



# **Yapay Zeka I: Veri Bilimi ve Makine Öğrenmesine Giriş Sertifika Programı**

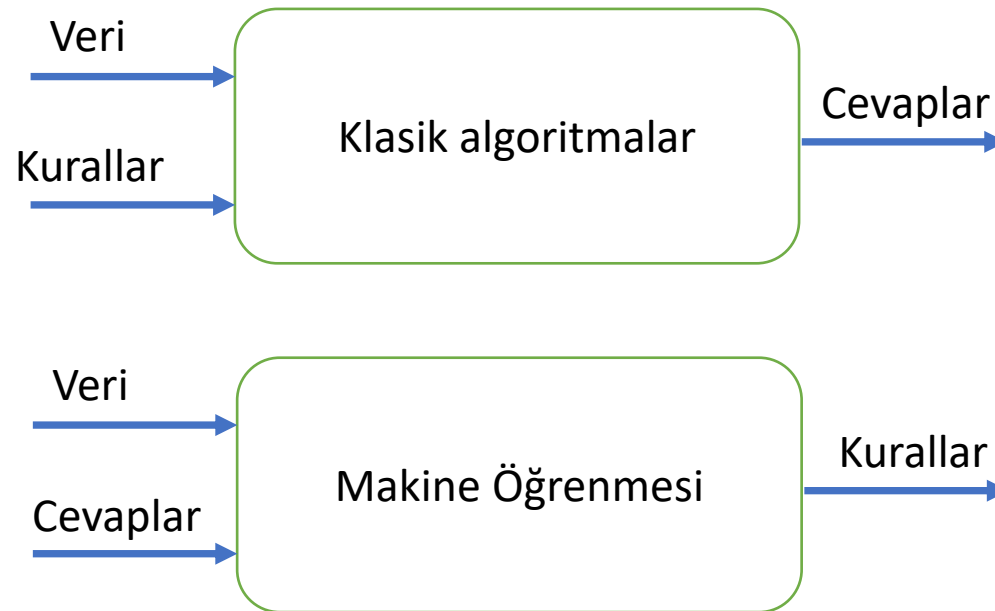
Doç. Dr. Taner Arsan

H. Fuat Alsan, PhD(c)

Sena Kılınç, PhD(c)

# Makine Öğrenmesine Giriş

- Yapay zekanın bir alt dalıdır
- Sabit kodlanmış kurallar yerine örüntü tanıma (pattern recognition)



# Öğrenme Paradigmaları

- Denetimli Öğrenme (Supervised Learning)
  - Etiketli veri kümesi gereklidir
  - Sonuçları değerlendirmek daha kolaydır
- Denetimsiz Öğrenme (Unsupervised learning)
  - Etiketli veri kümesine gerek duyulmaz
  - Sonuçları değerlendirmek daha zordur
- Ayrıca:
  - Yarı denetimli öğrenme (Semi-supervised learning)
  - Pekiştirmeli öğrenme (Reinforcement learning)
  - Kendinden denetimli öğrenme (Self-supervised learning)
  - Zayıf denetimli öğrenme (Weakly-supervised learning)
  - vb.

# Makine Öğrenmesi Görevleri

- Regresyon (Regression):
  - Sürekli (continuous) bir değeri tahmin
  - Örnek: Ev fiyatlarını tahmin etme
- Sınıflandırma (Classification):
  - Verilerin hangi nesne sınıfına ait olduğuna (önceden belirtilmiş sabit sayıda nesne sınıfı arasında) karar verme
  - Örnek: Bir görüntü verildiğinde, bu resmin bir araba mı yoksa kamyon mu olduğuna karar verme
- Kümeleme (Clustering):
  - Benzer verileri (bir benzerlik ölçütü kullanarak) gruptama
  - Denetimsiz (etiketli veri seti gerektirmez)
  - Örnek: Müşteri segmentasyonu

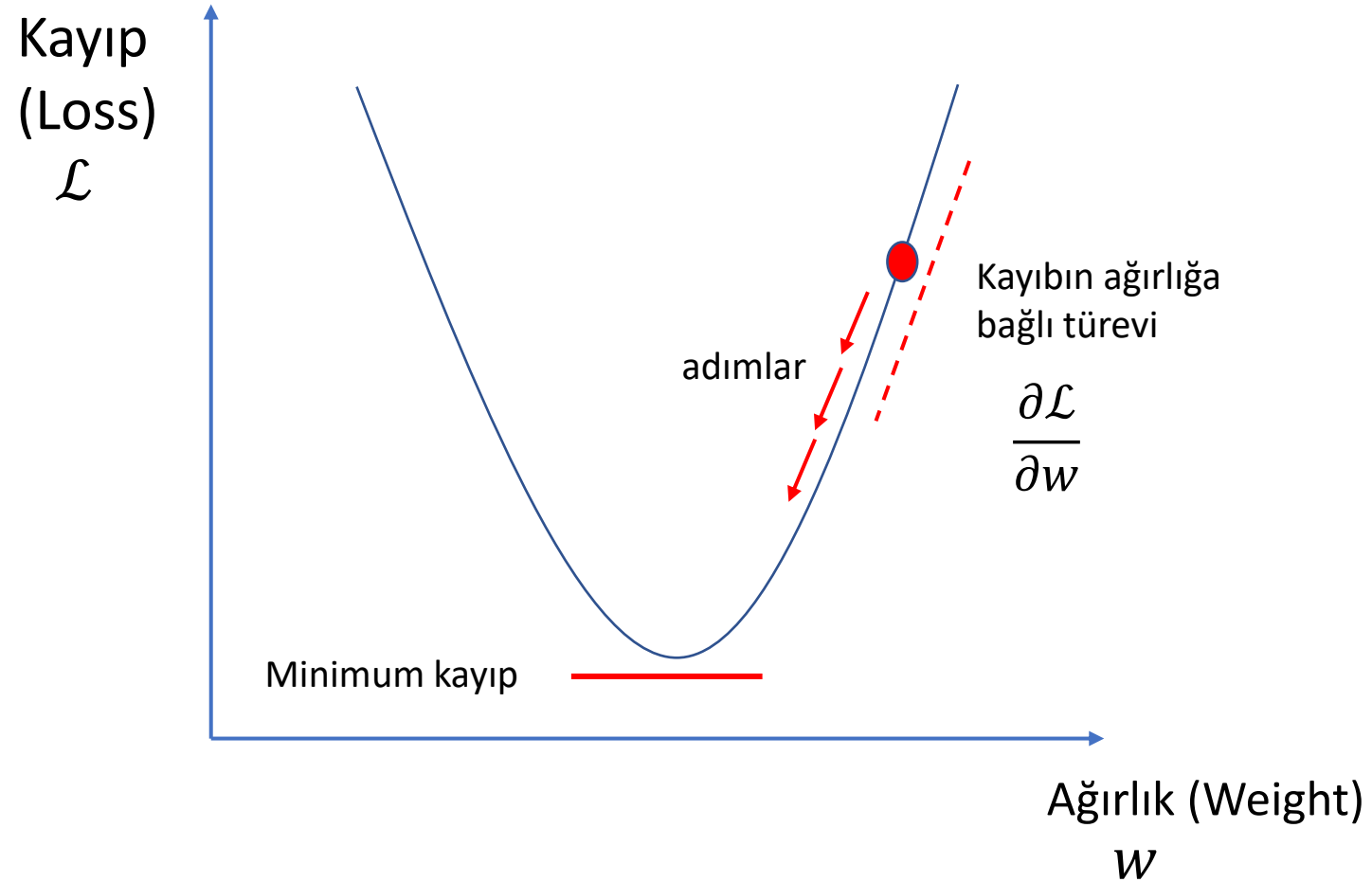
# Veri Üretmek

- Veri kümelerinin lisansları vardır ve ticari uygulamalara yönelik yasal sorunlar oluşturabilir
- Sentetik veri üretebiliriz
- Regresyon için fonksiyonla manuel olarak üretebiliriz:
  - $y = x + \epsilon$
- Aşağıdaki sklearn fonksiyonları da kullanılabilir:
  - (from sklearn import datasets)
  - datasets.make\_regression()
  - datasets.make\_classification()
  - datasets.make\_blobs()

# Gradyan Düşüşü (Gradient Descent)

- Makine öğreniminde, modelleri eğitmek için kapalı formda analitik çözümlerimiz yoktur.
- Bunun yerine, gradyan düşüşü gibi iteratif çözümleri kullanırız.
- **Döngü kullanarak kayıp fonksiyonunun model parametreleriyle olan türevini alır ve adım adım minimum kayba doğru hareket ederiz.**
- Yeterli iterasyondan sonra, modeller eğitilmiş olur ve tahmin yapmaya hazırdır.

# Gradyan Düşüşü (Görsel Olarak)



# Örnek: Temel Linear Regresyon

- Temel linear regresyon modeli:  $y = w_0 + w_1x$
- $w_0, w_1$  model parametreleri,  $x$  girdiler ve  $y$  çıktılar (tahminler)
- Kayıp (loss) fonksiyonu:  $\mathcal{L}(\hat{y}, y) = \frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i)^2$ 
  - $N$  toplam veri sayısı,  $y_i$  tahmin etmek istenilen hedef,  $\hat{y}_i$  modelin tahmini
  - Model tahmini ile hedef veriler arasındaki ortalama kare hata
  - Kayıbın düşük olması daha iyidir
- **Amaç:** Kayıp fonksiyonunun minimize edildiği  $w_0, w_1$  değerlerini bulmak istiyoruz. Buna modelin eğitilmesi (veya fit edilmesi) denir



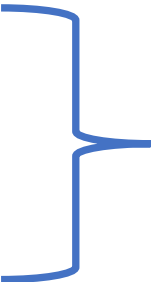
# Gradyan Hesabı

- Güç kuralını kullanarak:  $\frac{\partial}{\partial x} f(x)^n = n f(x)^{n-1} f'(x)$
- $\frac{\partial \mathcal{L}}{\partial w_0} = \frac{\partial \mathcal{L}}{\partial w_0} \frac{1}{N} \sum_{i=1}^N (w_0 + w_1 x_i - y_i)^2 = \frac{2}{N} \sum_{i=1}^N (w_0 + w_1 x_i - y_i)(1)$
- $\frac{\partial \mathcal{L}}{\partial w_0} = \frac{2}{N} \sum_{i=1}^N (\hat{y}_i - y_i)$
- $\frac{\partial \mathcal{L}}{\partial w_1} = \frac{\partial \mathcal{L}}{\partial w_1} \frac{1}{N} \sum_{i=1}^N (w_0 + w_1 x_i - y_i)^2 = \frac{2}{N} \sum_{i=1}^N (w_0 + w_1 x_i - y_i)(x_i)$
- $\frac{\partial \mathcal{L}}{\partial w_1} = \frac{2}{N} \sum_{i=1}^N (\hat{y}_i - y_i)(x_i)$

# Model Optimizasyonu

- Kaybın azaltılması için model parametrelerinin güncellenmesi gerekir

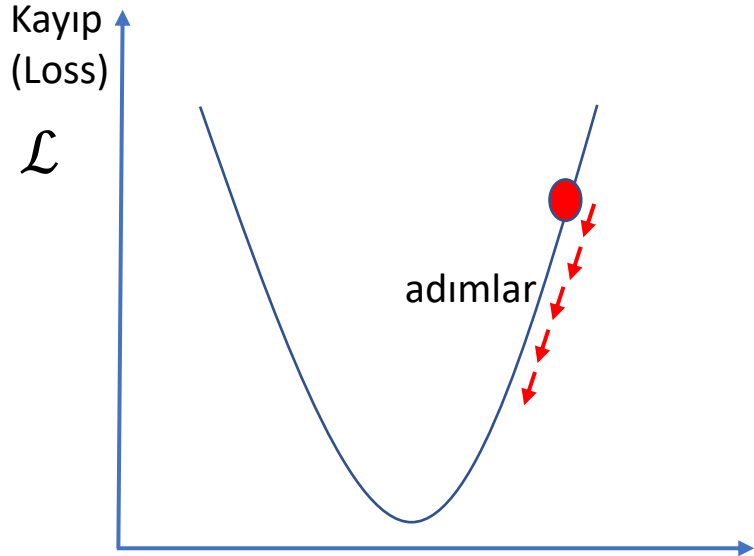
- $w_0 = w_0 - \alpha \frac{\partial \mathcal{L}}{\partial w_0}$
- $w_1 = w_1 - \alpha \frac{\partial \mathcal{L}}{\partial w_1}$



Kayıp en aza indirilene kadar döngü olarak devam eder

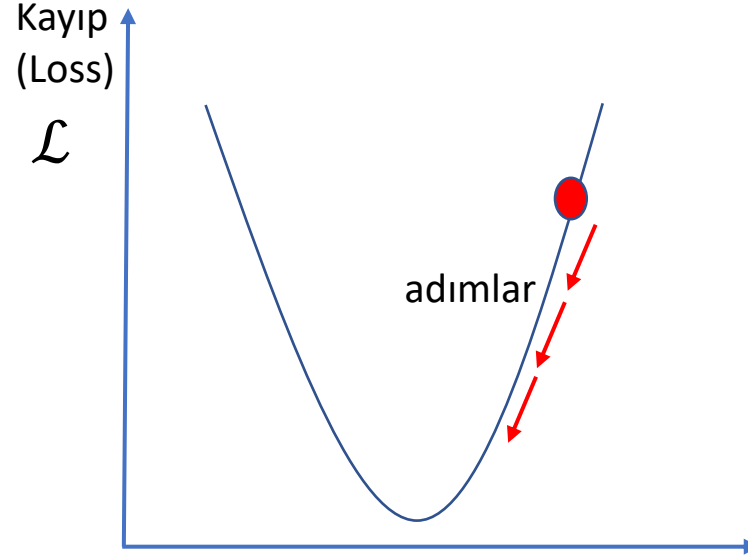
- $\alpha$  öğrenme oranı (learning rate), öğrenmenin hızını belirler
  - Dikkatli seçilmelidir
  - Çok küçük olursa -> eğitim çok uzun sürer
  - Çok büyük olursa -> eğitim dengesizleşir

# Öğrenme Oranını Doğru Seçmek



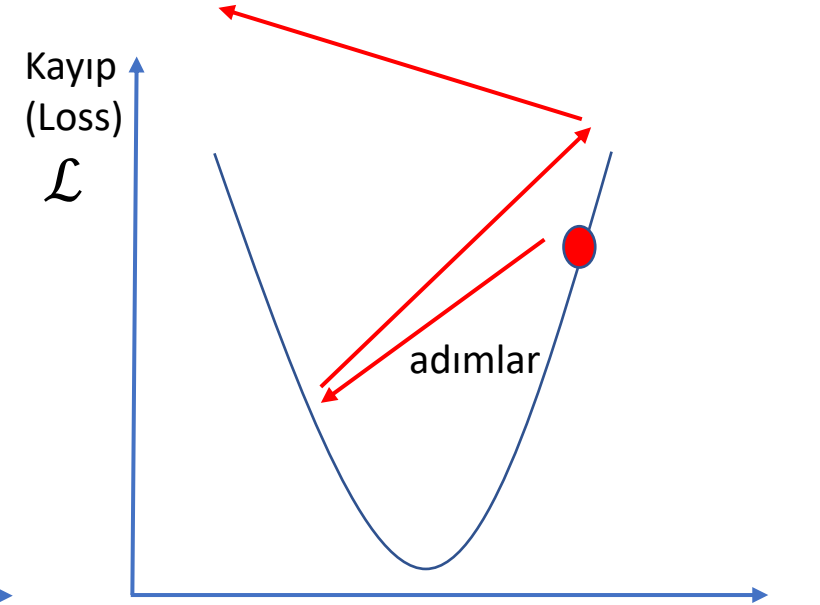
## Düşük Öğrenme Oranı

Küçük adımlar, eğitim çok uzun sürer



## Doğru Öğrenme Oranı

Dengeli öğrenme sağlar



## Yüksek Öğrenme Oranı

Adımlar çok büyük atıldığı için dengesizlik oluşturur

# İkinci Örnek: Polinom Regresyonu

- İkinci dereceden model:  $y = w_0 + w_1x + w_2x^2$
- Gradyanları daha önce aynı şekilde hesapladığımızda aşağıdakileri ederiz:
  - *(hesaplamalar atlanmıştır ancak egzersiz için bunu kendi başınıza hesaplayabilirsiniz)*

- $\frac{\partial \mathcal{L}}{\partial w_0} = \frac{2}{N} \sum_{i=1}^N (\hat{y}_i - y_i)$

- $\frac{\partial \mathcal{L}}{\partial w_1} = \frac{2}{N} \sum_{i=1}^N (\hat{y}_i - y_i)(x_i)$

- $\frac{\partial \mathcal{L}}{\partial w_2} = \frac{2}{N} \sum_{i=1}^N (\hat{y}_i - y_i)(x_i^2)$

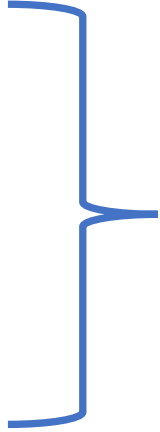
# Polinom Regresyon Modeli Optimizasyonu

- Yöntem olarak aynı fakat öğrenilebilir üç parametremiz var

- $w_0 = w_0 - \alpha \frac{\partial \mathcal{L}}{\partial w_0}$

- $w_1 = w_1 - \alpha \frac{\partial \mathcal{L}}{\partial w_1}$

- $w_2 = w_2 - \alpha \frac{\partial \mathcal{L}}{\partial w_2}$



Kayıp en aza indirilene  
kadar döngü olarak  
devam eder

# Gradyan Düşüşü Hakkında Notlar

- Gradyan Patlamaları (Exploding gradients)
  - Gradyan değerleri çok büyük sayılara ulaşabilir (buna dikkat edin)
- Kaybolan Gradyanlar (Vanishing gradients)
  - Gradyan değerleri çok küçük sayılara ulaşabilir (buna dikkat edin)
- Bazı fonksiyonların sınırlayıcı özellikleri vardır
  - Örnek:  $\log()$  fonksiyonu negatif sayıları alamaz (buna dikkat edin)
- Öğrenme oranı dikkatlice seçilmelidir

# Diğer Eğitim Yöntemleri

- Least Squares
  - Basit ancak sadece temel modellere uygulanabilir (örnek: Lineer Regresyon)
- Maximum Likelihood Estimation (MLE)
  - Olasılık dağılımlarını uygun hale getirmek için kullanışlıdır
- Maximum A Posteriori (MAP) Estimation
  - MLE'ye benzer ancak Bayes Kuralı'nı kullanır