden ayırmaya çalışır.
 - Bu iki ağ birbirini geliştirerek daha iyi veri üretir.
- **Kullanım Alanları:** Görüntü ve ses üretimi, sahte veri oluşturma, stil transferi.
- **Örnek:** Deepfake videolar, yapay resim üretimi.

6. Transformer Tabanlı Modeller

- **Tanım:** Özellikle doğal dil işleme (NLP) için geliştirilmiş mimarilerdir.
- **Özellikler:**
 - **Self-attention mekanizması** sayesinde uzun bağımlılıkları öğrenir.
 - Paralel işlem yaparak daha hızlı çalışır.
- **Kullanım Alanları:** Dil modelleme, çeviri, sohbet botları.
- **Örnek:** GPT, BERT, ChatGPT.

Derin Öğrenmede Araçlar ve Kütüphaneler

- TensorFlow, Keras, PyTorch, Theano
- Model Eğitimi ve Değerlendirme
- GPU / TPU Kullanımı

Derin Öğrenmenin Kullanım Alanları

- **Derin öğrenme**, otomotiv, havacılık, imalat, elektronik, tıbbi araştırma ve daha birçok alanda kullanılmaktadır.
- Bu teknoloji, **karmaşık verileri analiz ederek otomatik kararlar** alınmasını sağlar.
- ◆ **Kullanım Örnekleri:**
 - **Otonom otomobiller:**
 - Yol işaretlerini, yayaları ve diğer araçları **otomatik olarak algılamak** için derin öğrenme modelleri kullanılır.
 - **Savunma sistemleri:**
 - Uydu görüntülerinde **ilgi alanlarını** (örneğin askeri hedefleri veya riskli bölgeleri) otomatik olarak işaretler.
 - **Tıbbi görüntü analizi:**
 - Röntgen, MR veya mikroskop görüntülerinde **kanser hücrelerini otomatik olarak tespit etmek** için derin öğrenme uygulanır.
 - **İmalat ve fabrika otomasyonu:**
 - Derin öğrenme sistemleri, **insanların veya nesnelerin makinelere tehlikeli derecede yakınlaştığını algılayarak** güvenliği artırır.

Derin Öğrenmenin Uygulama Alanları

- Görüntü Tanıma ve Sınıflandırma
- Doğal Dil İşleme (NLP)
- Ses ve Konuşma Tanıma
- Otonom Araçlar
- Tıbbi Görüntü Analizi
- Siber Güvenlik ve Savunma Uygulamaları

Derin Öğrenmede Karşılaşılan Sorunlar

- Aşırı Uyum (Overfitting)
- Yetersiz Veri Sorunu
- Model Karmaşıklığı ve Hesaplama Maliyeti
- Açıklanabilir Yapay Zeka (Explainable AI)

Sonuç



Dr. Fatih KALEMKUŞ

Sorular



Dr. Fatih KALEMKUŞ

TEŐEKKÜRLER

Dr. Fatih KALEMKUŐ