

# Phase 0 · Temel Matematik

ML Builder için Türkçe Notlar

Phase 1

2026-06-18



# İçindekiler

<b>Önsöz</b>	<b>1</b>
0.1. Bu Kurs Nedir?	1
0.2. Nasıl Okunmalı	1
0.3. Dersler	1
<b>I. Bölüm 1 — Cebir</b>	<b>5</b>
<b>1. Değişken, İfade ve Denklem</b>	<b>7</b>
1.1. Değişken	7
1.2. İfade	7
1.3. Denklem	7
1.4. İfade mi, denklem mi?	7
1.5. Örnek	8
1.6. Alıştırmalar	9
<b>2. Denklem Mantiği</b>	<b>11</b>
2.1. Cebir nedir?	11
2.2. Değişken, ifade, denklem	11
2.3. Denklem bir terazidir	12
2.4. Ters işlemler	12
2.5. Örnekler	12
2.6. Kesirli denkleme küçük giriş	13
2.7. Alıştırmalar	13
<b>3. Kesirli Denklemler</b>	<b>15</b>
3.1. Temel fikir: paydayı sil	15
3.2. Bilinmeyen paydadaysa	15
3.3. İki kesir eşitse: çapraz çarpım	16
3.4. Birden çok kesir: ortak payda	16
3.5. Örnekler	16
3.6. Alıştırmalar	17
<b>4. Üslü İfadeler</b>	<b>19</b>
4.1. Üs nedir?	19
4.2. Temel kurallar	19
4.3. Sıfır ve negatif üs	19
4.4. Kesirli üs = kök	20
4.5. Örnekler	20
4.6. Alıştırmalar	21

<b>5. Çarpanlara Ayırma</b>	<b>23</b>
5.1. Çarpanlara ayırma nedir?	23
5.2. Ortak çarpan	23
5.3. Üç terimli ifade	23
5.3.1. İşaretleri hızlı bulma	24
5.4. Örnekler	24
5.5. Alıştırmalar	25
<b>6. Özdeşlikler</b>	<b>27</b>
6.1. Üç özdeşlik	27
6.2. $(a + b)^2$ neden orta terimli?	27
6.3. $(a - b)^2$	27
6.4. İki kare farkı	29
6.5. Örnekler	29
6.6. Alıştırmalar	30
<b>7. İkinci Derece Denklemler</b>	<b>31</b>
7.1. İkinci derece denklem nedir?	31
7.2. Yol 1 — Çarpanlara ayırma	31
7.3. Yol 2 — Formül	31
7.4. Diskriminant: kaç kök var?	32
7.5. Geometrik anlam: parabolün kökleri	32
7.6. Örnekler	33
7.7. Alıştırmalar	33
<b>8. Eşitsizlikler</b>	<b>35</b>
8.1. Eşitsizlik nedir?	35
8.2. Çözme: tek kritik fark	35
8.3. Çözümü gösterme — sayı doğrusu	35
8.4. Aralık notasyonu	36
8.5. Bileşik eşitsizlik	36
8.6. Örnekler	36
8.7. Alıştırmalar	37
<b>9. Logaritma</b>	<b>39</b>
9.1. Logaritma nedir?	39
9.2. Üs ve logaritma: ayna görüntüsü	39
9.3. Yaygın tabanlar	39
9.4. Logaritma kuralları	41
9.5. Örnekler	41
9.6. Alıştırmalar	42
<b>II. Bölüm 2 — Fonksiyonlar</b>	<b>43</b>
<b>10. Fonksiyon Kavramı</b>	<b>45</b>
10.1. Fonksiyon nedir?	45
10.2. $f(x)$ gösterimi	45
10.3. Tanım ve görüntü kümesi	46

10.4. Örnekler . . . . .	46
10.5. Alıştırmalar . . . . .	47
<b>11. Doğrusal Fonksiyon ve Eğim</b>	<b>49</b>
11.1. Doğrusal fonksiyon: $y = mx + b$ . . . . .	49
11.2. Eğim . . . . .	49
11.3. $y$ -kesişimi . . . . .	49
11.4. İki noktadan eğim . . . . .	51
11.5. Örnekler . . . . .	51
11.6. Alıştırmalar . . . . .	52
<b>12. Parabol</b>	<b>53</b>
12.1. Parabol: $y = ax^2 + bx + c$ . . . . .	53
12.2. $a$ kolların yönünü ve genişliğini belirler . . . . .	53
12.3. Tepe noktası . . . . .	53
12.4. Kökler ve diskriminant . . . . .	55
12.5. Örnek . . . . .	55
12.6. Alıştırmalar . . . . .	55
<b>13. Üstel ve Logaritmik Fonksiyon</b>	<b>57</b>
13.1. Üstel fonksiyon: $f(x) = b^x$ . . . . .	57
13.2. Doğrusal vs üstel: sabit fark mı, sabit oran mı? . . . . .	57
13.3. Logaritmik fonksiyon: $f(x) = \log_b x$ . . . . .	59
13.4. Doğal taban $e$ . . . . .	59
13.5. Örnek . . . . .	59
13.6. Alıştırmalar . . . . .	60
<b>14. Fonksiyon Bileşkesi</b>	<b>61</b>
14.1. Bileşke nedir? . . . . .	61
14.2. Hesaplama . . . . .	62
14.3. Sıra önemlidir . . . . .	62
14.4. Örnek . . . . .	62
14.5. Alıştırmalar . . . . .	63
<b>15. Grafik Okuma</b>	<b>65</b>
15.1. Grafik bir fonksiyonun resmidir . . . . .	65
15.2. Neler okunur? . . . . .	65
15.3. Bir değeri okumak . . . . .	65
15.4. Alıştırmalar . . . . .	67
<b>III. Bölüm 3 — Toplam, Diziler ve Sayma</b>	<b>69</b>
<b>16. Diziler</b>	<b>71</b>
16.1. Dizi nedir? . . . . .	71
16.2. Aritmetik dizi: sabit fark . . . . .	71
16.3. Geometrik dizi: sabit oran . . . . .	71
16.4. Bir terimi bulmak . . . . .	72
16.5. Örnek . . . . .	72

16.6. Alıştırmalar	73
<b>17. Toplam Sembölü (<math>\Sigma</math>)</b>	<b>75</b>
17.1. $\Sigma$ ne demek?	75
17.2. Nasıl açılır?	76
17.3. $\Sigma$ kuralları	76
17.4. Ortalama bir $\Sigma$ 'dir	76
17.5. Örnek	77
17.6. Alıştırmalar	77
<b>18. Pi Çarpım Notasyonu</b>	<b>79</b>
18.1. $\Pi$ ne demek?	79
18.2. Nasıl hesaplanır?	79
18.3. Faktöriyel bir çarpımdır	79
18.4. $\Pi \rightarrow \log \rightarrow \Sigma$	80
18.5. Örnek	81
18.6. Alıştırmalar	81
<b>19. Seriler</b>	<b>83</b>
19.1. Seri nedir?	83
19.2. Aritmetik seri	83
19.3. Geometrik seri	84
19.4. Örnek	85
19.5. Alıştırmalar	85
<b>20. Faktöriyel ve Sayma</b>	<b>87</b>
20.1. Faktöriyel: $n!$	87
20.2. Sayma temel ilkesi	87
20.3. Permütasyon: sıra önemli	87
20.4. Kombinasyon: sıra önemsiz	88
20.5. Örnek	89
20.6. Alıştırmalar	89
<b>IV. Bölüm 4 — Vektörler</b>	<b>91</b>
<b>21. Vektör Kavramı</b>	<b>93</b>
21.1. Vektör nedir?	93
21.2. Boyut	93
21.3. Gösterim ve eşitlik	93
21.4. Örnek	95
21.5. Alıştırmalar	95
<b>22. Vektör İşlemleri</b>	<b>97</b>
22.1. Vektör toplama	97
22.2. Skaler çarpım	97
22.3. Vektör çıkarma	97
22.4. Örnek	99
22.5. Alıştırmalar	99

<b>23. İç Çarpım</b>	<b>101</b>
23.1. İç çarpım nasıl hesaplanır?	101
23.2. Sonuç bir skalerdir	101
23.3. Geometrik anlam: açı	101
23.4. Nöronun hesabı	102
23.5. Örnek	102
23.6. Alıştırmalar	103
<b>24. Vektör Uzunluğu ve Kosinüs Benzerliği</b>	<b>105</b>
24.1. Vektör uzunluğu (norm)	105
24.2. Birim vektör	105
24.3. Kosinüs benzerliği	107
24.4. Örnek	107
24.5. Alıştırmalar	108
<b>25. Machine Learning Bağlantısı</b>	<b>109</b>
25.1. Model bir fonksiyondur	109
25.2. Bir nöron: dört bölümün buluştuğu nokta	109
25.3. Öğrenme: hatayı azaltmak	109
25.4. Derin ağ: fonksiyon bileşkesi	110
25.5. Benzerlik ve getirme: embedding / RAG	110
25.6. Foundation → ML haritası	110
25.7. Alıştırmalar	111



# Önsöz


## 0.1. Bu Kurs Nedir?

Bu kurs, makine öğrenmesine başlamadan önce gereken **temel matematiği sıfırdan** kurar. Calculus, Lineer Cebir ve Olasılık derslerine geçmeden cebir, denklem mantığı ve sayısal düşünmeyi sağlam bir zemine oturtmayı hedefler.

Amaç işlem ezberlemek değil, *neden* öyle yaptığımızın mantığını anlamaktır. Bu mantık ileride bir modelin bilinmeyen parametresini çözmekten doğrusal model  $y = wx + b$ 'yi anlamaya kadar her yerde karşına çıkar.

## 0.2. Nasıl Okunmalı

Sıralı oku. Her ders kısa bir kavramla başlar, bol örnekle ilerler ve alıştırmalarla biter.

 Pratik bir tavsiye

Alıştırma çözümleri açılır kutularda — önce kendin dene, sonra aç. Her dersteki **ML köprüsü** kutusu, öğrendiğin konunun ileride makine öğrenmesinde nerede işine yarayacağını gösterir.

## 0.3. Dersler

#	Ders	Ana Fikir
1	Değişken, İfade, Denklem	Cebirin üç temel kelimesi — başlangıç
2	Denklem Mantığı	Bilinmeyeni izole etme; dengeyi bozmadan çözme
3	Kesirli Denklemler	Paydadan kurtulma; çapraz çarpım, ortak payda
4	Üslü İfadeler	Tekrarlı çarpma kuralları; sıfır, negatif, kesirli üs
5	Çarpanlara Ayırma	Dağıtmanın tersi; ortak çarpan, üç terimli ifade
6	Özdeşlikler	Tam kare ve iki kare farkı; açma + çarpanlara ayırma
7	İkinci Derece Denklemler	Çarpanlara ayırma vs formül; diskriminant, parabol kökleri

#	Ders	Ana Fikir
8	Eşitsizlikler	Yön kuralı, sayı doğrusu, aralık notasyonu
9	Logaritma	Üssün tersi; çarpımı toplama çeviren kurallar
10	Fonksiyon Kavramı	Girdi→çıkıktı kuralı; $f(x)$ , tanım/görüntü kümesi
11	Doğrusal Fonksiyon ve Eğim	$y = mx + b$ ; eğim, kesişim — lineer regresyon/nöron
12	Parabol	$y = ax^2 + bx + c$ ; tepe, kökler, simetri — kayıp çukuru
13	Üstel ve Logaritmik Fonksiyon	$b^x$ büyüme/çürüme; $e$ , sigmoid'ın temeli
14	Fonksiyon Bileşkesi	$f(g(x))$ ; içten dışa — sınır ağının iskeleti
15	Grafik Okuma	Eğriden bilgi çıkarmak; kesişim, artan/azalan, maks — kayıp eğrisi
16	Diziler	Sıralı sayı listeleri; aritmetik/geometrik — veri indeksleme
17	Toplam Sembolü ( $\Sigma$ )	$\Sigma$ anatomisi ve kuralları; ortalama, MSE, iç çarpım
18	Pi (II) Çarpım Notasyonu	$\Sigma$ 'nın çarpım kardeşi; faktöriyel, olabilirlik
19	Seriler	Dizinin toplamı; aritmetik/geometrik formül, yakınsama
20	Faktöriyel ve Sayma	$n!$ , permütasyon, kombinasyon — olasılığa kapı
21	Vektör Kavramı	Sayı listesi = nokta/ok; boyut — ML'in doğal nesnesi
22	Vektör İşlemleri	Toplama, skaler çarpım — gradyan inişi güncellemesi
23	İç Çarpım	Bileşen çarp+topla; açılış/ışaret — nöronun hesabı $w \cdot x + b$
24	Vektör Uzunluğu ve Kosinüs Benzerliği	Norm, birim vektör, kosinüs — embedding/RAG
25	Machine Learning Bağlantısı	Tüm parçaların ML'de buluşması — büyük harita

! Bir tek şey

Cebir, bilmediğin bir sayıyı bir harfle gösterip onu bulma yöntemidir. Denklem çözmek = bir bilinmeyeni **izole etmek**. Bu fikir,  $y = wx + b$ 'den binlerce parametrelili modellere kadar değişmez.

*“Tuning is physics, mathematics, and logic — not magic.”*



**Kısım I.**

**Bölüm 1 — Cebir**



# 1. Değişken, İfade ve Denklem

Cebirin üç temel kelimesi — başlangıç noktası

Denklem çözmeye geçmeden önce üç kelime: **değişken**, **ifade**, **denklem**. Cebirin en temel sözlüğü — ve en sık karıştırılan üçlü. Bu ayrımı bir kez oturtursan, gerisi çok daha kolay gelir.

## 1.1. Değişken

Değişken, bilmediğimiz ya da değişebilen bir sayının yerini tutan **harftir**:  $x, y, n$ . Harf kullanmamızın sebebi, bir sayıyı henüz bilmesek de onunla işlem yapabilmek ve genel kurallar yazabilmektir. Örneğin “bir sayının 2 katının 3 fazlası” ifadesini  $2x + 3$  diye yazarız.

## 1.2. İfade

İfade; sayıların, değişkenlerin ve işlemlerin birleşimidir — ama **eşittir işareti yoktur**. Bir değeri temsil eder (değişkene bağlı olarak değişir). İfadeyi *hesaplarsın* ya da *sadeleştirirsin*, ama “çözmezsin”. Örneğin  $2x + 3$  bir ifadedir;  $x = 5$  koyunca değeri  $2(5) + 3 = 13$  olur.

## 1.3. Denklem

Denklem, iki ifadeyi **eşittir** ile birleştiren ifadedir:  $2x + 3 = 7$ . Bu,  $x$ 'in değerine göre doğru ya da yanlış olabilen bir iddiadır; “çözmek”, onu doğru yapan  $x$ 'i bulmaktır (burada  $x = 2$ ).

Yukarıdaki  $2x + 3 = 7$  denkleminde: 2 değişkenin **katsayısı**,  $x$  **değişken**, 3 **sabit** terimdir.  $2x$  ve 3 ise birer **terimdir**.

## 1.4. İfade mi, denklem mi?

Tek bir soru ayırır:

- **Eşittir işareti yok** → ifade. Onu hesaplar veya sadeleştirirsin.
- **Eşittir işareti var** → denklem. Onu çözer (bilinmeyeni bulur)sun.

Örneğin  $3x - 1$  bir ifadedir;  $3x - 1 = 8$  bir denklemdir.

## 1. Değişken, İfade ve Denklem

ifade

$$2x + 3 = 7$$

eşittir → bunu denklem yapar

katsayı      değişken      sabit

Şekil 1.1.: Bir denklemin parçaları: katsayı, değişken ve sabit. Eşittir işaretinin solu bir ifadedir; eşittir işareti bunu bir denkleme çevirir.

### ! Tek ayırım

Eşittir işareti yoksa **ifadedir** (hesaplanır/sadeleştirilir); eşittir işareti varsa **denklemdir** (çözülür). Bu tek ayırım, başlangıçtaki kafa karışıklığının çoğunu önler.

### ⚠ İki tuzak

- İfadeyi “çözemezsin” — çözülecek bir eşitlik yok; ifadeyi yalnızca hesaplıyorsun.
- $2x$ ,  $2 \cdot x$  demektir; bitişik yazım çarpımı gizler (çarpı işareti yazılmaz).

## 1.5. Örnek

**Örnek 1.** “Bir sayının 5 fazlası”  $\rightarrow x + 5$  (ifade).

**Örnek 2.**  $2x + 1$  ifadesini  $x = 4$  için hesapla:  $2(4) + 1 = 9$ .

**Örnek 3.** “Bir sayının 2 katı 10’a eşittir”  $\rightarrow 2x = 10$  (denklem).

### 💡 ML köprüsü

Makine öğrenmesinde çözmeye çalıştığımız bilinmeyenler, modelin **parametreleridir** (ağırlıklar). Bir modelin formülü — örneğin  $wx + b$  — bir **ifadedir**; eğitim, bu ifadeyi veriye uydurmak için en iyi  $w$  ve  $b$  değerlerini arayan bir problem kurar. Yani “değişkeni izole edip bulma” fikri, doğrudan “parametreyi öğrenme” fikrine dönüşür.

**1.6. Alıřtırmalar**

1. “Bir sayının 5 fazlası” ifadesini  $x$  ile yaz.
2.  $3x - 2$  bir ifade mi, denklem mi?
3.  $3x - 2 = 7$  bir ifade mi, denklem mi?
4.  $2x + 1$  ifadesini  $x = 4$  için hesapla.
5.  $5x + 3$  ifadesinde katsayı ve sabit nedir?
6. “Bir sayının 2 katı 10’a eřittir” cümlesini denklem olarak yaz.

**i** Çözümler (önce kendin dene, sonra aç)

1.  $x + 5$
2. İfade (eřittir iřareti yok → hesaplanır, çözümlmez)
3. Denklem (eřittir iřareti var →  $x$  çözümlür)
4.  $2(4) + 1 = 9$
5. Katsayı 5, sabit 3
6.  $2x = 10$

---

Sonraki ders: *Denklem Mantığı (doğrusal denklemler).*



## 2. Denklem Mantığı

Bu derste cebirin en temel taşını koyuyoruz: **denklem çözmek**. Amaç işlem ezberlemek değil, *neden* öyle yaptığımızın mantığını kurmak. Bu mantık ileride bir modelin bilinmeyen parametresini çözmekten doğrusal model  $y = wx + b$ 'yi anlamaya kadar her yerde karşına çıkacak.

### 2.1. Cebir nedir?

Cebir, bilmediğimiz bir sayıyı bir harfle gösterip onu bulma yöntemidir.

$$x + 5 = 12$$

Buradaki  $x$  henüz bilmediğimiz sayıdır. Soru şu: *hangi sayıya 5 eklersek 12 olur?* Cevap  $x = 7$ , çünkü  $7 + 5 = 12$ .

### 2.2. Değişken, ifade, denklem

#### **i** Kavram

- **Değişken:** bilmediğimiz ya da değişebilen sayı. Harfle gösterilir:  $x, y, a, b$ .
- **İfade:** eşittir işareti *olmayan* matematiksel yazı. Örn.  $3x + 5$ .
- **Denklem:** iki tarafı birbirine eşit olan yazı. Örn.  $3x + 5 = 20$ .

$3x + 5$  ifadesinde:

- $x \rightarrow$  değişken
- $3 \rightarrow$  katsayı
- $5 \rightarrow$  sabit sayı
- $3x \rightarrow 3 \times x$  demektir ( $3 + x$  **değil**)

Örneğin  $x = 4$  ise  $3x = 3 \times 4 = 12$ .

## 2. Denklem Mantığı

### 2.3. Denklem bir terazidir

$x + 5 = 12$  denkleminde sol taraf ile sağ taraf eşit ağırlıktadır. Dengeyi bozmamak için bir tarafa yaptığın işlemi diğerine de yapmalısın. İki taraftan da 5 çıkaralım:

$$x + 5 - 5 = 12 - 5 \Rightarrow x = 7$$

#### ! Altın kural

Denklemin bir tarafına yaptığın işlemi diğer tarafına da yapmalısın. Amaç:  $x$ 'i yalnız bırakmak.

### 2.4. Ters işlemler

Bir sayının yanındaki işlemi kaldırmak için **ters** işlemi kullanırız:

İşlem	Ters işlem
+5	-5
-5	+5
$\times 5$	$\div 5$
$\div 5$	$\times 5$

### 2.5. Örnekler

**Örnek 1.**  $x + 8 = 15$ . Yanındaki +8'in tersi -8:

$$x + 8 - 8 = 15 - 8 \Rightarrow x = 7 \quad (\text{kontrol: } 7 + 8 = 15)$$

**Örnek 2.**  $x - 4 = 10$ . Tersini +4:

$$x - 4 + 4 = 10 + 4 \Rightarrow x = 14$$

**Örnek 3.**  $3x = 18$ .  $x$ , 3 ile çarpıldığı için iki tarafı 3'e böleriz:

$$\frac{3x}{3} = \frac{18}{3} \Rightarrow x = 6$$

**Örnek 4 (iki işlem).**  $2x + 5 = 17$ . İşlemleri ters sırayla kaldır — önce +5, sonra  $\times 2$ :

$$2x = 12 \Rightarrow x = 6 \quad (\text{kontrol: } 2(6) + 5 = 17)$$

**Örnek 5 (iki tarafta da  $x$ ).**  $5x - 4 = 3x + 10$ . Önce  $x$ 'leri bir tarafta topla (iki taraftan  $3x$  çıkar):

$$2x - 4 = 10 \Rightarrow 2x = 14 \Rightarrow x = 7$$

Kontrol: sol taraf  $5(7) - 4 = 31$ , sağ taraf  $3(7) + 10 = 31$ . ✓

⚠️ “Karşıya geçince işaret değişir” aslında ne demek?

Okulda “ $x + 5 = 12 \rightarrow 5$  karşıya  $-5$  olarak geçer” denir. Bu pratik bir kısayol, ama sayı kendi kendine karşıya geçmez. Gerçekte yaptığın şey **iki taraftan da 5 çıkarmaktır**:

$$x + 5 - 5 = 12 - 5$$

Mantığı böyle öğrenirsen daha karmaşık denklemlerde hata yapmazsın.

## 2.6. Kesirli denkleme küçük giriş

$$\frac{x}{4} = 5$$

Burada  $x$ , 4’e bölünmüş. Bölmenin tersi çarpmadır; iki tarafı 4 ile çarp:

$$4 \times \frac{x}{4} = 5 \times 4 \Rightarrow x = 20$$

💡 ML köprüsü

Denklem çözmek = bir bilinmeyeni **izole etmek**. Makine öğrenmesinde doğrusal modelin kendisi bir denklemdir:  $y = wx + b$ . Eğitimde “doğru  $w$  ve  $b$  nedir?” sorusunu, tıpkı buradaki gibi dengeyi bozmadan adım adım çözeriz. Bugün öğrendiğin izole etme mantığı, ileride binlerce parametrelili sistemlerin temelidir.

## 2.7. Alıştırmalar

Aşağıdaki denklemleri çöz:

1.  $x + 9 = 16$
2.  $x - 7 = 12$
3.  $4x = 28$
4.  $2x + 3 = 15$
5.  $3x - 5 = 16$
6.  $5x + 2 = 3x + 18$
7.  $\frac{x}{4} + 3 = 8$
8. **Sözel:** Bir sayının 3 katının 4 fazlası 25’tir. Denklemi  $3x + 4 = 25$ ;  $x$ ’i bul.

📌 Çözümler (önce kendin dene, sonra aç)

1.  $x = 7$
2.  $x = 19$
3.  $x = 7$
4.  $2x = 12 \Rightarrow x = 6$

## 2. Denklem Mantığı

$$5. 3x = 21 \Rightarrow x = 7$$

$$6. 5x - 3x = 18 - 2 \Rightarrow 2x = 16 \Rightarrow x = 8$$

$$7. \frac{x}{4} = 5 \Rightarrow x = 20$$

$$8. 3x = 21 \Rightarrow x = 7$$

---

*Sonraki ders: Kesirli Denklemler ve Üslü İfadeler.*

## 3. Kesirli Denklemler

Paydadadan kurtulma — denklem mantığının kesirli hâli

Bir önceki derste denklemi terazi gibi çözmeyi öğrendik: bir tarafa yaptığını diğerine de yap, bilinmeyeni yalnız bırak. Kesirli denklemde aynı mantık geçerli; tek ek adım, **paydadadan kurtulmak**. Kesir gözünü korkutmasın — bir kez paydayı temizleyince elinde sıradan bir denklem kalır.

### 3.1. Temel fikir: paydayı sil

Bir tarafta payda varsa, **iki tarafı o paydayla çarp**. Bölmenin tersi çarpmadır:

$$\frac{x}{4} = 5 \quad \xrightarrow{\times 4} \quad x = 20$$

İki tarafı 4 ile çarptık; soldaki 4 sadeleşti,  $x$  yalnız kaldı.

### 3.2. Bilinmeyen paydadaysa

Bilinmeyen paydadaysa da yöntem aynı — iki tarafı o ifadeyle çarp:

$$\frac{12}{x} = 3 \quad \xrightarrow{\times x} \quad 12 = 3x \quad \Rightarrow \quad x = 4$$

 Bir şart: payda sıfır olamaz

$\frac{12}{x}$  ifadesinde  $x = 0$  olamaz, çünkü sıfıra bölme tanımsızdır. Bilinmeyen paydadayken bulduğun cevabın paydayı sıfır yapmadığını her zaman kontrol et.

### 3. Kesirli Denklemler

#### 3.3. İki kesir eşitse: çapraz çarpım

$\frac{a}{b} = \frac{c}{d}$  biçiminde iki kesir eşitse, **çaprazları çarparsın**:  $a \cdot d = b \cdot c$ . Bu aslında iki tarafı  $b \cdot d$  ile çarpmanın kısayoludur.

$$\frac{x+1}{2} = \frac{x-1}{3} \Rightarrow 3(x+1) = 2(x-1)$$
$$3x+3 = 2x-2 \Rightarrow x = -5$$

Kontrol: sol  $\frac{-5+1}{2} = \frac{-4}{2} = -2$ , sağ  $\frac{-5-1}{3} = \frac{-6}{3} = -2$ . ✓

#### 3.4. Birden çok kesir: ortak payda

Toplama/çıkarma içeren kesirli denklemde iki tarafı **ortak payda**yla (paydaların en küçük ortak katı) çarp. Böylece tüm kesirler aynı anda silinir:

$$\frac{x}{2} + \frac{x}{3} = 5 \xrightarrow{\times 6} 3x + 2x = 30 \Rightarrow 5x = 30 \Rightarrow x = 6$$

#### ! Altın kural

Paydayla çarparken **her terimi** çarp — sadece kesirli olanları değil, eşittirin diğer tarafındaki sayıyı da.  $\frac{x}{2} + \frac{x}{3} = 5$ 'te 6 ile çarpınca 5 de 30 olur. En sık hata, bir terimi atlamaktır.

#### 3.5. Örnekler

**Örnek 1.**  $\frac{x}{5} = 4$ . İki tarafı 5 ile çarp:

$$x = 20$$

**Örnek 2.**  $\frac{20}{x} = 4$  ( $x \neq 0$ ). İki tarafı  $x$  ile çarp:

$$20 = 4x \Rightarrow x = 5$$

**Örnek 3 (çapraz çarpım).**  $\frac{x+2}{3} = \frac{x-4}{2}$ :

$$2(x+2) = 3(x-4) \Rightarrow 2x+4 = 3x-12 \Rightarrow x = 16$$

Kontrol:  $\frac{16+2}{3} = 6$ ,  $\frac{16-4}{2} = 6$ . ✓

**Örnek 4 (ortak payda).**  $\frac{x}{2} + \frac{x}{4} = 9$ . Ortak payda 4; iki tarafı 4 ile çarp:

$$2x + x = 36 \Rightarrow 3x = 36 \Rightarrow x = 12$$

 ML köprüsü

Kesir aslında bir **orandır**. Makine öğrenmesinde oranlar her yerde: olasılık  $p = \frac{\text{istenen}}{\text{toplam}}$ , bir vektörü birim uzunluğa indirgerken büyüklüğüne böleriz (normalizasyon), öğrenme oranı (learning rate) küçük bir kesirdir. Paydayı sadeleştirme ve oranlarla rahat çalışma, bu kavramların hepsinin temelinde yatar.

### 3.6. Alıştırmalar

Çöz:

1.  $\frac{x}{5} = 4$
2.  $\frac{18}{x} = 6$
3.  $\frac{x}{3} + 2 = 7$
4.  $\frac{x}{2} + \frac{x}{4} = 9$
5.  $\frac{x+2}{3} = \frac{x-4}{2}$
6.  $\frac{20}{x} = 4$

 Çözümler (önce kendin dene, sonra aç)

1.  $x = 20$
2.  $\frac{18}{x} = 6x \Rightarrow x = 3$
3.  $\frac{x}{3} = 5 \Rightarrow x = 15$
4.  $\times 4 : 2x + x = 36 \Rightarrow 3x = 36 \Rightarrow x = 12$
5. Çapraz:  $2(x+2) = 3(x-4) \Rightarrow 2x+4 = 3x-12 \Rightarrow x = 16$
6.  $20 = 4x \Rightarrow x = 5$

Sonraki ders: Üslü İfadeler.



## 4. Üslü İfadeler

Tekrarlı çarpmanın kuralları — sıfır, negatif ve kesirli üsler

Üs, bir sayıyı kendisiyle kaç kez çarptığımızı gösteren kısa yazımdır. Büyük ve küçük sayılarla, büyümeyle, mesafeyle çalışırken her yerde karşına çıkar — bu yüzden kurallarının mantığını oturtmak önemli.

### 4.1. Üs nedir?

$$a^n = \underbrace{a \cdot a \cdots a}_{n \text{ kez}}$$

Burada  $a$  **taban**,  $n$  **üstür**. Örneğin  $2^3 = 2 \cdot 2 \cdot 2 = 8$ .

Dikkat:  $2^3$ ,  $2 \times 3 = 6$  **değildir**. Üs, çarpan sayısını söyler, çarpanın kendisini değil.

### 4.2. Temel kurallar

$$a^m \cdot a^n = a^{m+n} \quad \frac{a^m}{a^n} = a^{m-n} \quad (a^m)^n = a^{mn}$$

$$(ab)^n = a^n b^n \quad \left(\frac{a}{b}\right)^n = \frac{a^n}{b^n}$$

Mantığı basit: **aynı tabanı çarparsan üsler toplanır** ( $a^2 \cdot a^3 = (aa)(aaa) = a^5$ ), **bölürsen çıkarılır, üssün üssünü alırsan çarpılır**.

### 4.3. Sıfır ve negatif üs

Bunları ezberleme — bölme kuralındaki örüntüden çıkar.  $2$ 'nin kuvvetlerinde her adımda  $2$ 'ye bölersek:

Üs	Değer
$2^3$	8
$2^2$	4 $\div 2$
$2^1$	2 $\div 2$
$2^0$	1 $\div 2$

#### 4. Üslü İfadeler

Üs	Değer
$2^{-1}$	$\frac{1}{2} \quad \div 2$
$2^{-2}$	$\frac{1}{4} \quad \div 2$

Örüntü kendiliğinden iki kuralı veriyor:

$$a^0 = 1 \quad (a \neq 0) \quad a^{-n} = \frac{1}{a^n}$$

Negatif üs, sayıyı negatif yapmaz; **tersini (resiprokalini)** alır.

#### 4.4. Kesirli üs = kök

Payda kök derecesini verir:

$$a^{1/n} = \sqrt[n]{a} \quad a^{m/n} = \sqrt[n]{a^m}$$

Örneğin  $16^{1/2} = \sqrt{16} = 4$  ve  $8^{1/3} = \sqrt[3]{8} = 2$ .

##### ! Altın kural

Üs kuralları yalnızca **çarpma, bölme ve kuvvet** işlemlerinde geçerlidir. Toplama bunları birleştirmiz:

$$a^m \cdot a^n = a^{m+n} \quad \checkmark \quad a^m + a^n \neq a^{m+n}$$

$a^m + a^n$  sadeleşmez, öyle bırakılır.

##### ! İki sinsi hata

- $(a + b)^n \neq a^n + b^n$ . Örneğin  $(x + 3)^2 = x^2 + 6x + 9$ , asla  $x^2 + 9$  değil. (Bunu bir sonraki derste özdeşliklerle göreceğiz.)
- İşaret–parantez farkı:  $-3^2 = -(3^2) = -9$ , ama  $(-3)^2 = 9$ . Parantez yoksa üs sadece sayıya etki eder, eksiye değil.

#### 4.5. Örnekler

**Örnek 1.**  $2^3 \cdot 2^4 = 2^{3+4} = 2^7 = 128$

**Örnek 2.**  $\frac{x^6}{x^2} = x^{6-2} = x^4$

**Örnek 3.**  $(x^3)^2 = x^{3 \cdot 2} = x^6$

**Örnek 4.**  $(2x)^3 = 2^3 x^3 = 8x^3$

**Örnek 5.**  $5^0 = 1$  ve  $2^{-3} = \frac{1}{2^3} = \frac{1}{8}$

**Örnek 6.**  $27^{1/3} = \sqrt[3]{27} = 3$

 ML köprüsü

Üsler makine öğrenmesinin her katmanında: üstel fonksiyon  $e^x$  softmax ve sigmoid'in içinde,  $2^n$  büyümesi algoritma karmaşıklığında, bilimsel gösterim ( $10^{-4}$  gibi) küçük öğrenme oranları ve gradyanlarda. Ayrıca vektör büyüklüğündeki kareler ( $x^2 + y^2$ ) ve bir sonraki dersin konusu logaritma — hepsi üssün dilini konuşur. Logaritma, üssün **tersidir**.

## 4.6. Alıştırmalar

Sadeleştir / hesapla:

1.  $x^3 \cdot x^5$
2.  $\frac{a^7}{a^3}$
3.  $(y^2)^4$
4.  $3^{-2}$
5.  $25^{1/2}$
6.  $(2x^2)^3$
7.  $7^0 + 2^3$
8.  $-3^2$  kaçtır? (parantez yok)

 Çözümler (önce kendin dene, sonra aç)

1.  $x^{3+5} = x^8$
2.  $a^{7-3} = a^4$
3.  $y^{2 \cdot 4} = y^8$
4.  $\frac{1}{3^2} = \frac{1}{9}$
5.  $\sqrt{25} = 5$
6.  $2^3(x^2)^3 = 8x^6$
7.  $1 + 8 = 9$  (çünkü  $7^0 = 1$ )
8.  $-(3^2) = -9$  — parantez olmadığı için üs sadece 3'e etki eder. (Karşılaştır:  $(-3)^2 = 9$ .)

Sonraki ders: Çarpanlara Ayırma.



## 5. Çarpanlara Ayırma

Dağıtmanın tersi — ortak çarpan ve üç terimli ifadeler

Çarpanlara ayırma, bir toplamı **çarpıma** çevirmektir — yani dağıtmanın (parantez açmanın) tersi. Bu beceri ikinci derece denklemleri çözenin, ifadeleri sadeleştirmenin ve köklerini bulmanın temelidir.

### 5.1. Çarpanlara ayırma nedir?

Dağıtma çarpımı toplama açar; çarpanlara ayırma toplamı geri çarpıma toplar:

$$a(b + c) = ab + ac \quad \Leftrightarrow \quad ab + ac = a(b + c)$$

Soldan sağa giderken (dağıtma) parantez açılır; sağdan sola giderken (çarpanlara ayırma) ortak parça dışarı alınır.

### 5.2. Ortak çarpan

İlk bakılacak şey: tüm terimlerde ortak olan en büyük parça var mı? Varsa dışarı al.

$$6x + 9 = 3(2x + 3) \quad x^2 + x = x(x + 1) \quad 4x^2 + 6x = 2x(2x + 3)$$

Kontrol için her zaman geri dağıt:  $3(2x + 3) = 6x + 9$ . ✓

### 5.3. Üç terimli ifade

$x^2 + bx + c$  biçimindeki ifadeyi çarpanlara ayırmanın mantığı, çarpımı açmaktan gelir:

$$(x + p)(x + q) = x^2 + (p + q)x + pq$$

Yani **çarpımı c, toplamı b olan iki sayıyı** ( $p$  ve  $q$ ) ararız.

Örnek:  $x^2 + 5x + 6$  için çarpımı 6, toplamı 5 olan iki sayı  $\rightarrow$  2 ve 3:

## 5. Çarpanlara Ayırma

$$x^2 + 5x + 6 = (x + 2)(x + 3)$$

İşaretili örnek:  $x^2 - x - 6$  için çarpımı  $-6$ , toplamı  $-1 \rightarrow -3$  ve  $+2$ :

$$x^2 - x - 6 = (x - 3)(x + 2)$$

### 5.3.1. İşaretleri hızlı bulma

$c$ işareti	$b$ işareti	Çarpanların işareti
+	+	ikisi de +
+	-	ikisi de -
-	herhangi	<b>zıt</b> işaretli (mutlak değeri büyük olan, $b$ 'nin işaretini alır)

#### ! Altın kural

Çarpanlara ayırma = dağıtmanın tersi. Sonucundan emin olmak için her zaman **geri çarp**:  $(x - 3)(x + 2)$ 'yi açınca  $x^2 - x - 6$  çıkıyorsa doğru ayırmışsın demektir.

#### ⚠ Sadeleştirmede sık hata

Bir kesri sadeleştirirken yalnızca **ortak çarpanı** götürebilirsin, terimi değil:

$$\frac{x + 2}{x} \neq \frac{2}{1}$$

Buradaki  $x$  bir terim, çarpan değil; götüremezsin. Önce pay çarpanlara ayrılır, sonra ortak **çarpan** sadeleşir.

## 5.4. Örnekler

**Örnek 1.**  $10x + 15 = 5(2x + 3)$

**Örnek 2.**  $x^2 + 3x = x(x + 3)$

**Örnek 3.**  $x^2 + 7x + 12$ : çarpım 12, toplam 7  $\rightarrow 3, 4 \rightarrow (x + 3)(x + 4)$

**Örnek 4.**  $x^2 - 5x + 6$ : çarpım 6, toplam  $-5 \rightarrow -2, -3 \rightarrow (x - 2)(x - 3)$

**Örnek 5.**  $x^2 + 2x - 15$ : çarpım  $-15$ , toplam 2  $\rightarrow 5, -3 \rightarrow (x + 5)(x - 3)$

 ML köprüsü

Çarpanlara ayırma, bir ifadeyi yapı taşlarına bölmektir. İkinci derecede bu, fonksiyonun **nerede sıfır olduğunu** (köklerini) bulmak demektir — bir sonraki konularda parabolün eksenini kestiği noktalar. Makine öğrenmesinde de bir fonksiyonun sıfır/minimum noktasını bulmak (örneğin türevi sıfıra eşitleyip çözmek) optimizasyonun kalbidir; aynı “sıfıra eşitle ve çöz” fikri.

## 5.5. Alıştırmalar

Çarpanlara ayır:

1.  $8x + 12$
2.  $x^2 + 5x$
3.  $x^2 + 7x + 12$
4.  $x^2 - 5x + 6$
5.  $x^2 - x - 12$
6.  $x^2 + 2x - 15$

 Çözümler (önce kendin dene, sonra aç)

1. Ortak çarpan 4:  $4(2x + 3)$
2. Ortak çarpan  $x$ :  $x(x + 5)$
3. Çarpım 12, toplam 7  $\rightarrow$  3, 4:  $(x + 3)(x + 4)$
4. Çarpım 6, toplam  $-5 \rightarrow -2, -3$ :  $(x - 2)(x - 3)$
5. Çarpım  $-12$ , toplam  $-1 \rightarrow -4, +3$ :  $(x - 4)(x + 3)$
6. Çarpım  $-15$ , toplam 2  $\rightarrow +5, -3$ :  $(x + 5)(x - 3)$

---

Sonraki ders: Özdeşlikler.



## 6. Özdeşlikler

Üç özel kalıp — hem açmak hem çarpanlara ayırmak için

Özdeşlikler, sürekli karşına çıkan birkaç özel çarpım kalıbıdır. Ezberlemeye değer, çünkü her biri **iki yönde** çalışır: soldan sağa gidersen parantez açarsın (dağıtma), sağdan sola gidersen çarpanlara ayırırsın. Bir önceki dersin doğrudan devamı.

### 6.1. Üç özdeşlik

$$(a + b)^2 = a^2 + 2ab + b^2$$

$$(a - b)^2 = a^2 - 2ab + b^2$$

$$a^2 - b^2 = (a - b)(a + b)$$

İlk ikisi **tam kare**, üçüncüsü **iki kare farkıdır**.

### 6.2. $(a + b)^2$ neden orta terimli?

En sık yapılan hata  $(a + b)^2$ 'yi  $a^2 + b^2$  sanmaktır — ortadaki  $2ab$  unutulur. Neden orada olduğunu kare alanıyla görmek en kalıcı yoldur: kenarı  $a + b$  olan bir karenin alanı dört parçaya bölünür.

Toplam alan:  $a^2 + ab + ab + b^2 = a^2 + 2ab + b^2$ . “ $a^2 + b^2$ ” deseydin, iki turuncu dikdörtgeni — yani  $2ab$ 'yi — atmış olurdun.

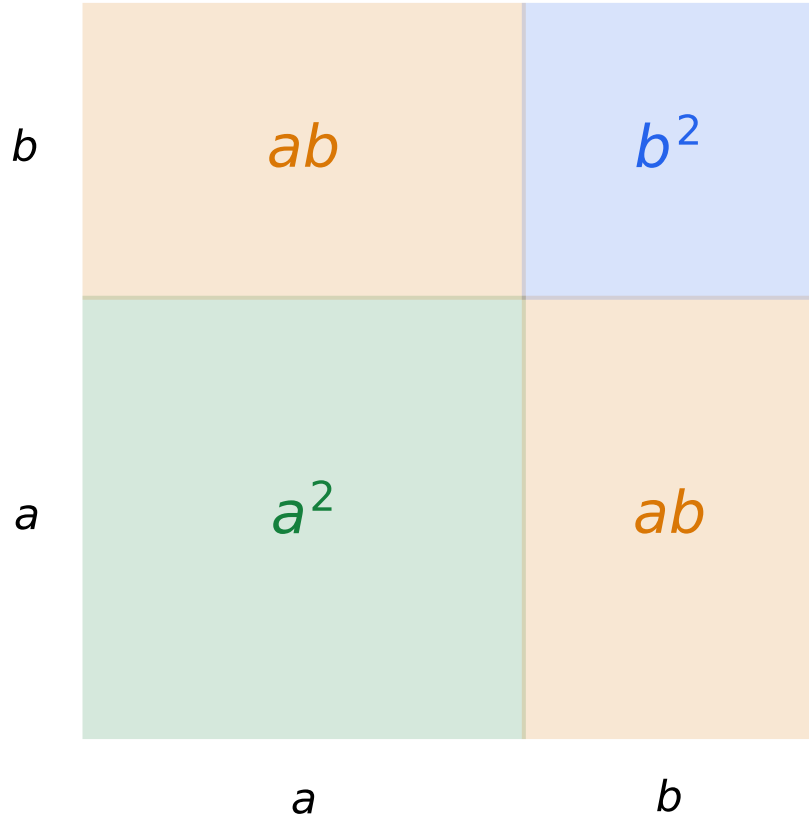
### 6.3. $(a - b)^2$

Aynı kalıp, tek farkla: orta terim **eksidir**.

$$(a - b)^2 = a^2 - 2ab + b^2$$

Örnek:  $(x - 3)^2 = x^2 - 6x + 9$ .

## 6. Özdeşlikler



Şekil 6.1.:  $(a + b)^2$  alan modeli: kenarı  $a + b$  olan karenin alanı dört parçadır. İki turuncu dikdörtgen, sık unutulmuş  $2ab$  terimidir.

## 6.4. İki kare farkı

Bu özdeşlik özellikle çarpanlara ayırmada işe yarar: iki tam karenin **farkı**, toplamlarıyla farklarının çarpımıdır.

$$a^2 - b^2 = (a - b)(a + b)$$

Örnek:  $x^2 - 25 = x^2 - 5^2 = (x - 5)(x + 5)$ .

### ! Altın kural

Üç özdeşlik de iki yönde çalışır — hem açarsın hem çarpanlara ayırırsın. Kalıbı tanı: kare + kare ortada bir terimle mi (tam kare), yoksa kare eksi kare mi (iki kare farkı)?

### ! İki kritik tuzak

- $(a + b)^2 \neq a^2 + b^2$ . Orta terim  $2ab$ 'yi asla unutma.
- İki kare farkı yalnızca **çıkarmada** çalışır:  $a^2 - b^2$  çarpanlara ayrılır, ama  $a^2 + b^2$  (kare **toplamı**) reel sayılarda ayrılmaz.

## 6.5. Örnekler

### Açma:

- $(x + 4)^2 = x^2 + 8x + 16$
- $(x - 3)^2 = x^2 - 6x + 9$
- $(x + 5)(x - 5) = x^2 - 25$

### Çarpanlara ayırma:

- $x^2 - 9 = (x - 3)(x + 3)$
- $x^2 + 10x + 25 = (x + 5)^2$
- $x^2 - 16 = (x - 4)(x + 4)$

### 💡 ML köprüsü

$(a - b)^2$  özdeşliği, makine öğrenmesinin en yaygın kayıp fonksiyonunun tam içindedir. Kare hata (squared error): bir örnek için  $(y - \hat{y})^2$ . Açarsak:

$$(y - \hat{y})^2 = y^2 - 2y\hat{y} + \hat{y}^2$$

Modelin parametrelerine göre türevini (gradyanını) alırken bu açılım doğrudan işine yarar. Yani bu “okul özdeşliği”, MSE kaybının ve gradyan inişinin cebirsel temelidir.

## 6. Özdeşlikler

### 6.6. Alıştırmalar

İlk üçü aç, son üçünü çarpanlara ayır:

1.  $(x + 3)^2$
2.  $(x - 4)^2$
3.  $(x + 6)(x - 6)$
4.  $x^2 - 25$
5.  $x^2 + 8x + 16$
6.  $x^2 - 49$

**i** Çözümler (önce kendin dene, sonra aç)

1.  $x^2 + 6x + 9$
2.  $x^2 - 8x + 16$
3.  $x^2 - 36$
4.  $(x - 5)(x + 5)$
5.  $(x + 4)^2$  — tam kare ( $16 = 4^2$ ,  $8 = 2 \cdot 4$ )
6.  $(x - 7)(x + 7)$

---

Sonraki ders: İkinci Derece Denklemler.

## 7. İkinci Derece Denklemler

İki çözüm yolu — çarpanlara ayırma, formül ve diskriminant

İkinci derece denklem, bilinmeyen en yüksek kuvvetinin 2 olduğu denklemdir. Çözmenin iki yolu var: ifade güzelce çarpanlara ayrılıyorsa hızlı yol, ayrılmıyorsa her zaman çalışan formül. İkisi de aynı kökleri verir.

### 7.1. İkinci derece denklem nedir?

$$ax^2 + bx + c = 0 \quad (a \neq 0)$$

$a, b, c$  katsayılarıdır;  $a$  sıfır olamaz (olsa denklem doğrusal olurdu).

### 7.2. Yol 1 — Çarpanlara ayırma

Bir önceki derslerin doğrudan uygulaması. Dayandığı kural: **bir çarpım sıfırsa, çarpanlardan en az biri sıfırdır.**

$$A \cdot B = 0 \quad \Rightarrow \quad A = 0 \text{ veya } B = 0$$

Önce çarpanlara ayır, sonra her çarpanı sıfıra eşitle:

$$x^2 - x - 6 = 0 \Rightarrow (x - 3)(x + 2) = 0 \Rightarrow x = 3 \text{ veya } x = -2$$

### 7.3. Yol 2 — Formül

İfade çarpanlara ayrılmasa bile çalışır:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Aynı denklem ( $a = 1, b = -1, c = -6$ ):

## 7. İkinci Derece Denklemler

$$x = \frac{-(-1) \pm \sqrt{(-1)^2 - 4(1)(-6)}}{2(1)} = \frac{1 \pm \sqrt{25}}{2} = \frac{1 \pm 5}{2} \Rightarrow x = 3 \text{ veya } x = -2$$

Aynı sonuç — iki yol da doğru.

### 7.4. Diskriminant: kaç kök var?

Formüldeki kök içindeki ifadeye **diskriminant** denir:

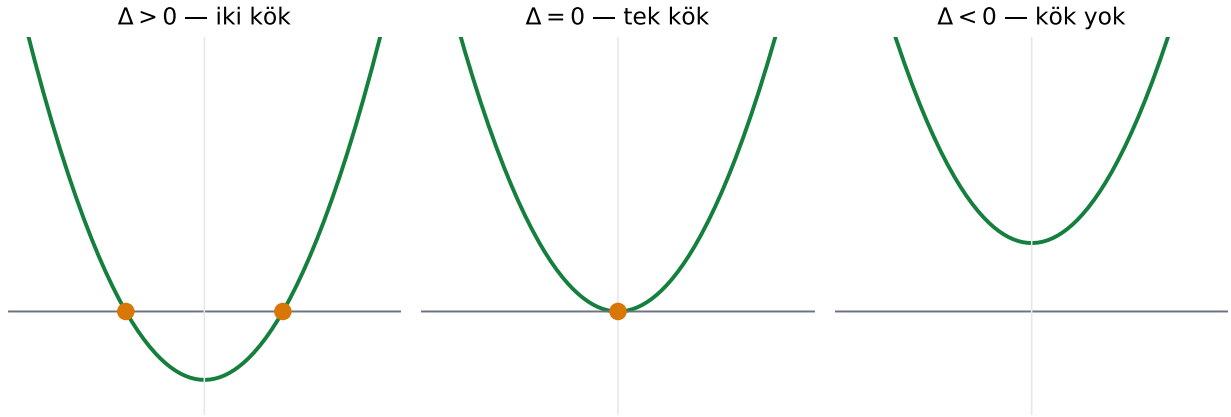
$$\Delta = b^2 - 4ac$$

İşareti, denklemin kaç reel kökü olduğunu söyler:

- $\Delta > 0 \rightarrow$  iki farklı reel kök
- $\Delta = 0 \rightarrow$  tek (çift) kök
- $\Delta < 0 \rightarrow$  reel kök yok (kökler karmaşık)

### 7.5. Geometrik anlam: parabolün kökleri

Kökler aslında  $y = ax^2 + bx + c$  parabolünün **x eksenini kestiği noktalar**dır. Diskriminant da bu kesişimin kaç kez olduğunu söyler:



Şekil 7.1.: Diskriminant ve parabol:  $\Delta > 0$  iki kesişim (iki kök),  $\Delta = 0$  teğet (tek kök),  $\Delta < 0$  kesişim yok (reel kök yok).

Üç parabol de aynı  $y = x^2 + c$  ailesinden, sadece yukarı/aşağı kaymış: dibi eksenin altındaysa iki kez keser, tam değiyorsa bir kez, üstündeyse hiç kesmez.

! Hangi yolu, ne zaman?

Çarpanlara kolay ayrılıyorsa o hızlıdır; ayrılmıyorsa formül her zaman çalışır; sadece kaç kök olduğunu merak ediyorsan diskriminantta bakman yeter. **Her durumda önce denklemini = 0 formuna getir** —  $x^2 = 5x - 6$  ise önce  $x^2 - 5x + 6 = 0$  yaz.

! Formülde işaret tuzağı

- $-b$ :  $b$  negatifse  $-b$  **pozitif** olur.  $b = -4$  ise  $-b = +4$ .
- $b^2$ : negatif olamaz ( $b = 0$ 'da sıfır, yani  $\geq 0$ ),  $b$  negatif olsa bile. Diskriminantta ve formülde en çok hata bu iki noktada yapılır.

## 7.6. Örnekler

**Örnek 1 (çarpanlara ayırma).**  $x^2 + 7x + 12 = 0 \Rightarrow (x + 3)(x + 4) = 0 \Rightarrow x = -3$  veya  $-4$

**Örnek 2 (formül).**  $x^2 - 4x + 3 = 0$ :  $\Delta = 16 - 12 = 4$ ,  $x = \frac{4 \pm 2}{2} \Rightarrow x = 3$  veya  $1$

**Örnek 3 (diskriminant).**  $x^2 + 2x + 5 = 0$ :  $\Delta = 4 - 20 = -16 < 0 \Rightarrow$  reel kök yok.

💡 ML köprüsü

Tek parametrelili kare hata kaybı bir **paraboldür**; en iyi parametre, bu parabolün en alçak noktasıdır (tepe noktası). Gradyan inişi aslında o dibi arar. İkinci derecenin geometrisi — parabol, kökler, tepe noktası — optimizasyonun görsel temelidir. “Türevi sıfıra eşitle ve çöz” fikri de buradaki “sıfıra eşitle ve çöz” ile aynı.

## 7.7. Alıştırmalar

1. Çarpanlara ayırarak çöz:  $x^2 + 5x + 6 = 0$
2. Çarpanlara ayırarak çöz:  $x^2 - x - 6 = 0$
3. Formülle çöz:  $x^2 - 4x + 3 = 0$
4. Formülle çöz:  $x^2 + 6x + 9 = 0$
5. Kaç reel kök (sadece diskriminant):  $2x^2 + 3x + 5 = 0$
6. Kaç reel kök (sadece diskriminant):  $x^2 - 6x + 9 = 0$

i Çözümler (önce kendin dene, sonra aç)

1.  $(x + 2)(x + 3) = 0 \Rightarrow x = -2$  veya  $-3$
2.  $(x - 3)(x + 2) = 0 \Rightarrow x = 3$  veya  $-2$
3.  $\Delta = 4$ ,  $x = \frac{4 \pm 2}{2} \Rightarrow x = 3$  veya  $1$
4.  $\Delta = 36 - 36 = 0$ ,  $x = \frac{-6}{2} = -3$  (çift kök;  $(x + 3)^2 = 0$ )

## 7. İkinci Derece Denklemler

$$5. \Delta = 9 - 40 = -31 < 0 \Rightarrow \text{reel kök yok}$$

$$6. \Delta = 36 - 36 = 0 \Rightarrow \text{tek (çift) kök, } x = 3$$

---

*Sonraki ders: Eşitsizlikler.*

## 8. Eşitsizlikler

Bir aralık çözmek — yön kuralı, sayı doğrusu, aralık notasyonu

Eşitsizlik, eşittir yerine bir **karşılaştırmadır**: bir taraf diğerinden küçük ya da büyüktür. Denklemden tek büyük farkı, çözümün tek bir sayı değil, bir **aralık** (sonsuz çözüm) olmasıdır.

### 8.1. Eşitsizlik nedir?

Dört sembol:  $<$  (küçük),  $>$  (büyük),  $\leq$  (küçük veya eşit),  $\geq$  (büyük veya eşit). Örneğin  $x > 3$ , “3’ten büyük tüm sayılar” demektir — 3,1, 4, 100... hepsi çözümdür.

### 8.2. Çözme: tek kritik fark

Eşitsizliği denklem gibi çözersin — terimleri taşı, böl. Tek ek kural şu:

! Altın kural

**Negatif bir sayıyla çarpar veya bölersen, eşitsizliğin yönü değişir.**

$$-2x < 6 \xrightarrow{\div(-2)} x > -3$$

$<$  işareti  $>$  oldu, çünkü  $-2$ 'ye böldük. Toplama/çıkarmada ve **pozitif**le çarpma/bölmede yön sabit kalır; sadece **negatif**te döner.

Yön değişmeyen örnek (pozitifle bölme):

$$3x - 5 \leq 7 \Rightarrow 3x \leq 12 \Rightarrow x \leq 4$$

### 8.3. Çözümü gösterme — sayı doğrusu

Bir eşitsizliğin çözümünü sayı doğrusunda taranmış bir aralık olarak gösteririz. Uç noktanın dahil olup olmadığını dairenin türü söyler:

İki kural: **açık uç** = sınır dahil değil ( $<$ ,  $>$  için), **kapalı uç** = sınır dahil ( $\leq$ ,  $\geq$  için).

## 8. Eşitsizlikler



Şekil 8.1.:  $-3 < x \leq 5$  çözümü:  $-3$ 'te açık uç (dahil değil),  $5$ 'te kapalı uç (dahil), arası taralı.

### 8.4. Aralık notasyonu

Çözümü kısa yazmanın yolu. Parantez ( dahil değil, köşeli parantez [ dahil:

$$x > -3 \rightarrow (-3, \infty) \quad -3 < x \leq 5 \rightarrow (-3, 5]$$

Sonsuz ( $\infty$ ) **her zaman parantez** alır, asla köşeli — sonsuza “ulaşamadığın” için.

### 8.5. Bileşik eşitsizlik

$x$  iki değer arasındaysa, **her parçaya** aynı işlemi yaparak çözersin:

$$-1 < 2x + 1 \leq 7 \xrightarrow{-1} -2 < 2x \leq 6 \xrightarrow{\div 2} -1 < x \leq 3$$

#### ⚠ İki tuzak

- Negatifle çarp/böldüğünde yön çevirmeyi unutmak — en sık kaçırılan kural.
- Sonsuza köşeli parantez koymak — yanlış;  $\infty$  ve  $-\infty$  her zaman parantez alır.

### 8.6. Örnekler

**Örnek 1.**  $2x + 3 > 11 \Rightarrow 2x > 8 \Rightarrow x > 4$ , yani  $(4, \infty)$

**Örnek 2 (yön döner).**  $-3x \leq 9 \xrightarrow{\div(-3)} x \geq -3$ , yani  $[-3, \infty)$

**Örnek 3 (yön döner).**  $5 - 2x < 1 \Rightarrow -2x < -4 \xrightarrow{\div(-2)} x > 2$ , yani  $(2, \infty)$

#### 💡 ML köprüsü

Eşitsizlik makine öğrenmesinde her yerde. Olasılık  $0 \leq p \leq 1$  aralığındadır. En yaygın aktivasyon ReLU,  $\max(0, x)$  yani “ $x > 0$  ise  $x$ , değilse 0” — bir eşitsizlik kararı. Karar eşikleri (skor  $> 0,5$  ise sınıf 1), optimizasyon kısıtları ve gradyan kırma (gradient clipping) hep eşitsizliktir.

## 8.7. Alıřtırmalar

Çöz ve aralık notasyonuyla yaz:

1.  $2x + 3 > 11$
2.  $-3x \leq 9$  (dikkat — yön!)
3.  $5 - 2x < 1$  (dikkat — yön!)
4.  $x \geq 2$  (sadece aralık notasyonu)
5.  $-1 < x + 3 \leq 4$  (bileşik)
6.  $\frac{x}{2} \geq 3$

**i** Çözümler (önce kendin dene, sonra aç)

1.  $2x > 8 \Rightarrow x > 4 \Rightarrow (4, \infty)$
2.  $\div(-3)$ , yön döner:  $x \geq -3 \Rightarrow [-3, \infty)$
3.  $-2x < -4$ ,  $\div(-2)$  yön döner:  $x > 2 \Rightarrow (2, \infty)$
4.  $[2, \infty)$
5. Her parçadan 3 çıkar:  $-4 < x \leq 1 \Rightarrow (-4, 1]$
6.  $x \geq 6 \Rightarrow [6, \infty)$

---

Sonraki ders: Logaritma.



## 9. Logaritma

Üssün tersi — çarpımı toplama çeviren işlem

Logaritma, üs almanın **tersidir**. Üs “bu tabanı şu kadar kez çarparsam ne olur?” diye sorar; logaritma ters yönden sorar: “**bu sayıyı elde etmek için tabanı kaçınıcı kuvvete yükseltmeliyim?**” Bu ters bakış, makine öğrenmesinin en kritik araçlarından biridir (log-loss, olabilirlik).

### 9.1. Logaritma nedir?

Tanım iki yönde okunur:

$$\log_b(x) = y \iff b^y = x$$

Örneğin  $\log_2(8) = 3$ , çünkü  $2^3 = 8$ . Yani  $\log_2 8$  sorusu aslında “2’yi kaçınıcı kuvvete yükseltirsem 8 olur?” demektir.

Üs tablosundan okumak en kolayı:

Üs ifadesi	Logaritma ifadesi
$2^3 = 8$	$\log_2 8 = 3$
$10^2 = 100$	$\log_{10} 100 = 2$
$5^0 = 1$	$\log_5 1 = 0$
$3^1 = 3$	$\log_3 3 = 1$

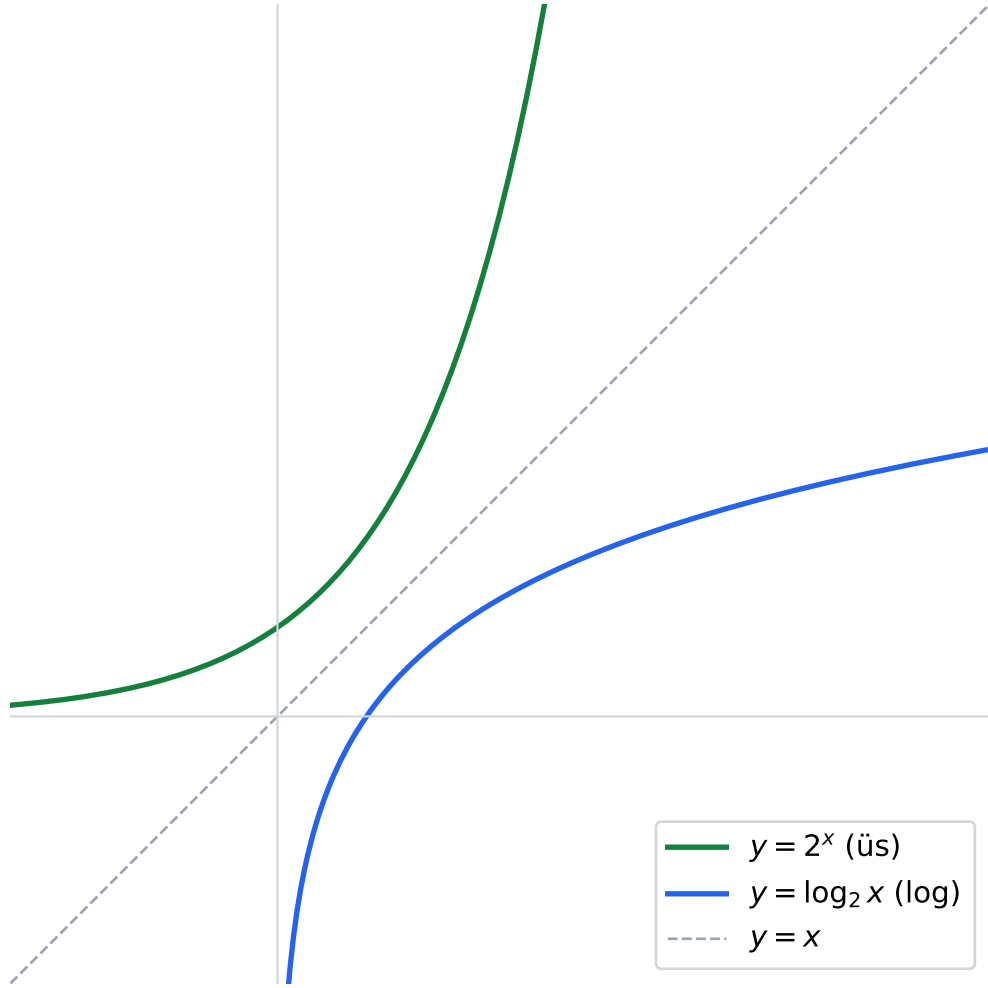
### 9.2. Üs ve logaritma: ayna görüntüsü

İkisi birbirinin tersi olduğu için grafikleri  $y = x$  doğrusuna göre **simetrik** — biri diğerinin aynadaki yansımasıdır:

### 9.3. Yaygın tabanlar

- **Taban 10** —  $\log_{10}$ , çoğu yerde sadece log yazılır.
- **Taban  $e$**  —  $\log_e$  yerine  $\ln$  yazılır (doğal logaritma).  $e \approx 2,718$ . Kalkülüs ve ML’de en sık bu kullanılır.
- **Taban 2** —  $\log_2$ , bilgisayar bilimi ve bilgi teorisinde.

## 9. Logaritma



Şekil 9.1.: Üs ve logaritma birbirinin tersidir:  $y = 2^x$  ile  $y = \log_2 x$ ,  $y = x$  doğrusuna göre simetriktir.

## 9.4. Logaritma kuralları

Üssün tersi olduğu için kurallar da üs kurallarının aynası: üste **toplana**n şey, logaritmada **çarpıma**; üste çarpılan, logaritmada **toplama** karşılık gelir.

$$\log_b(xy) = \log_b x + \log_b y \quad \log_b\left(\frac{x}{y}\right) = \log_b x - \log_b y$$

$$\log_b(x^n) = n \log_b x \quad \log_b 1 = 0 \quad \log_b b = 1$$

! En işe yarayan özellik

Logaritma **çarpımı toplamaya, kuvveti çarpmaya** çevirir:

$$\log(xy) = \log x + \log y \quad \log(x^n) = n \log x$$

Bu, devasa çarpımları ve kuvvetleri sıradan toplama/çarpmaya indirger — ML'de olabirlik hesabının ve sayısal kararlılığın anahtarı.

! İki tuzak

- Toplama kuralı **çarpım** içindir, toplam için değil:  $\log(x + y) \neq \log x + \log y$ .
- Logaritma yalnızca **pozitif** sayılar için tanımlıdır:  $\log 0$  ve negatif sayının logaritması yoktur.

## 9.5. Örnekler

**Örnek 1.**  $\log_2 16 = 4$ , çünkü  $2^4 = 16$ .

**Örnek 2.**  $\log_{10} 1000 = 3$ , çünkü  $10^3 = 1000$ .

**Örnek 3 (ürün kuralı).**  $\log_2(4 \cdot 8) = \log_2 4 + \log_2 8 = 2 + 3 = 5$  (kontrol:  $4 \cdot 8 = 32 = 2^5$ ). ✓

**Örnek 4 (kuvvet kuralı).**  $\log_2(8^2) = 2 \log_2 8 = 2 \cdot 3 = 6$  (kontrol:  $8^2 = 64 = 2^6$ ). ✓

💡 ML köprüsü

Bir modelin olabirliği (likelihood), yüzlerce olasılığın **çarpımıdır** — her biri 1'den küçük olduğu için bu çarpım sayısal olarak sıfıra çöker (underflow). Logaritma alınca çarpım **toplama** dönüşür:

$$\log(p_1 p_2 \cdots p_n) = \log p_1 + \log p_2 + \cdots + \log p_n$$

İşte bu yüzden ML modelleri log-olabirliği (log-likelihood) maksimize eder, çapraz entropi / log-loss kaybı —  $-\log(p)$  kullanır. Bu dersteki ürün ve kuvvet kuralları, o kaybın cebirsel temelidir.

## 9. Logaritma

### 9.6. Alıştırılmalar

1.  $\log_2 16$
2.  $\log_{10} 100$
3.  $\log_7 1$
4.  $\log_5 5$
5. Ürün kuralıyla:  $\log_2(8 \cdot 2)$
6. Kuvvet kuralıyla:  $\log_3(9^2)$

**i** Çözümler (önce kendin dene, sonra aç)

1. 4 (çünkü  $2^4 = 16$ )
2. 2 (çünkü  $10^2 = 100$ )
3. 0 (her tabanın 0. kuvveti 1'dir)
4. 1 (her tabanın 1. kuvveti kendisidir)
5.  $\log_2 8 + \log_2 2 = 3 + 1 = 4$  (kontrol:  $16 = 2^4$ )
6.  $2\log_3 9 = 2 \cdot 2 = 4$  (kontrol:  $81 = 3^4$ )

---

*Bölüm 1 — Cebir tamamlandı. Sonraki bölüm: Fonksiyonlar.*

**Kısım II.**

**Bölüm 2 — Fonksiyonlar**



## 10. Fonksiyon Kavramı

Girdiyi çıktıya bağlayan kural — ML modellerinin temel taşı

Bölüm 1’de denklemleri çözdük. Şimdi bir adım yukarı çıkıyoruz: **fonksiyon**. Fonksiyon, bir girdiyi bir çıktıya bağlayan kuraldır. Bu kavram tüm bölümün — ve aslında makine öğrenmesinin — temelidir: bir model, sonuçta bir fonksiyondur.

### 10.1. Fonksiyon nedir?

Fonksiyonu bir **makine** gibi düşün: içine bir girdi koyarsın, kuralını uygular, sana tek bir çıktı verir.



Şekil 10.1.: Fonksiyon bir makinedir: bir girdi alır ( $x$ ), kuralı uygular ( $f$ ), tek bir çıktı verir ( $f(x)$ ).

Fonksiyonun tek belirleyici kuralı şudur: **her girdiye karşılık tam olarak bir çıktı** vardır. Aynı girdi her seferinde aynı çıktıyı vermelidir.

### 10.2. $f(x)$ gösterimi

$f(x)$ , “ $f$  fonksiyonunun  $x$  girdisindeki değeri” demektir —  $f$  çarpı  $x$  değil. Kural genelde bir formülle yazılır:

$$f(x) = 2x + 1$$

## 10. Fonksiyon Kavramı

Bir değeri hesaplamak için  $x$  yerine sayıyı koyarsın:

$$f(3) = 2(3) + 1 = 7 \quad f(0) = 1 \quad f(-2) = -3$$

### 10.3. Tanım ve görüntü kümesi

- **Tanım kümesi (domain):** fonksiyona girebilecek tüm geçerli girdiler.
- **Görüntü kümesi (range):** çıkabilecek tüm olası çıktılar.

Bazı kurallar her girdiyi kabul etmez:

$$f(x) = \sqrt{x} \Rightarrow \text{tanım kümesi } x \geq 0 \quad h(x) = \frac{1}{x-2} \Rightarrow x \neq 2$$

Çünkü negatife karekökü ve sıfıra bölme tanımsızdır.

! Fonksiyonu fonksiyon yapan kural

Her girdi **tam olarak bir** çıktıya gider. Bir girdi iki farklı çıktı verebiliyorsa, o bir fonksiyon değildir. (Grafikte: dikey bir doğru eğriyi en fazla bir noktada kesmeli — dikey doğru testi.)

! İki tuzak

- $f(x)$  bir çarpım değildir; “ $x$ ’teki  $f$  değeri” demektir.
- Tanım kümesini kontrol et: paydada sıfır olamaz, kök içinde negatif olamaz.

### 10.4. Örnekler

**Örnek 1.**  $f(x) = 2x + 1$  için  $f(5) = 2(5) + 1 = 11$ .

**Örnek 2.**  $g(x) = x^2$  için  $g(-3) = (-3)^2 = 9$ .

**Örnek 3 (tanım kümesi).**  $f(x) = \frac{1}{x}$  için tanım kümesi  $x \neq 0$ .

💡 ML köprüsü

Bir makine öğrenmesi modeli, **fonksiyonun ta kendisidir**: girdi olarak özellikleri ( $x$ ) alır, çıktı olarak tahmini ( $f(x)$ ) verir. Daha önce gördüğümüz  $y = wx + b$  bir fonksiyondur; bir sinir ağı ise iç içe geçmiş dev bir fonksiyon bileşkesidir (bunu ileride göreceğiz). Bir olasılık çıktısının görüntü kümesi  $[0, 1]$ 'dir — Bölüm 1'deki eşitsizlik aralığıyla aynı fikir.

**10.5. Alıřtırmalar**

1.  $f(x) = 3x - 2$  için  $f(4) = ?$
2.  $f(x) = x^2 + 1$  için  $f(-3) = ?$
3.  $g(x) = 2x + 5$  için  $g(0) = ?$
4.  $f(x) = \sqrt{x}$  fonksiyonunun tanım kümesi nedir? (aralık notasyonuyladır)
5.  $f(x) = \frac{1}{x}$  fonksiyonunun tanım kümesi nedir?
6.  $f(x) = 5 - 2x$  için  $f(3) = ?$

**i** Çözümler (önce kendin dene, sonra aç)

1.  $3(4) - 2 = 10$
2.  $(-3)^2 + 1 = 9 + 1 = 10$
3.  $2(0) + 5 = 5$
4.  $x \geq 0$ , yani  $[0, \infty)$  (kareköke negatif giremez)
5.  $x \neq 0$  (sıfıra bölme tanımsız)
6.  $5 - 2(3) = 5 - 6 = -1$

---

*Sonraki ders: Doğrusal Fonksiyon ve Eğim.*



# 11. Doğrusal Fonksiyon ve Eğim

$y = mx + b$  — ML'in kalbindeki doğru

En basit ve en önemli fonksiyon biçimi doğrudur. Grafiği düz bir çizgi olan fonksiyona **doğrusal fonksiyon** denir. Bu dersteki  $y = mx + b$  kalıbı, makine öğrenmesinin tam merkezinde duruyor — lineer regresyon ve tek bir yapay nöron, aynen bu denklemdir.

## 11.1. Doğrusal fonksiyon: $y = mx + b$

$$f(x) = mx + b$$

İki sayı doğruyu tamamen belirler:

- $m$  — **eğim (slope)**: doğru ne kadar dik, hangi yöne gidiyor.
- $b$  —  **$y$ -kesişimi (intercept)**: doğrunun  $y$  eksenini kestiği yer, yani  $f(0)$ .

## 11.2. Eğim

Eğim, “ne kadar yukarı / ne kadar sağa” oranıdır — dikey değişimin yatay değişime bölümü:

$$m = \frac{\Delta y}{\Delta x} = \frac{y_2 - y_1}{x_2 - x_1}$$

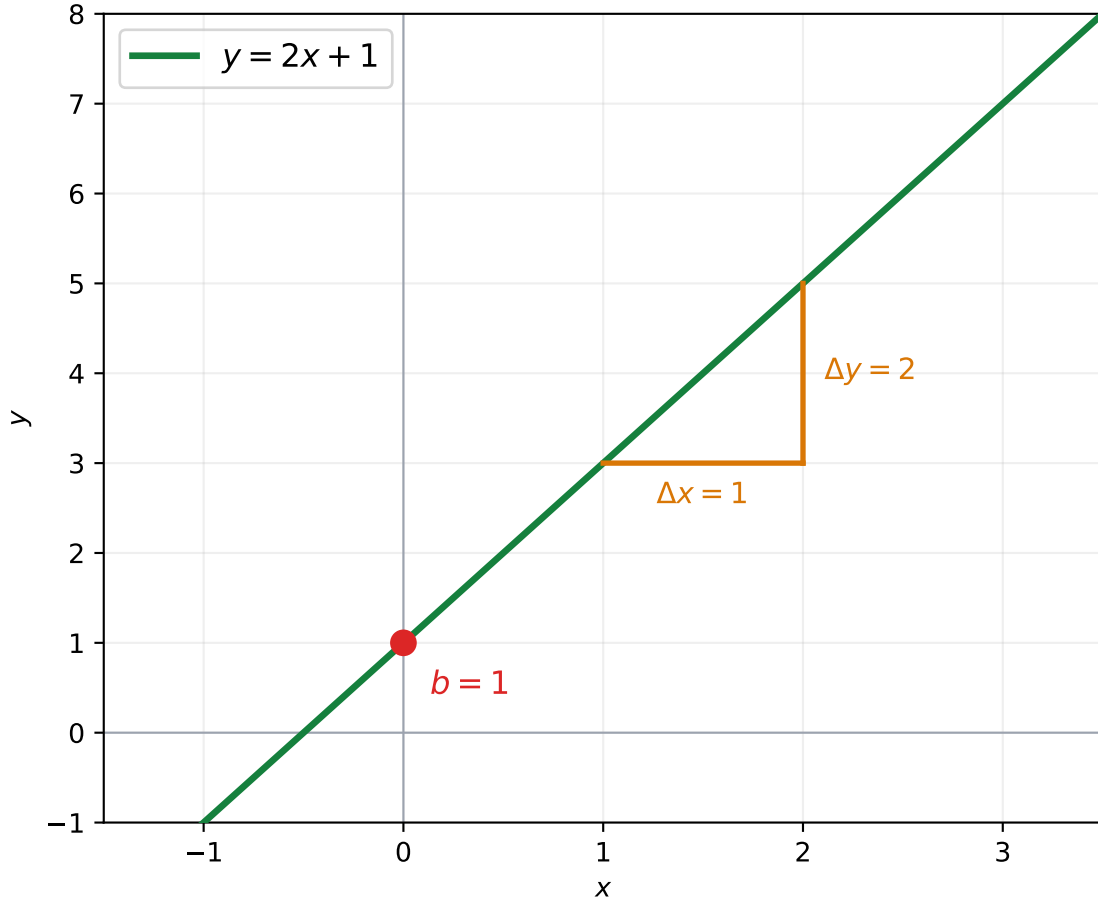
Eğimin işareti doğrunun yönünü söyler:

- $m > 0$  → doğru yukarı çıkar.
- $m < 0$  → doğru aşağı iner.
- $m = 0$  → yatay doğru.
- $|m|$  büyüdükçe doğru dikleşir.

## 11.3. $y$ -kesişimi

$b$ , doğrunun  $y$  eksenini deldiği noktadır;  $x = 0$  koyarak bulunur:  $f(0) = m \cdot 0 + b = b$ . Yukarıdaki örnekte  $f(0) = 1$ , yani doğru  $(0, 1)$ 'den geçer.

11. Doğrusal Fonksiyon ve Eğim



Şekil 11.1.:  $y = 2x + 1$  doğrusu: eğim  $m = 2$  (1 sağa gidince 2 yukarı),  $y$ -kesişimi  $b = 1$ , yani  $(0, 1)$  noktası.

! İki sayı, bir doğru

$m$  doğrunun **eğimini ve yönünü**,  $b$  ise **dikey konumunu** belirler. Bu ikisi verildiğinde doğru tek bir şekilde sabitlenir — başka hiçbir bilgiye gerek yoktur.

## 11.4. İki noktadan eğim

(1, 2) ve (3, 8) noktalarından geçen doğrunun eğimi:

$$m = \frac{8 - 2}{3 - 1} = \frac{6}{2} = 3$$

! İki tuzak

- Eğim  $\frac{\Delta y}{\Delta x}$ 'tir,  $\frac{\Delta x}{\Delta y}$  değil — dikey önce, yatay sonra.
- Noktaların sırasını ikisinde de aynı tut:  $\frac{y_2 - y_1}{x_2 - x_1}$ . Birini ters çevirirsen işaret yanlış çıkar.

## 11.5. Örnekler

**Örnek 1.**  $y = 3x - 2$ : eğim  $m = 3$ , kesişim  $b = -2$ .

**Örnek 2.**  $y = -2x + 5$ : eğim  $m = -2$  (aşağı iner), kesişim  $b = 5$ .

**Örnek 3 (iki nokta).** (0, 4) ve (2, 0):  $m = \frac{0 - 4}{2 - 0} = -2$ .

💡 ML köprüsü

$y = mx + b$  denklemini ML notasyonu ile yaz:  $y = wx + b$ . Bu, **lineer regresyonun ve tek bir yapay nöronun** tam tanımıdır.

- $w$  (ağırlık / weight) = eğim  $m \rightarrow$  modelin **öğrendiği** şey.
- $b$  (bias) =  $y$ -kesişimi  $\rightarrow$  çıktığı yukarı/aşağı kaydırıcı sabit.

Modeli eğitmek, veriye en iyi uyan  $w$  ve  $b$  değerlerini bulmaktır; gradyan inişi tam olarak bu eğimi ve kesişimi ayarlar. Bir sinir ağı ise üst üste yığılmış binlerce böyle doğrudur (daha doğrusu: aktivasyon — doğrusal-olmayanlık — olmadan bu bileşke tek bir doğruya çökerdi; ağı bir doğrudan fazlası yapan, araya giren aktivasyonlardır). Bu tek satır, tüm derin öğrenmenin tohumudur.

### 11.6. Alıştırmalar

1.  $y = 4x + 3$  doğrusunun eğimi ve  $y$ -kesişi nedir?
2.  $(2, 5)$  ve  $(4, 11)$  noktalarından geçen doğrunun eğimi?
3.  $y = -x + 7$  doğrusunun eğimi nedir?
4. Eğimi 2,  $y$ -kesişi  $-1$  olan doğrunun denklemi?
5.  $(0, 4)$  ve  $(2, 0)$  noktalarından geçen doğrunun eğimi?
6.  $f(x) = 5x$  doğrusunun  $y$ -kesişi nedir?

**i** Çözümler (önce kendin dene, sonra aç)

1.  $m = 4, b = 3$
2.  $\frac{11 - 5}{4 - 2} = \frac{6}{2} = 3$
3.  $m = -1$  (çünkü  $y = -1 \cdot x + 7$ )
4.  $y = 2x - 1$
5.  $\frac{0 - 4}{2 - 0} = \frac{-4}{2} = -2$
6.  $b = 0$  (çünkü  $y = 5x + 0$ ; doğru orijinden geçer)

---

Sonraki ders: Parabol.

## 12. Parabol

İkinci derece fonksiyonun U biçimi — ML'deki kayıp çukuru

İkinci derece bir fonksiyonun grafiği bir **parabol**dür — U biçimli bir eğri. Bölüm 1'de ikinci derece denklemleri çözmüştük; şimdi o denklemin grafiğine bakıyoruz. Bu şekil ML'de doğrudan karşına çıkar: kayıp fonksiyonu (loss) çoğu zaman tam böyle bir çukurdur ve eğitim, o çukurun dibini aramaktır.

### 12.1. Parabol: $y = ax^2 + bx + c$

$$f(x) = ax^2 + bx + c \quad (a \neq 0)$$

### 12.2. $a$ kolların yönünü ve genişliğini belirler

- $a > 0 \rightarrow$  kollar **yukarı** açılır, parabolün bir **minimumu** vardır (çukur).
- $a < 0 \rightarrow$  kollar **aşağı** açılır, bir **maksimumu** vardır (tepe).
- $|a|$  büyüdükçe parabol **darlaşır**; küçüldükçe yayvanlaşır.

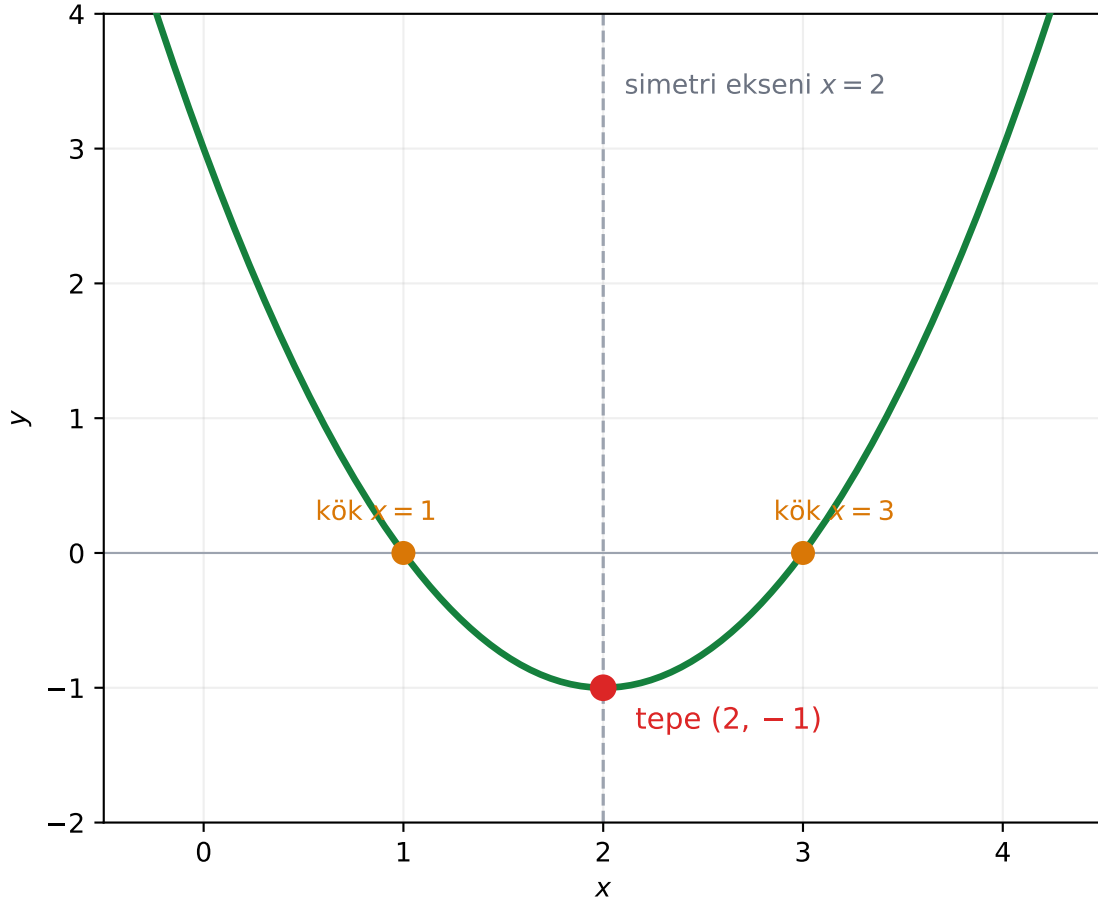
### 12.3. Tepe noktası

Tepe noktası, parabolün döndüğü yerdir — minimum ya da maksimum.  $x$  koordinatı:

$$x_{\text{tepe}} = -\frac{b}{2a}$$

Bu  $x$ 'i fonksiyona koyunca tepe noktasının  $y$  değeri çıkar.  $a > 0$  ise tepe en alçak nokta (minimum),  $a < 0$  ise en yüksek nokta (maksimum).

12. Parabol



Şekil 12.1.:  $y = x^2 - 4x + 3$  parabolü: kollar yukarı ( $a > 0$ ), tepe noktası  $(2, -1)$ , kökler  $x = 1$  ve  $x = 3$ , simetri eksenini  $x = 2$ .

## 12.4. Kökler ve diskriminant

Parabolün  $x$  eksenini kestiği yerler **köklerdir** ( $y = 0$ ). Bunlar Bölüm 1'deki ikinci derece denklemin çözümleridir; kaç tane olduğunu diskriminant  $\Delta = b^2 - 4ac$  söyler:

- $\Delta > 0 \rightarrow$  iki kök (parabol eksenini iki noktada keser),
- $\Delta = 0 \rightarrow$  bir kök (tepe tam eksenin üstünde),
- $\Delta < 0 \rightarrow$  reel kök yok (parabol eksenini hiç kesmez).

### ! Önemli olan ne?

$a$  parabolün **yönünü ve genişliğini**, tepe noktası ise **dönüm yerini** (min/max) verir. Optimizasyonda asıl peşinde olduğumuz şey bu tepe noktasıdır — fonksiyonun en küçük (veya en büyük) değeri orada.

## 12.5. Örnek

$f(x) = x^2 - 4x + 3$  parabolü:

- $a = 1 > 0 \rightarrow$  kollar yukarı, minimum var.
- Tepe:  $x = -\frac{-4}{2(1)} = 2$ , sonra  $f(2) = 4 - 8 + 3 = -1 \rightarrow$  tepe  $(2, -1)$ .
- Kökler:  $x^2 - 4x + 3 = (x - 1)(x - 3) = 0 \Rightarrow x = 1, x = 3$ .
- $y$ -kesişi:  $c = 3$ .

### ! İki tuzak

- Tepe formülü  $-\frac{b}{2a}$ 'dir; baştaki eksi işaretini unutma.
- Her parabol  $x$  eksenini kesmez ( $\Delta < 0$  ise kök yoktur) — ama yine de bir tepe noktası vardır. “Kök yok” demek “tepe yok” demek değildir.

### 💡 ML köprüsü

Bir modeli eğitirken, hatayı ölçen bir **kayıp fonksiyonu** (loss) tanımlarız — en yaygını ortalama kare hata (MSE). Tek bir ağırlığın fonksiyonu olarak çizildiğinde MSE tam bir **paraboldür**: yukarı açılan bir çukur. Eğitim, bu çukurun **dibine** (tepe noktası = minimum kayıp) inmektir; gradyan inişi adım adım dibe yuvarlanır.

$a > 0$  olan çukur biçimi “konveks” demektir — tek bir minimum vardır, bu yüzden optimize etmesi kolaydır. Parabolün geometrisi, modelin nasıl öğrendiğinin doğrudan resmidir.

## 12.6. Alıştırmalar

1.  $y = x^2 - 6x + 5$  parabolü yukarı mı aşağı mı açılır?
2.  $y = -2x^2 + 4x$  parabolü tepe noktasının  $x$  koordinatı nedir?

## 12. Parabol

3.  $y = x^2 - 4$  parabolü kökleri nelerdir?
4.  $y = 3x^2 + 2x - 1$  parabolü  $y$ -kesişimi nedir?
5.  $y = x^2 - 6x + 5$  parabolü tepe noktası ( $x$  ve  $y$ )?
6.  $y = -x^2 + 1$  parabolü maksimum mu minimum mu vardır?

### **i** Çözümler (önce kendin dene, sonra aç)

1. Yukarı ( $a = 1 > 0$ ).
2.  $x = -\frac{4}{2(-2)} = -\frac{4}{-4} = 1$ .
3.  $x^2 - 4 = 0 \Rightarrow x^2 = 4 \Rightarrow x = 2$  veya  $x = -2$ .
4.  $c = -1$ .
5.  $x = -\frac{-6}{2(1)} = 3$ , sonra  $f(3) = 9 - 18 + 5 = -4 \rightarrow$  tepe  $(3, -4)$ .
6. Maksimum ( $a = -1 < 0$ , kollar aşağı).

---

Sonraki ders: Üstel ve Logaritmik Fonksiyon.

## 13. Üstel ve Logaritmik Fonksiyon

Büyüme, çürüme ve onların tersi — sigmoid'in temeli

Bölüm 1'de üs (Üslü İfadeler dersi) ve logaritmanın (Logaritma dersi) cebirini gördük. Şimdi bunları **fonksiyon** olarak ele alıp grafiklerine bakıyoruz. Bu iki şekil modern ML'in motorudur: sigmoid ve softmax doğrudan üstel fonksiyona, kayıp fonksiyonu ise logaritmaya dayanır.

### 13.1. Üstel fonksiyon: $f(x) = b^x$

Değişkenin **üste** çıktığı fonksiyondur (taban sabit, üs değişken):

$$f(x) = b^x \quad (b > 0, b \neq 1)$$

Davranışı tabanın değerine bağlıdır:

- $b > 1 \rightarrow$  **büyüme**: giderek daha hızlı artar (örn.  $2^x$ ).
- $0 < b < 1 \rightarrow$  **çürüme**: sifıra doğru azalır (örn.  $(1/2)^x$ ).
- Her zaman **pozitif** (görüntü kümesi  $> 0$ ) ve  $b^0 = 1$  olduğu için daima  $(0, 1)$ 'den geçer.

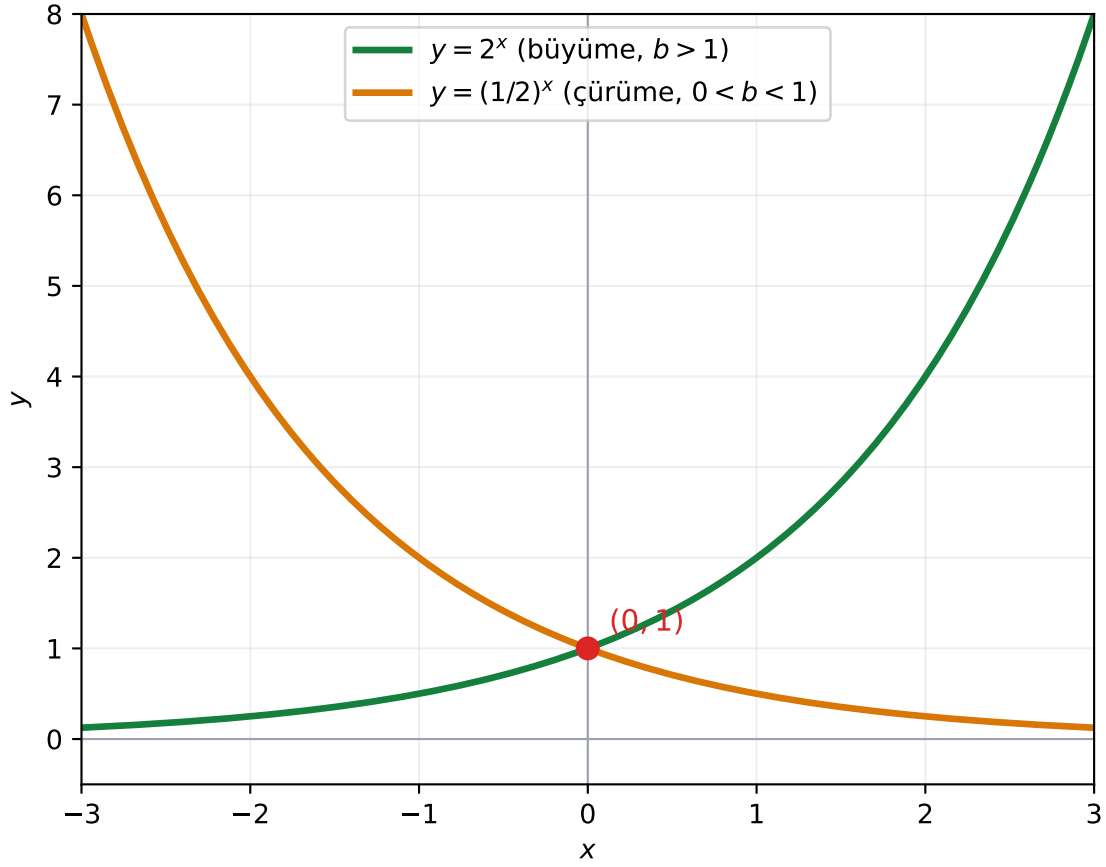
### 13.2. Doğrusal vs üstel: sabit fark mı, sabit oran mı?

Doğrusal fonksiyon her adımda sabit bir miktar **ekler**; üstel fonksiyon her adımda sabit bir oranla **çarpar**. Fark görünüşte küçük, sonuç dramatiktir:

$x$	Doğrusal $y = 2x (+2)$	Üstel $y = 2^x (\times 2)$
0	0	1
1	2	2
2	4	4
3	6	8
4	8	16
5	10	32

Başta yakınlar, ama üstel kısa sürede doğrusalı ezer. “Üstel büyüme” deyiminin gücü buradan gelir.

13. Üstel ve Logaritmik Fonksiyon



Şekil 13.1.: Üstel fonksiyon:  $b > 1$  büyüme ( $2^x$ ),  $0 < b < 1$  çürüme ( $(1/2)^x$ ). İkisi de  $(0, 1)$ 'den geçer ve daima pozitiftir.

### 13.3. Logaritmik fonksiyon: $f(x) = \log_b x$

Logaritma, üstelin **tersidir** (Logaritma dersindeki ayna ilişkisi). Bu yüzden üstelin zıttı gibi davranır:

- Artar ama giderek **yavaşlar** (üstelin tam tersi).
- Tanım kümesi  $x > 0$  (pozitif sayıların logaritması vardır).
- $\log_b 1 = 0$  olduğu için daima  $(1, 0)$ 'dan geçer.

### 13.4. Doğal taban $e$

ML ve kalkülüste en sık kullanılan taban  $e \approx 2,718$ 'dir.  $e^x$  üstel fonksiyonu,  $\ln x = \log_e x$  ise onun tersidir. Pratikte “üstel” denince çoğu zaman  $e^x$ , “log” denince  $\ln$  kastedilir.

#### ! İki uç

Üstel fonksiyon **çarpımsaldır** (sabit oran) ve yaygın fonksiyonlar içinde en hızlı büyüyendir. Logaritma onun tersidir: en yavaş yükselen fonksiyon. Bu ikisi, büyümenin ve onu geri çözmenin iki ucudur.

#### ! İki tuzak

- $b^x$  ile  $x^b$  karıştırılmaz: üstel fonksiyonda değişken **üstte** ( $2^x$ ), kuvvet fonksiyonunda değişken **tabanda** ( $x^2$ ) — büyüme hızları tamamen farklıdır.
- Üstel fonksiyon asla negatif ya da sıfır olmaz; çıktısı hep pozitiftir.

### 13.5. Örnek

**Örnek 1.**  $f(x) = 2^x$ :  $f(3) = 8$ ,  $f(0) = 1$ ,  $f(-1) = \frac{1}{2}$ .

**Örnek 2.**  $g(x) = \left(\frac{1}{3}\right)^x$ : taban  $0 < b < 1$  olduğu için çürüme;  $g(2) = \frac{1}{9}$ .

**Örnek 3.**  $\log_2 8 = 3$  (Logaritma dersinden): üssün tersi.

#### 💡 ML köprüsü

Üstel ve logaritma, sinir ağlarının her yerindedir:

- **Sigmoid:**  $\sigma(x) = \frac{1}{1 + e^{-x}}$  — herhangi bir reel sayıyı  $(0, 1)$  aralığına, yani bir olasılığa sıkıştırır. Kalbinde  $e^{-x}$  vardır.
- **Softmax:** sınıf skorlarını üstel alıp olasılığa çevirir (çok sınıflı sınıflandırmanın çıkışı).
- **Log-loss:** Logaritma dersinde gördüğün logaritma, kaybın temelidir.
- **Öğrenme oranı çürümesi:** eğitimde adım boyu çoğu zaman üstel çürümeyle küçültülür.

Bu dersteki büyüme/çürüme eğrileri, o fonksiyonların ham hâlidir.

### 13.6. Alıştırmalar

1.  $f(x) = 3^x$  için  $f(2) = ?$
2.  $f(x) = 2^x$  büyüme mi çürüme mi gösterir?
3.  $f(x) = \left(\frac{1}{2}\right)^x$  büyüme mi çürüme mi gösterir?
4. Her üstel fonksiyon  $b^x$  hangi noktadan geçer?
5.  $\log_3 9 = ?$
6.  $f(x) = 5^x$  için  $f(0) = ?$

**i** Çözümler (önce kendin dene, sonra aç)

1.  $3^2 = 9$
2. Büyüme (taban  $b = 2 > 1$ , artar)
3. Çürüme (taban  $0 < \frac{1}{2} < 1$ , azalır)
4.  $(0, 1)$ , çünkü  $b^0 = 1$
5. 2, çünkü  $3^2 = 9$
6. 1, çünkü  $5^0 = 1$

---

Sonraki ders: *Fonksiyon Bileşkesi.*

## 14. Fonksiyon Bileşkesi

$f(g(x))$  — sinir ağının iskeleti

Bir fonksiyonun çıktısını başka bir fonksiyona girdi yapmaya **bileşke** denir. Basit görünür ama derin öğrenmenin kavramsal anahtarlarıdır: bir sinir ağı, üst üste binmiş fonksiyonların bileşkesinden başka bir şey değildir.

### 14.1. Bileşke nedir?

$f(g(x))$  ifadesi şu sırayı söyler: **önce  $g$ 'yi uygula, sonra çıkan sonuca  $f$ 'yi uygula**. Gösterimi:

$$(f \circ g)(x) = f(g(x))$$

Bunu zincirlenmiş makineler gibi düşün — birinin çıktısı diğerinin girdisidir:



Şekil 14.1.: Fonksiyon bileşkesi  $f(g(x))$ : girdi önce  $g$  makinesinden geçer, onun çıktısı  $g(x)$  sonra  $f$  makinesine girer.

İçteki fonksiyon ( $g$ ) önce çalışır, dıştaki ( $f$ ) sonra. İfadeyi **içten dışa** doğru oku.

## 14. Fonksiyon Bileşkesi

### 14.2. Hesaplama

Bir sayıyı koymak için içten başla.  $f(x) = x + 1$  ve  $g(x) = x^2$  olsun:

$$f(g(2)) : g(2) = 4 \Rightarrow f(4) = 5$$

Formülün kendisini bulmak için  $g(x)$ 'i  $f$ 'nin içine yazarsın:

$$f(g(x)) = f(x^2) = x^2 + 1$$

### 14.3. Sıra önemlidir

Bileşkede sıra genellikle sonucu değiştirir:  $f(g(x)) \neq g(f(x))$ . Aynı  $f$  ve  $g$  ile ters sırayı dene:

$$g(f(x)) = g(x + 1) = (x + 1)^2 = x^2 + 2x + 1$$

$x^2 + 1$  ile aynı değil — hangi makinenin önce çalıştığı önemlidir.

#### ! İçten dışa

$f(g(x))$ 'te **içteki fonksiyon önce**, dıştaki sonra çalışır. Sırayı değiştirmek farklı bir fonksiyon verir;  $f(g(x))$  ile  $g(f(x))$  aynı şey değildir.

#### ⚠ İki tuzak

- $f(g(x))$  bir çarpım değildir:  $f(g(x)) \neq f(x) \cdot g(x)$ . Bileşke “uygulama”dır, çarpma değil.
- Parantezleri içten dışa çöz; en içteki girdiyi en önce hesapla.

### 14.4. Örnek

$f(x) = 2x$  ve  $g(x) = x^2$  olsun:

- $f(g(x)) = f(x^2) = 2x^2$
- $g(f(x)) = g(2x) = (2x)^2 = 4x^2$  — farklı.
- Sayıyla:  $f(g(3)) = f(9) = 18$ ;  $g(f(3)) = g(6) = 36$ .

#### 💡 ML köprüsü

Bir **sinir ağı**, **fonksiyon bileşkesinden ibarettir**. Her katman bir fonksiyondur; ağ bunları sırayla uygular:

$$\text{çıktı} = f_n(\dots f_2(f_1(x)))$$

Her katman genelde bir **doğrusal parça** ( $wx + b$ , Doğrusal Fonksiyon ve Eğim dersi) ve onun üstüne

bir **aktivasyon** ( $\sigma$ , ReLU — Üstel ve Logaritmik Fonksiyon dersi) bileşkesidir. Yani derin ağ = içten dışa zincirlenmiş yüzlerce makine.

Eğitimde kullanılan **geri yayılım (backprop)**, bu bileşkenin türevini **zincir kuralıyla** hesaplar — zincir kuralı bizzat  $f(g(x))$ 'in türev kuralıdır (kalkülüs sütununda göreceğiz). Bu dersteki bileşke, derin öğrenmenin iskeletidir.

## 14.5. Alıştırmalar

$f(x) = x + 2$  ve  $g(x) = 3x$  olsun (1–2):

1.  $f(g(x)) = ?$
2.  $g(f(x)) = ?$

$f(x) = x^2$  ve  $g(x) = x - 1$  olsun (3–4):

3.  $f(g(2)) = ?$
4.  $g(f(2)) = ?$
5.  $f(x) = 2x$ ,  $g(x) = x^2$  için  $f(g(3)) = ?$
6.  $f(x) = x + 5$ ,  $g(x) = x + 1$  için  $f(g(x)) = ?$

**i** Çözümler (önce kendin dene, sonra aç)

1.  $f(g(x)) = f(3x) = 3x + 2$
2.  $g(f(x)) = g(x + 2) = 3(x + 2) = 3x + 6$  (1'den farklı — sıra önemli)
3.  $g(2) = 2 - 1 = 1$ , sonra  $f(1) = 1^2 = 1$
4.  $f(2) = 4$ , sonra  $g(4) = 4 - 1 = 3$
5.  $g(3) = 9$ , sonra  $f(9) = 18$
6.  $g(x) = x + 1$ , sonra  $f(x + 1) = (x + 1) + 5 = x + 6$

Sonraki ders: Grafik Okuma.



## 15. Grafik Okuma

Eğriden bilgi çıkarmak — ML’de kayıp eğrilerini okumak

Bir grafik, bir fonksiyonun resmidir. Grafik okumak, o resimden gözle bilgi çıkarmaktır — yeni bir fonksiyon türü değil, Bölüm 2’de öğrendiklerimizi bir araya getiren bir beceri. Bu beceri ML’de günlük ekmektir: eğitim sırasında sürekli **kayıp eğrilerine** bakıp modelin ne yaptığını okursun.

### 15.1. Grafik bir fonksiyonun resmidir

Eğri üstündeki her  $(x, y)$  noktası “ $f(x) = y$ ” demektir. Bir değeri okumak için:  $x$  değerine git, dikey olarak eğriye çık, oradan yatay olarak  $y$  eksenine geç — bulduğun değer  $f(x)$ ’tir.

### 15.2. Neler okunur?

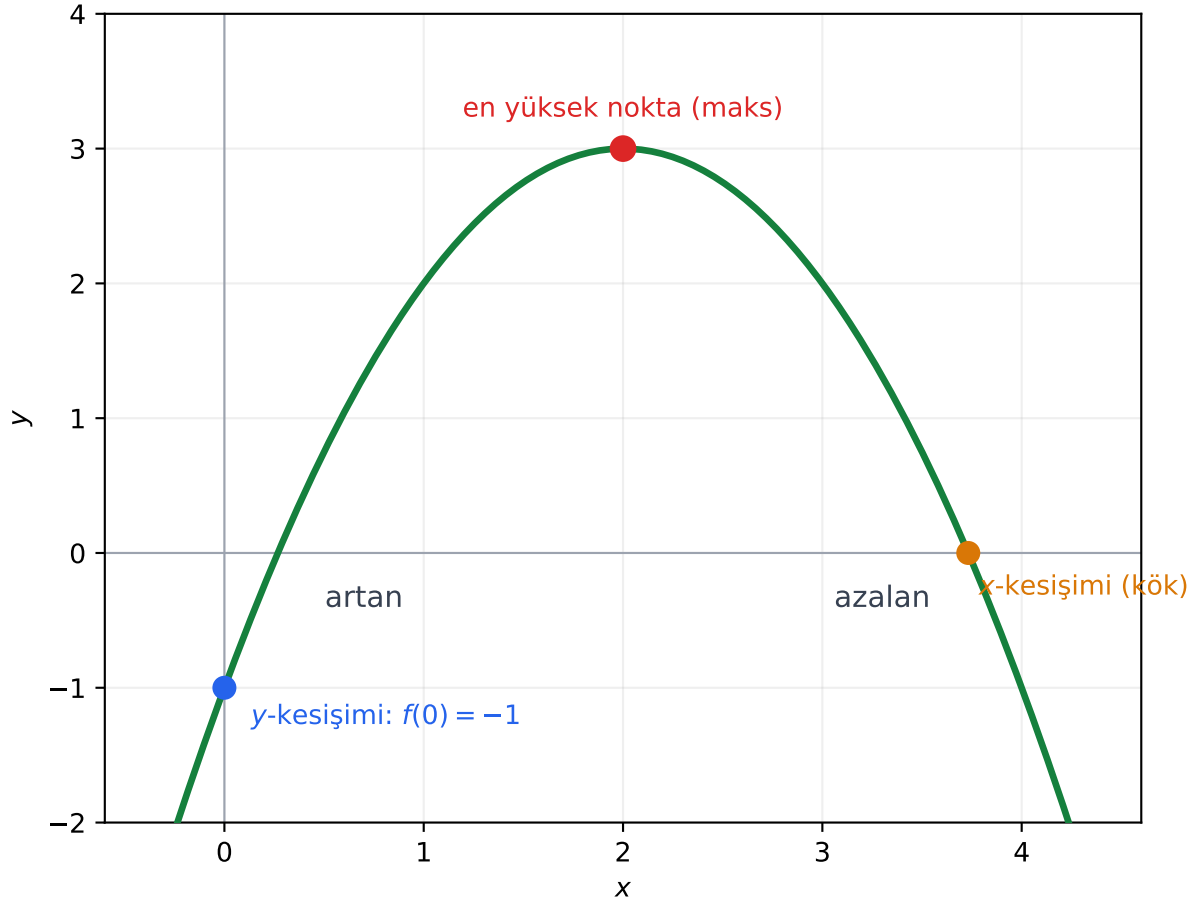
Bir grafiğe baktığında okuyacağın temel şeyler:

- **Eksenler ve ölçek:** her eksen neyi temsil ediyor, birim ne, ölçek doğrusal mı? (ML’de kayıp eğrileri çoğu zaman logaritmik eksenle çizilir.)
- **Kesişimler:**  $y$ -kesişi  $f(0)$ ’dır (eğri  $y$  eksenini nerede keser);  $x$ -kesişimleri köklerdir (eğri  $x$  eksenini nerede keser,  $y = 0$ ).
- **Artan / azalan bölgeler:**  $x$  büyürken  $y$  de büyüyorsa artan, küçülüyorsa azalan.
- **Maksimum / minimum:** tepe ve dip noktaları (parabolün tepe noktası gibi).
- **Diklik:** eğri ne kadar hızlı değişiyor — dik bölge hızlı değişim, yatay bölge yavaş değişim (türevin habercisi).
- **Genel davranış:** doğrusal mı, eğri mi, büyüyen mi, bir yerde düzleşiyor mu (plato)?

### 15.3. Bir değeri okumak

Yukarıdaki grafikte  $f(0)$ ’ı okumak için  $x = 0$ ’a git, eğriye bak: değer  $-1$ . En yüksek noktayı bulmak için tepeye bak:  $(2, 3)$ .

15. Grafik Okuma



Şekil 15.1.: Bir grafikten okunanlar:  $y$ -kesiřimi,  $x$ -kesiřimi (kök), artan ve azalan bölgeler, en yüksek nokta (maksimum).

## ! Önce eksenini oku

Bir grafięe bakarken ilk iř **eksenleri ve ölçeęi** anlamaktır: hangi eksen neyi gösteriyor, ölçek doęrusal mı logaritmik mi? Ölçeęi yanlış okursan grafięin tamamını yanlış okursun.

## ! İki tuzak

- Ölçeęin doęrusal olduęunu varsayma; ML grafiklerinde logaritmik eksen sık kullanılır (eřit aralıklar 1, 10, 100, 1000 olabilir).
- $x$ -kesiřimi (kök,  $y = 0$ ) ile  $y$ -kesiřimini ( $f(0)$ ,  $x = 0$ ) karıřtırma — ikisi farklı eksenlerde, farklı şeylerdir.

## 💡 ML köprüsü

Bir modeli eęitirken en çok baktıęın grafik **kayıp eęrisidir** (kayıp  $\leftrightarrow$  epoch/iterasyon). Onu okumak, bu derste öğrendięin her şeyi kullanır:

- **Azalıyor mu?** Kayıp düşüyorsa model öğreniyor.
- **Düzleřti mi (plato)?** Eęri yataylařtıysa öğrenme durmuř; öğrenme oranını deęiřtirme zamanı.
- **Yükseliyor / zıplıyor mu?** Eęitim iraksıyor (öğrenme oranı çok yüksek olabilir).
- **Ařırı öğrenme (overfitting):** eęitim kaybı düşerken doęrulama kaybı yükseliyorsa, iki eęrinin ayrıřması bunun imzasıdır.

Bu eęriler genelde **logaritmik ekseninde** (Üstel ve Logaritmik Fonksiyon dersi) çizilir; ölçeęi doęru okumak řart. Grafik okuma, modelin durumunu teřhis etme becerisidir.

## 15.4. Alıřtırmalar

Yukarıdaki fig-grafik-okuma grafięine bakarak yanıtla:

1. Grafikte  $y$ -kesiřimi kaçtır?
2. En yüksek nokta (maksimum) hangi koordinattadır?
3.  $x = 1$  civarında fonksiyon artan mı azalan mı?
4.  $x = 3$  civarında fonksiyon artan mı azalan mı?
5. Fonksiyonun bir maksimumu mu yoksa minimumu mu vardır?
6. Eęri  $x$  eksenini kaç noktada keser (kaç kök)?

## i Çözümler (önce kendin dene, sonra aç)

1.  $-1$  (çünkü  $f(0) = -1$ , mavi nokta).
2.  $(2, 3)$  (kırmızı nokta, tepe).
3. Artan ( $x < 2$  bölgesinde eęri yükseliyor).
4. Azalan ( $x > 2$  bölgesinde eęri iniyor).
5. Maksimum (kollar ařaęı; en yüksek noktası var, en alçak deęil).
6. İki noktada (eęri  $x$  eksenini iki kez keser  $\rightarrow$  iki kök).

15. *Grafik Okuma*

---

*Bölüm 2 — Fonksiyonlar tamamlandı. Sonraki bölüm: Toplam, Diziler ve Sayma.*

## **Kısım III.**

### **Bölüm 3 — Toplam, Diziler ve Sayma**



## 16. Diziler

Sıralı sayı listeleri — verinin indekslenmesi

Bir **dizi**, belirli bir sırada yazılmış sayı listesidir. Basit görünür ama notasyonu ML'in dilidir: veri noktalarını, ağırlıkları, iterasyonları hep alt indisli bir diziyle yazarız. Bu ders aynı zamanda bir sonraki dersin — toplam sembolünün — zeminidir.

### 16.1. Dizi nedir?

Bir dizi, terimleri sırayla dizilmiş sayılardır:

$$a_1, a_2, a_3, a_4, \dots$$

Her sayı bir **terim**, sırasını gösteren sayı ise **indistir**.  $a_n$ , dizinin  $n$ . **terimi** demektir. Çoğu dizi bir **genel terim** formülüyle tanımlanır; örneğin  $a_n = 2n$  dizisi 2, 4, 6, 8, ... verir (her  $n$ 'i koyarak terimi bulursun).

### 16.2. Aritmetik dizi: sabit fark

Ardışık terimler arasındaki **fark sabitse** dizi aritmetiktir. Bu sabite **ortak fark** ( $d$ ) denir:

$$a_n = a_1 + (n - 1) d$$

Örnek: 3, 7, 11, 15, ... — her adımda +4, yani  $d = 4$ . Bu, Doğrusal Fonksiyon ve Eğitim dersindeki doğrusal fonksiyonun ayrık hâlidir (sabit fark = sabit eğim).

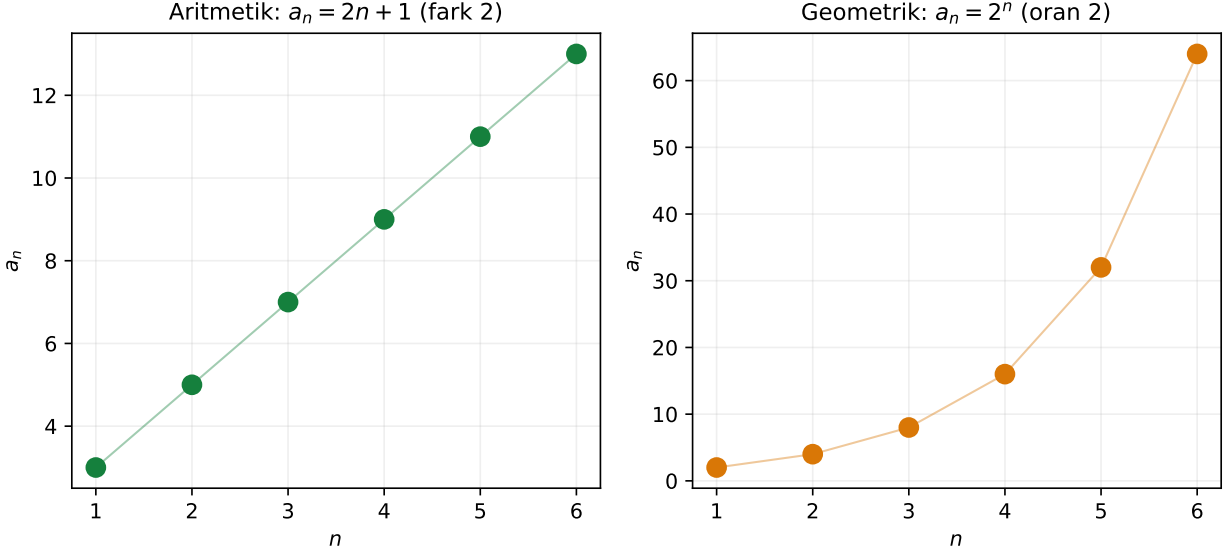
### 16.3. Geometrik dizi: sabit oran

Ardışık terimler arasındaki **oran sabitse** dizi geometriktir. Bu sabite **ortak oran** ( $r$ ) denir:

$$a_n = a_1 \cdot r^{n-1}$$

Örnek: 2, 6, 18, 54, ... — her adımda  $\times 3$ , yani  $r = 3$ . Bu da Üstel ve Logaritmik Fonksiyon dersindeki üstel fonksiyonun ayrık hâlidir (sabit oran = üstel büyüme).

## 16. Diziler



Şekil 16.1.: Aritmetik dizi (sabit fark) bir doğru üstünde, geometrik dizi (sabit oran) bir üstel eğri üstünde dizilen ayrı noktalar.

### 16.4. Bir terimi bulmak

Formüle indisi koyarsın.  $a_1 = 5$ ,  $d = 3$  olan aritmetik dizide:

$$a_4 = 5 + (4 - 1) \cdot 3 = 5 + 9 = 14$$

! İki rejim, tanıdık

**Aritmetik dizi = sabit fark** (ayrık doğru), **geometrik dizi = sabit oran** (ayrık üstel). Bunlar doğrusal (Doğrusal Fonksiyon ve Eğim dersi) ve üstel (Üstel ve Logaritmik Fonksiyon dersi) büyümenin tam olarak aynı iki rejimi — sadece sürekli eğri yerine ayrı noktalar.

! İki tuzak

- Formülde  $(n - 1)$  var,  $n$  değil: ilk terim  $a_1$ 'in çarpanı/üssü sıfırdır. En sık yapılan hata buradaki bir-kayma (off-by-one).
- Aritmetik mi geometrik mi karıştırma: ardışık terimleri **çıkarak** (fark sabit mi?) ve **bölerek** (oran sabit mi?) kontrol et.

### 16.5. Örnek

**Örnek 1.**  $a_n = 3n - 1$ : ilk terimler 2, 5, 8, 11, ... — aritmetik,  $d = 3$ .

**Örnek 2.** 5, 10, 20, 40, ...: her adımda  $\times 2 \rightarrow$  geometrik,  $r = 2$ , genel terim  $a_n = 5 \cdot 2^{n-1}$ .

 ML köprüsü

Alt indis notasyonu ML'in temel yazımıdır:

- Eğitim verisi sıralı bir dizidir:  $x_1, x_2, \dots, x_n$  (her  $x_i$  bir örnek). Ağırlıklar da öyle:  $w_1, w_2, \dots$
- Eğitim adımları (iterasyon/epoch) bir dizi oluşturur; öğrenme oranı çoğu zaman **geometrik bir diziyle** azaltılır (üstel çürüme — Üstel ve Logaritmik Fonksiyon dersi).
- Bir sonraki ders olan toplam sembolü, bir dizinin terimlerini toplamaktır — ki ML'de ortalama, kayıp ve iç çarpım hep böyle bir toplamdır.

## 16.6. Alıřtırmalar

1.  $a_n = 2n + 3$  dizisinin ilk üç terimi nedir?
2. 3, 7, 11, 15, ... dizisi aritmetik mi geometrik mi? Ortak fark/oran kaç?
3. 2, 6, 18, 54, ... dizisi aritmetik mi geometrik mi? Ortak fark/oran kaç?
4. Aritmetik dizide  $a_1 = 5, d = 3$  ise  $a_4 = ?$
5. Geometrik dizide  $a_1 = 2, r = 2$  ise  $a_5 = ?$
6.  $a_n = n^2$  dizisinin 4. terimi nedir?

 Çözümler (önce kendin dene, sonra aç)

1.  $2(1) + 3 = 5, 2(2) + 3 = 7, 2(3) + 3 = 9 \rightarrow 5, 7, 9$
2. Aritmetik; ardışık fark sabit 4  $\rightarrow d = 4$
3. Geometrik; ardışık oran sabit 3  $\rightarrow r = 3$
4.  $a_4 = 5 + (4 - 1) \cdot 3 = 5 + 9 = 14$
5.  $a_5 = 2 \cdot 2^{5-1} = 2 \cdot 16 = 32$
6.  $a_4 = 4^2 = 16$

---

Sonraki ders: Toplam Sembolü ( $\Sigma$ ).



## 17. Toplam Sembolü ( $\Sigma$ )

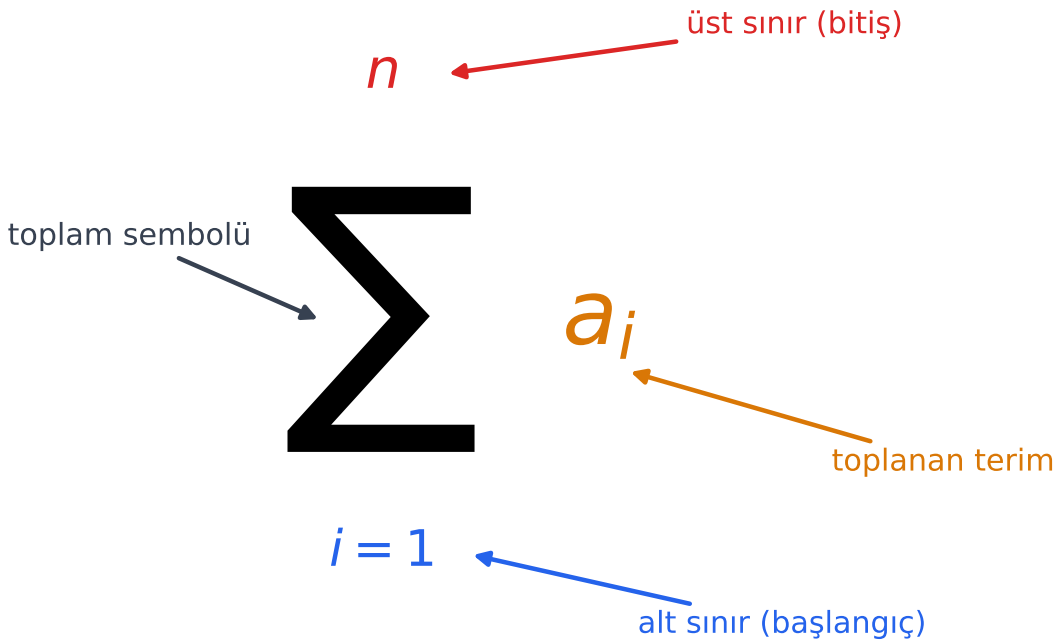
Çok terimi tek sembolde toplamak — ML'in en sık işlemi

Toplam sembolü  $\Sigma$  (büyük sigma), “şu terimleri topla” demenin kısa yoludur. ML'de en çok karşına çıkan semboldür: ortalama, kayıp fonksiyonu ve iç çarpım — hepsi birer toplamdır. Sembol ilk bakışta korkutucu görünür; aslında bir döngünün matematik yazımıdır.

### 17.1. $\Sigma$ ne demek?

$$\sum_{i=1}^n a_i = a_1 + a_2 + a_3 + \dots + a_n$$

Sembolün dört parçası vardır:



Şekil 17.1.: Toplam sembolünün anatomisi: alt sınır (başlangıç), üst sınır (bitiş), toplanan terim ve toplam sembolünün kendisi. İndis  $i$  alttan üste sırayla değer alır.

## 17. Toplam Sembolü ( $\Sigma$ )

- **Toplam değişkeni (indis):**  $i$  — alt sınırdan üst sınıra doğru tek tek değer alır.
- **Alt sınır:**  $i = 1$  — saymanın başladığı yer.
- **Üst sınır:**  $n$  — saymanın bittiği yer.
- **Toplanan terim:**  $a_i$  — her adımda hesaplanıp toplama eklenen ifade.

### 17.2. Nasıl açılır?

İndise sırayla değer ver, sonuçları topla:

$$\sum_{i=1}^4 i = 1 + 2 + 3 + 4 = 10$$

$$\sum_{i=1}^3 i^2 = 1^2 + 2^2 + 3^2 = 1 + 4 + 9 = 14$$

$$\sum_{i=1}^3 5 = 5 + 5 + 5 = 15 \quad (\text{sabit terim: } n \text{ kopya})$$

### 17.3. $\Sigma$ kuralları

Bu üç kural ML'de toplamları sadeleştirmenin temelidir:

$$\sum_{i=1}^n c \cdot a_i = c \sum_{i=1}^n a_i \quad \sum_{i=1}^n (a_i + b_i) = \sum_{i=1}^n a_i + \sum_{i=1}^n b_i \quad \sum_{i=1}^n c = nc$$

Yani sabit çarpan dışarı alınır, toplam üzerinden açılır, sabitin toplamı  $n$  kopyadır.

### 17.4. Ortalama bir $\Sigma$ 'dir

Bir veri kümesinin ortalaması, aslında ölçeklenmiş bir toplamdır:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

! Korkutucu değil

$\sum_{i=1}^n a_i$  sadece şu demektir: “ $i$ 'yi 1'den  $n$ 'e kadar yürüt, her adımda  $a_i$ 'yi hesapla, hepsini topla.” Bu, kodda bir for döngüsünün matematikteki yazımıdır — başka bir şey değil.

**⚠ İki tuzak**

- Sınırlar dahildir:  $i = 1$ 'den  $n$ 'e demek, iki uç da dahil  $n$  **terim** demektir.
- Sabit terimin toplamı  $n$  kopyadır:  $\sum_{i=1}^n c = nc$ ,  $c$  değil. İndis adı önemsizdir;  $i, j, k$  aynı kopyaya çıkar.

## 17.5. Örnek

**Örnek 1.**  $\sum_{i=1}^3 2i$  — kuralla:  $2 \sum_{i=1}^3 i = 2(1 + 2 + 3) = 2 \cdot 6 = 12$ .

**Örnek 2.**  $\sum_{k=1}^4 (k + 1) = (1+1) + (2+1) + (3+1) + (4+1) = 2 + 3 + 4 + 5 = 14$ .

**💡 ML köprüsü**

$\Sigma$ , makine öğrenmesinin en sık işlemidir:

- **Ortalama:**  $\bar{x} = \frac{1}{n} \sum x_i$ .
- **MSE kaybı:**  $MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$  — hataların karelerinin ortalaması (Parabol dersinden tanıdık parabol burada toplanıyor).
- **İç çarpım / ağırlıklı toplam:**  $y = \sum_{i=1}^n w_i x_i + b$  — tek bir nöronun hesabı. Bu, Doğrusal Fonksiyon ve Eğitim dersindeki  $wx + b$ 'nin çok özelliğe genelleşmiş hâlidir: her özellik  $x_i$  kendi ağırlığı  $w_i$  ile çarpılıp toplanır.

Kodda bu satırların hepsi bir for döngüsü veya bir np.sum çağrısıdır.

## 17.6. Alıştırmalar

1.  $\sum_{i=1}^5 i = ?$

2.  $\sum_{i=1}^3 i^2 = ?$

3.  $\sum_{i=1}^4 3 = ?$

4.  $\sum_{i=1}^3 2i = ?$

5.  $\sum_{k=1}^4 (k + 1) = ?$

6.  $x = (2, 4, 6)$  için ortalama  $\bar{x} = \frac{1}{3} \sum_{i=1}^3 x_i = ?$

17. Toplam Sembolü ( $\Sigma$ )

**i** Çözümler (önce kendin dene, sonra aç)

1.  $1 + 2 + 3 + 4 + 5 = 15$

2.  $1 + 4 + 9 = 14$

3.  $3 \cdot 4 = 12$  (dört kopya)

4.  $2(1) + 2(2) + 2(3) = 2 + 4 + 6 = 12$

5.  $2 + 3 + 4 + 5 = 14$

6.  $\frac{2 + 4 + 6}{3} = \frac{12}{3} = 4$

---

Sonraki ders: *Pi Çarpım Notasyonu.*

## 18. Pi Çarpım Notasyonu

$\Sigma$ 'nın çarpım kardeşi — olabirliğin dili

Toplam sembolü  $\Sigma$  terimleri **topluyordu**. Büyük pi ( $\Pi$ ) ise aynı yapıyla terimleri **çarpır**.  $\Sigma$ 'nın çarpım kardeşidir — ve makine öğrenmesinde olabirliğin (likelihood) dilidir; log-loss'un neden var olduğunu da bu sembol açıklar.

### 18.1. $\Pi$ ne demek?

$$\prod_{i=1}^n a_i = a_1 \cdot a_2 \cdot a_3 \cdots a_n$$

Anatomisi  $\Sigma$  ile birebir aynıdır (indis, alt sınır, üst sınır, terim) — tek fark, işlemin **çarpma** olmasıdır.

### 18.2. Nasıl hesaplanır?

İndise sırayla değer ver, sonuçları çarp:

$$\prod_{i=1}^4 i = 1 \cdot 2 \cdot 3 \cdot 4 = 24$$
$$\prod_{i=1}^3 2 = 2 \cdot 2 \cdot 2 = 8 \quad (\text{sabit terim: } 2^n)$$

### 18.3. Faktöriyel bir çarpımdır

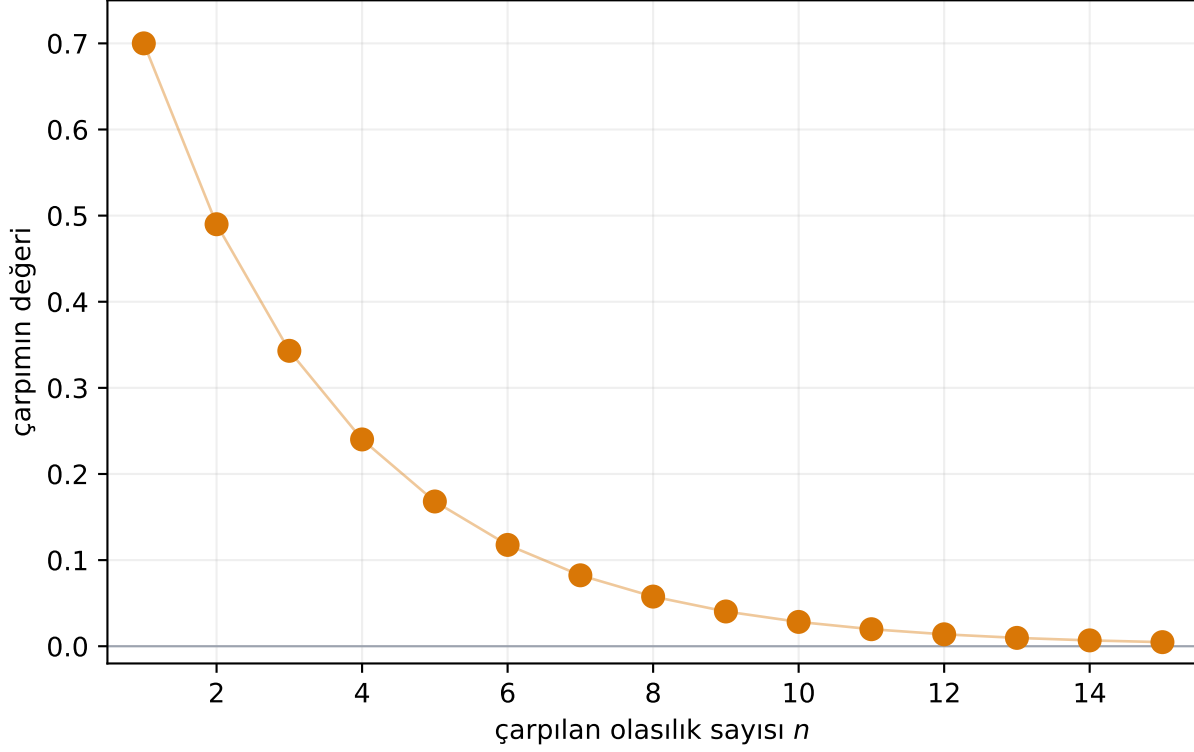
İleride göreceğin Faktöriyel ve Sayma dersindeki faktöriyel, aslında tam olarak bir  $\Pi$ 'dir:

$$n! = \prod_{i=1}^n i$$

Yani  $4! = \prod_{i=1}^4 i = 24$ . Çarpım notasyonu, faktöriyeli kısa ve kesin yazmanın yoludur.

18.4.  $\Pi \rightarrow \log \rightarrow \Sigma$ 

$\Pi$ 'nin ML'deki asıl önemi şu zincirde: olasılıkların çarpımı çok hızlı küçülür.



Şekil 18.1.: Olasılıkların çarpımı hızla küçülür: her biri 0.7 olan 15 olasılık çarpılınca sonuç  $\sim 0.005$ 'e iner; çok terimde sayısal olarak sifıra çöker. Bu yüzden logaritma alıp toplamaya döneriz.

Çözüm logaritmadır (Logaritma dersi): logaritma çarpımı toplamaya çevirdiği için, çarpım yerine logaritmaların toplamını kullanırız:

$$\log \left( \prod_{i=1}^n p_i \right) = \sum_{i=1}^n \log p_i$$

Sağ taraf sayısal olarak kararlıdır — küçük sayıların çarpımı sifıra çökerken, logaritmalarının toplamı çökmez.

!  $\Pi$  çarpar,  $\Sigma$  toplar

$\Pi$  terimleri çarpar,  $\Sigma$  toplar — anatomileri aynı, işlemleri farklı. Faktöriyel bir çarpımdır:  $n! = \prod_{i=1}^n i$ .  
Ve olasılık çarpımları, logaritma alınca toplama dönüşür.

## ⚠ İki tuzak

- $\Pi$ 'de sabit terim  $c^n$ 'dir ( $n$  kez çarpılır),  $nc$  değil —  $\Sigma$ 'daki  $nc$  ile karıştırma.
- Çok sayıda küçük sayının çarpımı sayısal olarak sıfıra çöker (underflow); pratikte bu yüzden çarpım yerine log-toplam kullanılır.

## 18.5. Örnek

Örnek 1.  $\prod_{i=1}^3 i = 1 \cdot 2 \cdot 3 = 6$  (yani  $3!$ ).

Örnek 2.  $\prod_{i=1}^3 (i + 1) = 2 \cdot 3 \cdot 4 = 24$ .

## 💡 ML köprüsü

Bir modelin **olabilirliği** (likelihood), tüm veri noktalarının olasılıklarının çarpımıdır (bağımsızlık varsayımıyla):

$$L = \prod_{i=1}^n p_i$$

Bu çarpım çok terimde sıfıra çöktüğü için, ML modelleri bunun yerine **log-olabilirliği** kullanır:  $\log L = \sum_{i=1}^n \log p_i$ . Maksimize edilen şey log-olabilirlik, minimize edilen kayıp ise log-loss'tur. Yani  $\Pi$  (bu ders) + logaritma (Logaritma dersi) +  $\Sigma$  (Toplam Sembölü dersi), neden log aldığımızın tam açıklamasıdır.

## 18.6. Alıştırmalar

1.  $\prod_{i=1}^3 i = ?$
2.  $\prod_{i=1}^4 i = ?$  (hangi tanıdık sayı?)
3.  $\prod_{i=1}^3 2 = ?$
4.  $n!$ 'i çarpım notasyonu ile nasıl yazarsın?
5.  $\prod_{i=1}^3 (i + 1) = ?$
6. Olabilirlik  $L = p_1 \cdot p_2 \cdot p_3$ 'ü  $\Pi$  ile yaz.

## 18. Pi Çarpım Notasyonu

**i** Çözümler (önce kendin dene, sonra aç)

1.  $1 \cdot 2 \cdot 3 = 6$

2.  $1 \cdot 2 \cdot 3 \cdot 4 = 24$  (yani  $4!$ )

3.  $2 \cdot 2 \cdot \frac{2}{n} = 8$

4.  $n! = \prod_{i=1}^n i$

5.  $2 \cdot 3 \cdot \frac{4}{3} = 24$

6.  $L = \prod_{i=1}^3 p_i$

---

Sonraki ders: Seriler.

## 19. Seriler

Bir dizinin toplamı — kapalı formüller ve yakınsama

**Dizi** sıralı bir sayı listesiydi; **seri** ise o listenin **toplamıdır**. Bir seriyi terim terim toplamak yerine kapalı bir formülle anında hesaplayabiliriz — ve bazı sonsuz toplamlar şaşırtıcı biçimde sonlu bir değere yakınsar. Bu yakınsama fikri, ileride kalkülüsteki limitin tohumudur.

### 19.1. Seri nedir?

Bir dizinin ilk  $n$  teriminin toplamına **kısmi toplam** denir:

$$S_n = a_1 + a_2 + \cdots + a_n = \sum_{i=1}^n a_i$$

Geçen dersteki  $\Sigma$ 'nın somut hâli — ama şimdi onu hızlıca hesaplayan formülleri arıyoruz.

### 19.2. Aritmetik seri

Bir aritmetik dizinin toplamı, “terim sayısı çarpı ilk ve son terimin ortalaması”dır:

$$S_n = \frac{n}{2} (a_1 + a_n)$$

En tanıdık özel hâli 1’den  $n$ ’e kadar sayıların toplamıdır:

$$1 + 2 + 3 + \cdots + n = \frac{n(n+1)}{2}$$

Bunun sezgisi şu eşleştirmedir: ilk ile son, ikinci ile sondan bir önceki... her çift aynı toplamı verir ( $a_1 + a_n$ ), ve  $\frac{n}{2}$  tane çift vardır.

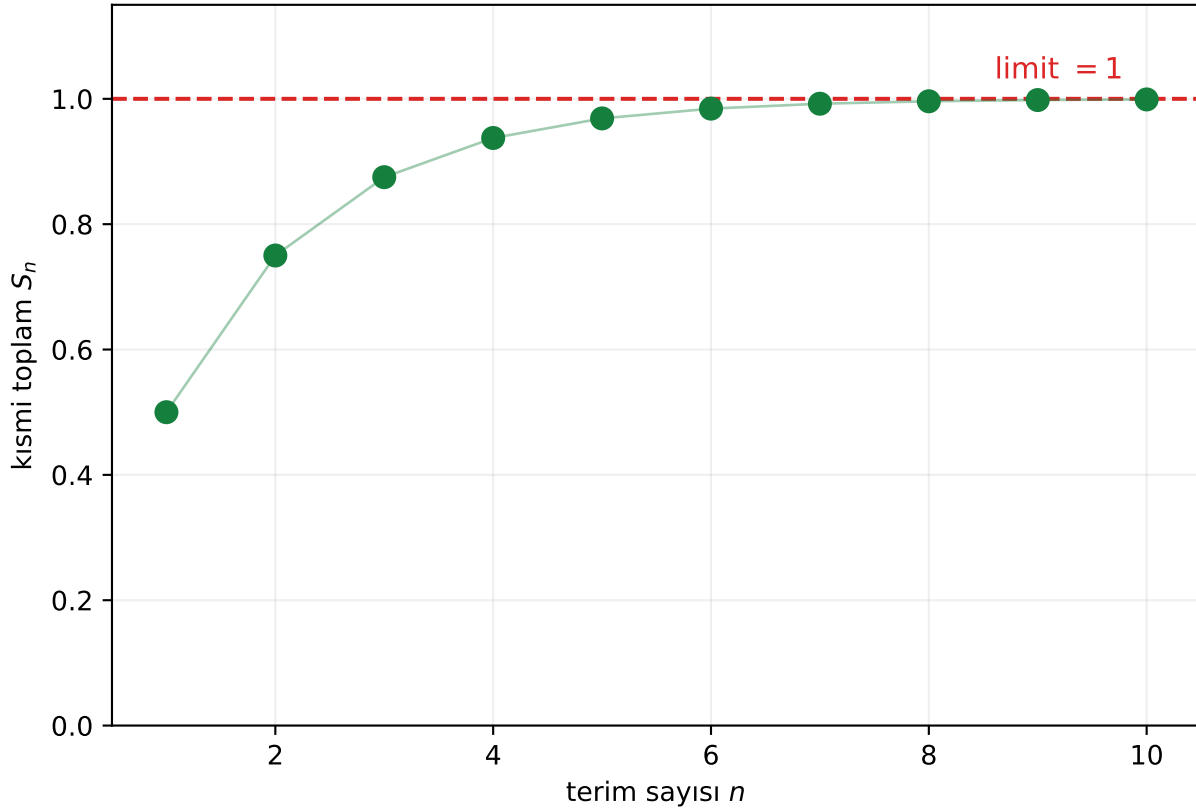
## 19.3. Geometrik seri

Bir geometrik dizinin toplamı ( $r \neq 1$  için):

$$S_n = a_1 \cdot \frac{r^n - 1}{r - 1}$$

Eğer  $|r| < 1$  ise, sonsuza kadar toplansan bile sonuç **sonludur** — seri yakınsar:

$$S_\infty = \frac{a_1}{1 - r} \quad (|r| < 1)$$



Şekil 19.1.: Sonsuz geometrik seri  $1/2 + 1/4 + 1/8 + \dots$  : kısmi toplamlar 1'e yakınsar ( $|r| < 1$ ).

Her terim bir öncekinin yarısı kadar eklediği için toplam 1'i asla geçmez; sonsuzda tam 1'e oturur.

! Tek hesapta toplam

Bir seri formülü, “ $n$  terimi tek tek topla” işini **tek bir hesaba** indirger. Dahası,  $|r| < 1$  olan geometrik seride sonsuz sayıda terimin toplamı bile sonlu çıkar — sonsuzluk her zaman sonsuz sonuç vermez.

### ⚠ Üç tuzak

- Geometrik sonlu formül  $r \neq 1$  ister (payda  $r - 1$  sıfır olamaz).
- Sonsuz toplam **yalnızca**  $|r| < 1$  ise yakınsar;  $|r| \geq 1$  ise terimler küçülmez ve toplam ıraksar (sonsuz gider).
- Diziyi (terimler) seriyle (terimlerin toplamı) karıştırma.

## 19.4. Örnek

**Örnek 1 (aritmetik).**  $1 + 2 + \dots + 10 = \frac{10 \cdot 11}{2} = 55$ .

**Örnek 2 (geometrik, sonlu).**  $2 + 6 + 18 + 54$  ( $a_1 = 2, r = 3, n = 4$ ):  $2 \cdot \frac{3^4 - 1}{3 - 1} = 2 \cdot \frac{80}{2} = 80$ .

**Örnek 3 (geometrik, sonsuz).**  $\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \dots = \frac{1/2}{1 - 1/2} = 1$ .

### 💡 ML köprüsü

- **Karmaşıklık analizi:** iç içe döngülerin adım sayısı çoğu zaman  $1 + 2 + \dots + n = \frac{n(n+1)}{2}$  çıkar — bu da  $O(n^2)$  demektir.
- **Pekiştirmeli öğrenme (RL):** bir ajanın iskontolu getirisi  $\sum_t \gamma^t r_t$  bir **geometrik ağırlıklı seridir**; iskonto çarpanı  $\gamma < 1$  olduğu için ( $|r| < 1$  koşulu) sonsuz ufuktaki toplam bile sonlu kalır.
- **Yakınsama:** kısmi toplamların bir limite oturması, kalkülüsteki **limit** kavramının ta kendisidir — ve eğitimde kayıp/tahminlerin bir değere oturmasının da resmidir.

## 19.5. Alıştırmalar

1.  $1 + 2 + 3 + \dots + 20 = ?$
2. Aritmetik seri:  $a_1 = 2, a_n = 20, n = 10$  için  $S_{10} = ?$
3.  $5 + 10 + 15 + 20 + 25 = ?$  (formülle)
4. Geometrik seri:  $2 + 6 + 18 + 54$  ( $a_1 = 2, r = 3, n = 4$ ) toplamı?
5. Sonsuz geometrik:  $\frac{1}{3} + \frac{1}{9} + \frac{1}{27} + \dots = ?$
6. Sonsuz geometrik:  $1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \dots = ?$

### i Çözümler (önce kendin dene, sonra aç)

1.  $\frac{20 \cdot 21}{2} = 210$
2.  $\frac{10}{2}(2 + 20) = 5 \cdot 22 = 110$

19. Seriler

$$3. \frac{5}{2}(5 + 25) = 2,5 \cdot 30 = 75$$

$$4. 2 \cdot \frac{3^4 - 1}{3 - 1} = 2 \cdot \frac{80}{2} = 80$$

$$5. a_1 = \frac{1}{3}, r = \frac{1}{3}: \frac{1/3}{1 - 1/3} = \frac{1/3}{2/3} = \frac{1}{2}$$

$$6. a_1 = 1, r = \frac{1}{2}: \frac{1}{1 - 1/2} = 2$$

---

Sonraki ders: Faktöriyel ve Sayma.

## 20. Faktöriyel ve Sayma

Kaç farklı yol? — olasılığa açılan kapı

Sayma, “kaç farklı yol var?” sorusunun matematiğidir. Faktöriyel, permütasyon ve kombinasyon, bu sorunun üç temel aracıdır. Aynı zamanda olasılık sütununun kapısıdır: bir olayın olasılığını hesaplamak çoğu zaman “kaç farklı sonuç var?” diye saymakla başlar.

### 20.1. Faktöriyel: $n!$

Faktöriyel,  $n$ 'den 1'e kadar olan sayıların çarpımıdır:

$$n! = n \cdot (n - 1) \cdot (n - 2) \cdots 2 \cdot 1$$

Örneğin  $4! = 4 \cdot 3 \cdot 2 \cdot 1 = 24$ . Tanım gereği  $0! = 1$ 'dir. Faktöriyel,  $n$  farklı nesnenin kaç farklı **sıralanışı** olduğunu sayar — örneğin 4 kişi bir sıraya  $4! = 24$  farklı şekilde dizilebilir.

### 20.2. Sayma temel ilkesi

Bir seçimin  $m$ , bağımsız bir başka seçimin  $n$  seçeneği varsa, ikisi birlikte  $m \times n$  farklı yol verir. Örneğin 3 farklı gömlek ve 2 farklı pantolonla  $3 \times 2 = 6$  farklı kombinasyon kurulur. Permütasyon ve kombinasyon bu ilkenin üstüne kurulur.

### 20.3. Permütasyon: sıra önemli

$n$  nesneden  $r$  tanesini **sıralı** seçmenin (dizmenin) yol sayısı:

$$P(n, r) = \frac{n!}{(n - r)!}$$

Örnek: 5 kişiden 3'ünü sıraya dizmek  $\rightarrow P(5, 3) = \frac{5!}{2!} = \frac{120}{2} = 60$ .

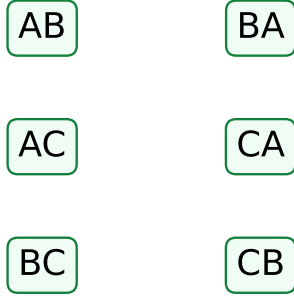
## 20.4. Kombinasyon: sıra önemsiz

$n$  nesneden  $r$  tanesini, **sırası önemsiz** biçimde seçmenin yol sayısı:

$$C(n, r) = \binom{n}{r} = \frac{n!}{r!(n-r)!}$$

Örnek: 5 kişiden 3'ünü (sıra fark etmeksizin) seçmek  $\rightarrow C(5, 3) = \frac{5!}{3!2!} = \frac{120}{6 \cdot 2} = 10$ .

### Permütasyon (sıra önemli)



6 sonuç =  $P(3, 2)$

### Kombinasyon (sıra önemsiz)



3 sonuç =  $C(3, 2)$

Şekil 20.1.: {A, B, C} kümesinden 2 seçmek: permütasyonda sıra önemli (6 sonuç), kombinasyonda sıra önemsiz (3 sonuç). AB ile BA permütasyonda ayrı, kombinasyonda aynıdır.

Kombinasyon, permütasyonun sıra-varyantlarını tek bir seçim olarak sayar; bu yüzden  $C(n, r) = \frac{P(n, r)}{r!}$   
— sıralanışları  $r!$ 'e bölerek elersin.

! Tek soru: sıra önemli mi?

Saymada sorulacak tek soru şudur: **sıra önemli mi?**

- Önemli (dizmek, sıralamak)  $\rightarrow$  **permütasyon.**
- Önemsiz (seçmek, grup oluşturmak)  $\rightarrow$  **kombinasyon.**

Kombinasyon = permütasyon  $\div r!$  (fazladan sayılan sıralanışları böl).

### ⚠ İki tuzak

- $0! = 1$ 'dir, 0 değil — bu, formüllerin kenar durumlarında kritik.
- Faktöriyel çok hızlı büyür ( $10! = 3.628.800$ ); büyük permütasyonları kaba kuvvetle tek tek üretmek pratikte imkânsızdır (kombinatoryal patlama). Bu, neden akıllı algoritmalara ihtiyaç duyduğumuzun bir sebebidir.

## 20.5. Örnek

**Örnek 1.**  $5! = 120$ .

**Örnek 2 (permütasyon).** 6 kitaptan 2'sini rafa sıralı dizmek:  $P(6, 2) = \frac{6!}{4!} = 6 \cdot 5 = 30$ .

**Örnek 3 (kombinasyon).** 6 kişiden 2 kişilik bir komite seçmek:  $C(6, 2) = \frac{6!}{2!4!} = \frac{6 \cdot 5}{2} = 15$ .

### 💡 ML köprüsü

Sayma, **olasılığın temelidir** — bir sonraki büyük sütun:

- Kombinasyon  $\binom{n}{r}$ , **binom katsayısıdır** ve doğrudan **binom dağılımına** girer ( $n$  denemede  $r$  başarı olasılığı). İstatistiğin temel taşlarından biri.
- Ayrık olasılıkta bir olayın olasılığı çoğu zaman  $\frac{\text{uygun sonuç sayısı}}{\text{toplam sonuç sayısı}}$ 'dır — pay ve payda birer sayma problemidir.

Bu yüzden saymayı sağlam oturtmak, olasılık ve istatistiğe geçişi kolaylaştırır.

## 20.6. Alıştırmalar

1.  $4! = ?$
2.  $0! = ?$
3.  $\frac{6!}{4!} = ?$
4.  $P(5, 2) = ?$  (5 nesneden 2'sini sıralı)
5.  $C(5, 2) = ?$  (5 nesneden 2'sini seç)
6.  $C(4, 2) = ?$

### i Çözümler (önce kendin dene, sonra aç)

1.  $4 \cdot 3 \cdot 2 \cdot 1 = 24$
2. 1 (tanım gereği)
3.  $\frac{6!}{4!} = \frac{6 \cdot 5 \cdot 4!}{4!} = 6 \cdot 5 = 30$

20. Faktöriyel ve Sayma

$$4. P(5, 2) = \frac{5!}{3!} = 5 \cdot 4 = 20$$

$$5. C(5, 2) = \frac{5!}{2!3!} = \frac{5 \cdot 4}{2} = 10$$

$$6. C(4, 2) = \frac{4!}{2!2!} = \frac{4 \cdot 3}{2} = 6$$

---

*Bölüm 3 — Toplam, Diziler ve Sayma tamamlandı. Sonraki bölüm: Vektörler.*

**Kısım IV.**

**Bölüm 4 — Vektörler**



## 21. Vektör Kavramı

Sayı listesi, nokta ve ok — ML'in doğal nesnesi

Son bölümdeyiz. **Vektör**, sıralı bir sayı listesidir — ama makine öğrenmesinin doğal nesnesidir: bir veri örneği, bir embedding, bir nöronun ağırlıkları, hepsi birer vektördür. ML aslında yüksek boyutlu vektör uzayında yapılan geometridir.

### 21.1. Vektör nedir?

Bir vektör, belirli sırada yazılmış sayılardan oluşur:

$$\vec{v} = (v_1, v_2, \dots, v_n)$$

Aynı vektörü iki şekilde düşünebilirsin:

- **Nokta:** uzaydaki bir konum (koordinatlar).
- **Ok:** orijinden o noktaya giden, bir yönü ve uzunluğu olan ok.

İki bakış da aynı sayılardır:  $(3, 2)$  hem  $(3, 2)$  noktası, hem de oraya giden oktur.

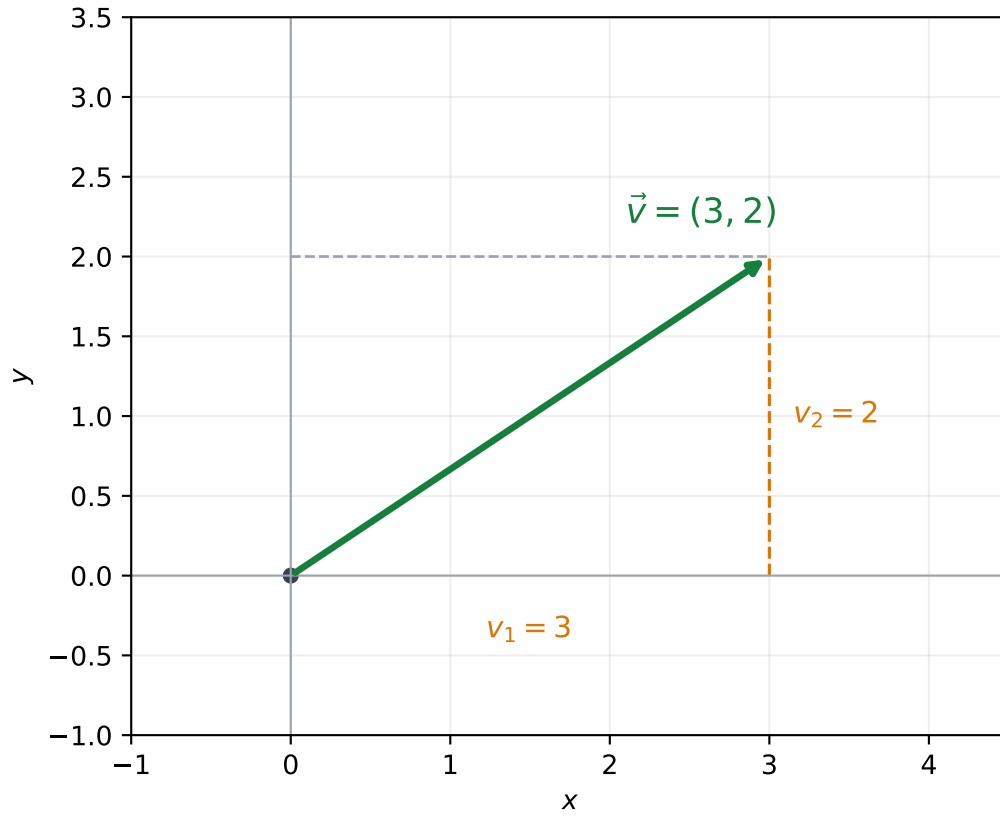
### 21.2. Boyut

Bir vektörün **boyutu**, bileşen sayısıdır.  $(3, 2)$  iki boyutlu,  $(1, 0, -2)$  üç boyutludur. ML'de boyut çok yüksektir: bir embedding yüzlerce, hatta binlerce bileşenli bir vektördür. 2B ve 3B çizebiliriz ama matematik her boyutta aynı çalışır.

### 21.3. Gösterim ve eşitlik

Vektör genelde üstünde ok ( $\vec{v}$ ) ya da kalın harfle yazılır; bileşenleri  $v_1, v_2, \dots$  ile gösterilir. İki vektör, ancak **tüm karşılıklı bileşenleri eşitse** eşittir — ve bunun için boyutları da aynı olmalıdır.

21. Vektör Kavramı



Şekil 21.1.:  $\vec{v} = (3, 2)$  vektörü: orijinden  $(3, 2)$  noktasına giden bir ok. Bileşenleri  $v_1 = 3$  (yatay),  $v_2 = 2$  (dikey).

! Aynı sayılar, iki resim

Bir vektör aynı anda hem bir **noktadır** (koordinatlar) hem de bir **oktur** (orijinden çıkan, yönü ve uzunluğu olan). Hangi resmin işine yaradığını probleme göre seçersin; sayılar değişmez.

⚠ İki tuzak

- Sıra önemlidir:  $(3, 2) \neq (2, 3)$ . Vektörün bileşenleri sıralıdır.
- İşlem yapmak veya karşılaştırmak için boyutlar eşleşmeli; 2B bir vektörle 3B bir vektör toplanamaz.

## 21.4. Örnek

**Örnek 1.**  $\vec{v} = (3, 2)$ : 2 boyutlu, bileşenleri  $v_1 = 3, v_2 = 2$ .

**Örnek 2.** Bir ev — alan  $120 \text{ m}^2$ , 3 oda, 10 yaş — şu vektörle gösterilebilir:  $(120, 3, 10)$ , yani 3 boyutlu bir veri noktası.

💡 ML köprüsü

Vektör, ML'in temel veri tipidir:

- **Veri örneği:** birden çok özelliğe sahip tek bir örnek bir vektördür — ev örneğindeki  $(120, 3, 10)$  gibi.
- **Embedding:** bir kelimenin, görüntünün veya belgenin anlamını kodlayan yüksek boyutlu bir vektördür. Bir RAG sistemi, belgeleri embedding vektörlerine çevirip bunları saklar ve arar — yani vektör uzayında benzerlik araması yapar.
- **Ağırlıklar:** bir nöronun ağırlıkları  $\vec{w}$  de bir vektördür (Toplam Sembolü dersindeki  $\sum w_i x_i$  aslında iki vektörün işlemidir — sıradaki derslerde göreceğiz).

ML, bu vektörler üzerinde yapılan geometridir.

## 21.5. Alıştırmalar

1.  $\vec{v} = (4, -1)$  kaç boyutludur?
2.  $\vec{u} = (2, 5, 1, 3)$  kaç boyutludur?
3.  $(3, 2)$  ile  $(2, 3)$  aynı vektör müdür?
4.  $\vec{v} = (3, 2)$  vektörünün birinci bileşeni  $v_1$  nedir?
5. Bir ev (alan 150, oda 4, yaş 8) hangi vektörle gösterilir ve kaç boyutludur?
6.  $\vec{a} = (1, 2)$  ile  $\vec{b} = (1, 2)$  eşit midir?

## 21. Vektör Kavramı

### **i** Çözümler (önce kendin dene, sonra aç)

1. 2 boyutlu (iki bileşen).
2. 4 boyutlu (dört bileşen).
3. Hayır; bileşenlerin sırası farklı,  $(3, 2) \neq (2, 3)$ .
4.  $v_1 = 3$ .
5.  $(150, 4, 8)$ , 3 boyutlu.
6. Evet; karşılıklı tüm bileşenler eşit.

---

Sonraki ders: Vektör İşlemleri.

## 22. Vektör İşlemleri

Toplama ve skaler çarpım — eğitim güncellemesinin ta kendisi

Vektörlerle iki temel işlem yaparız: **toplama** ve **skaler çarpım**. Tüm lineer cebir bunların üstüne kurulur — ve bir modeli eğitirken yapılan ağırlık güncellemesi tam olarak bu iki işlemden ibarettir.

### 22.1. Vektör toplama

İki vektörü toplamak için **karşılıklı bileşenleri** toplarsın:

$$\vec{u} + \vec{v} = (u_1 + v_1, u_2 + v_2, \dots)$$

Geometrik anlamı **uç-uca eklemektir**:  $\vec{v}$ 'yi  $\vec{u}$ 'nin ucuna koyarsın; toplam,  $\vec{u}$ 'nin başından  $\vec{v}$ 'nin ucuna giden oktur.

### 22.2. Skaler çarpım

Bir vektörü bir sayıyla (skalerle) çarpmak, **her bileşeni** o sayıyla çarpmaktır:

$$c\vec{v} = (cv_1, cv_2, \dots)$$

Geometrik anlamı, okun **uzunluğunu ölçeklemektir**:  $c = 2$  iki katı uzun,  $c = \frac{1}{2}$  yarısı;  $c < 0$  ise oku **ters yöne** çevirir. Örneğin  $2(3, 2) = (6, 4)$ ,  $-1(3, 2) = (-3, -2)$ .

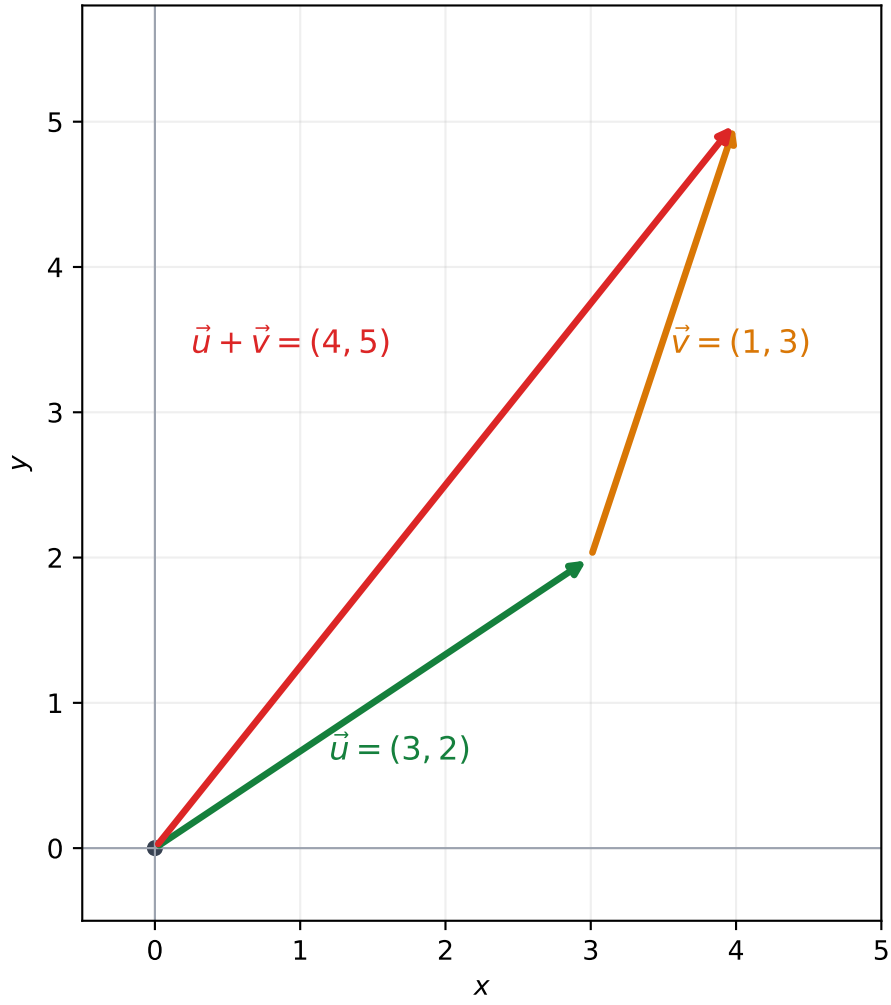
### 22.3. Vektör çıkarma

Çıkarma da bileşen bileşendir; aslında  $-1$  ile skaler çarpıp toplamaktır:

$$\vec{u} - \vec{v} = \vec{u} + (-1)\vec{v} = (u_1 - v_1, u_2 - v_2, \dots)$$

Örnek:  $(5, 4) - (2, 1) = (3, 3)$ .

## 22. Vektör İşlemleri



Şekil 22.1.: Vektör toplama (uç-uca):  $\vec{v}$ 'yi  $\vec{u}$ 'nin ucuna koy; toplam  $\vec{u} + \vec{v} = (4, 5)$ , orijinden son uca giden kırmızı oktur.

! Her şey bileşen bileşen

Hem toplama hem skaler çarpım **bileşen bazında** çalışır. Geometride: toplama = okları uç-uca ekleme, skaler çarpım = oku uzatma/kısaltma (ve negatifse ters çevirme).

! İki tuzak

- Yalnızca **aynı boyutlu** vektörler toplanır/çıkarılır.
- Skaler çarpım **her** bileşene uygulanır, sadece birine değil. Negatif skaler oku kısaltmaz, **yönünü ters çevirir** ( $c = -1$  tam ters yön).

## 22.4. Örnek

**Örnek 1 (toplama).**  $(2, 3) + (4, 1) = (6, 4)$ .

**Örnek 2 (skaler).**  $3(1, 2) = (3, 6)$ .

**Örnek 3 (çıkarma).**  $(5, 7) - (2, 3) = (3, 4)$ .

💡 ML köprüsü

Bir modeli eğitmenin kalbindeki **gradyan inişi güncellemesi** tam olarak bu iki işlemdir:

$$\vec{w} \leftarrow \vec{w} - \eta \nabla L$$

Burada  $\nabla L$  gradyan vektörü,  $\eta$  (öğrenme oranı) bir skalerdir. Yani: gradyanı bir skalerle çarp (skaler çarpım), sonucu ağırlık vektöründen çıkar (vektör çıkarma). Her eğitim adımı, ağırlık vektörünü gradyanın ters yönünde küçük bir adım kaydırmaktan ibarettir — bu derste öğrendiğin iki işlem. Embeddingleri birleştirmek veya ortalamak da yine vektör toplama ve skaler çarpımdır.

## 22.5. Alıştırmalar

1.  $(2, 3) + (4, 1) = ?$
2.  $3(1, 2) = ?$
3.  $(5, 7) - (2, 3) = ?$
4.  $-2(1, -3) = ?$
5.  $(1, 2, 3) + (4, 5, 6) = ?$
6.  $\frac{1}{2}(6, 4) = ?$

i Çözümler (önce kendin dene, sonra aç)

1.  $(2 + 4, 3 + 1) = (6, 4)$
2.  $(3 \cdot 1, 3 \cdot 2) = (3, 6)$
3.  $(5 - 2, 7 - 3) = (3, 4)$

## 22. Vektör İşlemleri

$$4. (-2 \cdot 1, -2 \cdot (-3)) = (-2, 6)$$

$$5. (1 + 4, 2 + 5, 3 + 6) = (5, 7, 9)$$

$$6. \left(\frac{1}{2} \cdot 6, \frac{1}{2} \cdot 4\right) = (3, 2)$$

---

*Sonraki ders: İç Çarpım.*

## 23. İç Çarpım

Bileşenleri çarp ve topla — bir nöronun hesabı

İç çarpım (nokta çarpımı), iki vektörü tek bir sayıya bağlayan işlemdir. Toplam Sembolü dersindeki  $\Sigma$ 'nın iki vektöre uygulanmış hâlidir — ve bir nöronun yaptığı hesabın ta kendisidir. Bu ders, foundation boyunca topladığımız parçaları tek formülde birleştirir.

### 23.1. İç çarpım nasıl hesaplanır?

Karşılıklı bileşenleri çarpıp toplarsın:

$$\vec{u} \cdot \vec{v} = u_1 v_1 + u_2 v_2 + \dots + u_n v_n = \sum_{i=1}^n u_i v_i$$

Örnek:  $(3, 2) \cdot (1, 4) = 3 \cdot 1 + 2 \cdot 4 = 3 + 8 = 11$ .

Dikkat: bu tam olarak Toplam Sembolü dersindeki  $\Sigma$ 'dır — sadece toplanan terim iki vektörün bileşenlerinin çarpımıdır.

### 23.2. Sonuç bir skalerdir

İç çarpımın sonucu bir **sayıdır** (skaler), vektör değil. Bu, onu vektör toplama ve skaler çarpımdan ayırır: o işlemler vektör üretti, iç çarpım tek bir sayı üretir.

### 23.3. Geometrik anlam: açı

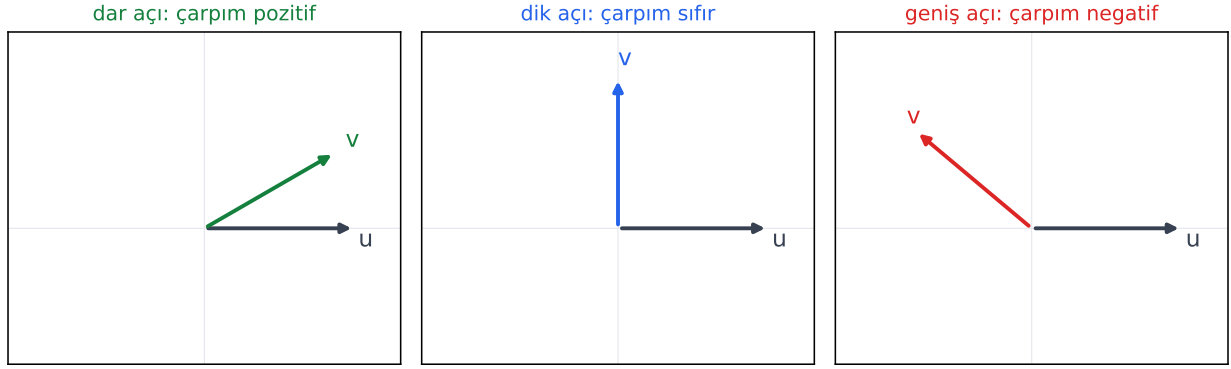
İç çarpımın bir de geometrik yüzü vardır. İki vektörün uzunlukları ve aralarındaki açı  $\theta$  cinsinden:

$$\vec{u} \cdot \vec{v} = |\vec{u}| |\vec{v}| \cos \theta$$

İşareti, vektörlerin birbirine göre yönünü söyler:

- Dar açı ( $< 90^\circ$ ) → pozitif: vektörler benzer yöne bakıyor.
- Dik açı ( $= 90^\circ$ ) → sıfır: vektörler birbirine dik (ilişkisiz).
- Geniş açı ( $> 90^\circ$ ) → negatif: vektörler zıt yöne bakıyor.

## 23. İç Çarpım



Şekil 23.1.: İç çarpımın işareti vektörler arası açığı söyler: dar açıda pozitif, dik açıda sıfır, geniş açıda negatif.

### 23.4. Nöronun hesabı

Şimdi parçaları birleştir. Bir nöronun çıktısı:

$$y = \vec{w} \cdot \vec{x} + b = \sum_{i=1}^n w_i x_i + b$$

Bu, Doğrusal Fonksiyon ve Eğim dersindeki tek-özellikli  $wx + b$ 'nin, Toplam Sembolü dersindeki  $\Sigma$ 'nin ve bu dersteki iç çarpımın aynı formülde buluşmasıdır.

! Tek sayı, çift anlam

İç çarpım = bileşen çarpımlarının toplamı = tek bir **skaler**. Bu sayının işareti, iki vektörün benzer yöne mi (pozitif), dik mi (sıfır), zıt yöne mi (negatif) baktığını söyler. Hesapsal yüzü “çarp ve topla”, geometrik yüzü “ne kadar aynı yöne bakıyorlar”.

! İki tuzak


- İç çarpımın çıktısı bir **skalerdir**, vektör değil — vektör toplamıyla karıştırmama.
- İç çarpım için boyutlar eşit olmalı. Ayrıca bileşen-bazlı çarpımla  $(u_1 v_1, u_2 v_2, \dots)$  — ki bu bir vektör kalır) karıştırmama; iç çarpım o çarpımları **toplar** ve sayıya iner.

### 23.5. Örnek

**Örnek 1.**  $(2, 5) \cdot (3, 1) = 2 \cdot 3 + 5 \cdot 1 = 6 + 5 = 11$ .

**Örnek 2 (dik).**  $(1, 0) \cdot (0, 1) = 0 \rightarrow$  vektörler birbirine dik.

**Örnek 3 (kendiyile).**  $(2, 3) \cdot (2, 3) = 4 + 9 = 13$  — bir vektörün kendisiyle iç çarpımı uzunluğunun karesidir (sıradaki ders).

 ML köprüsü

İç çarpım, sinir ağlarının en temel hesabıdır:

- **Nöron:**  $y = \vec{w} \cdot \vec{x} + b$  — bir nöron, ağırlık vektörü ile girdi vektörünün iç çarpımını alıp bias ekler. Tüm foundation buraya çıkıyor.
- **Katman:** bir katman, birçok nöronun iç çarpımıdır (matris-vektör çarpımı — ileride).
- **Dikkat (attention):** transformer'lerde hangi öğenin hangisiyle ne kadar ilgili olduğu, sorgu ve anahtar vektörlerinin iç çarpımıyla ölçülür — modern LLM'lerin çekirdeği.
- İç çarpım aynı zamanda **benzerliğin** temelidir; bir sonraki derste kosinüs benzerliğine bağlayacağız.

## 23.6. Alıştırmalar

1.  $(3, 2) \cdot (1, 4) = ?$
2.  $(2, 5) \cdot (3, 1) = ?$
3.  $(1, 0) \cdot (0, 1) = ?$  (sonuç ne diyor?)
4.  $(2, 3) \cdot (2, 3) = ?$
5.  $(1, 2, 3) \cdot (4, 5, 6) = ?$
6.  $(1, -2) \cdot (3, 1) = ?$

 Çözümler (önce kendin dene, sonra aç)

1.  $3 \cdot 1 + 2 \cdot 4 = 3 + 8 = 11$
2.  $2 \cdot 3 + 5 \cdot 1 = 6 + 5 = 11$
3.  $1 \cdot 0 + 0 \cdot 1 = 0 \rightarrow$  vektörler birbirine dik
4.  $2 \cdot 2 + 3 \cdot 3 = 4 + 9 = 13$
5.  $1 \cdot 4 + 2 \cdot 5 + 3 \cdot 6 = 4 + 10 + 18 = 32$
6.  $1 \cdot 3 + (-2) \cdot 1 = 3 - 2 = 1$

Sonraki ders: Vektör Uzunluğu ve Kosinüs Benzerliği.



## 24. Vektör Uzunluğu ve Kosinüs Benzerliği

Büyüklik ve yön — embedding ve RAG'ın motoru

Son derse geldik. Bir vektörün iki temel ölçüsü vardır: **uzunluğu** (ne kadar büyük) ve başka bir vektörle **yön benzerliği** (ne kadar aynı tarafa bakıyorlar). İkincisi — kosinüs benzerliği — embedding ve RAG sistemlerinin motorudur; yani bu ders, foundation'dan senin günlük işine giden yolu kapatır.

### 24.1. Vektör uzunluğu (norm)

Bir vektörün uzunluğu, okun boyudur ve Pisagor teoreminden gelir:

$$|\vec{v}| = \sqrt{v_1^2 + v_2^2 + \dots + v_n^2} = \sqrt{\vec{v} \cdot \vec{v}}$$

Sağdaki sadeleştirme İç Çarpım dersinden: bir vektörün kendisiyle iç çarpımı, uzunluğunun karesidir.

Örnek:  $|(3, 4)| = \sqrt{3^2 + 4^2} = \sqrt{9 + 16} = \sqrt{25} = 5$ .

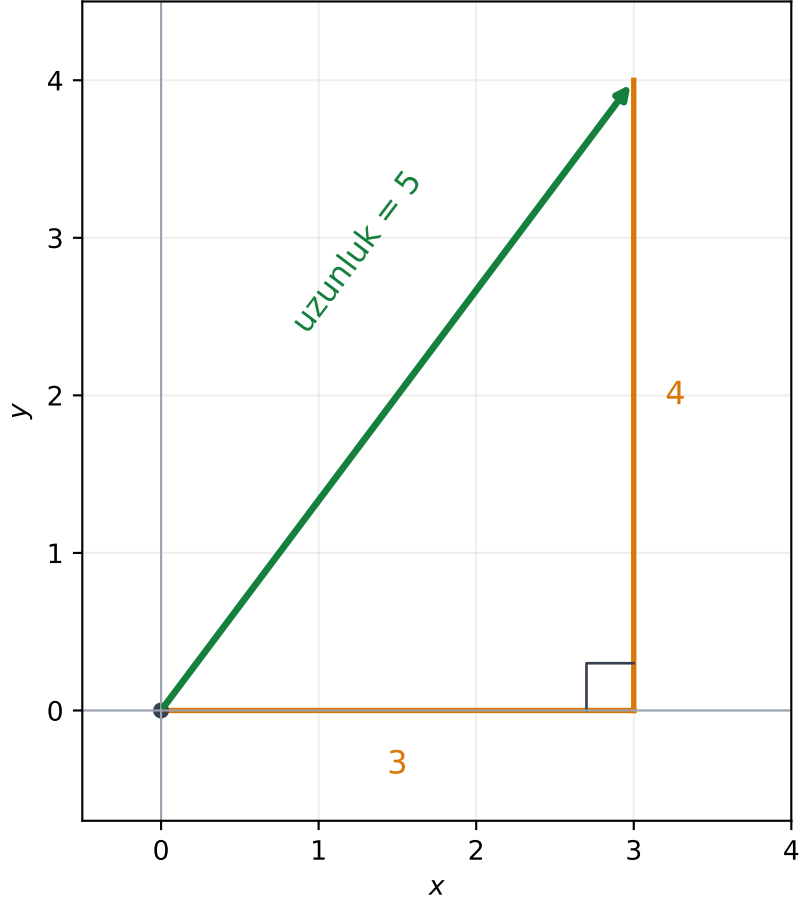
### 24.2. Birim vektör

Uzunluğu 1 olan vektöre **birim vektör** denir. Herhangi bir vektörü uzunluğuna bölerek birim vektöre çevirirsin (normalize):

$$\hat{v} = \frac{\vec{v}}{|\vec{v}|}$$

Bu, yönü korur ama uzunluğu 1'e sabitler. Örneğin  $(3, 4)$ 'ün birim vektörü  $(\frac{3}{5}, \frac{4}{5}) = (0,6, 0,8)$ .

24. Vektör Uzunluğu ve Kosinüs Benzerliği



Şekil 24.1.: Vektör uzunluğu Pisagor'dan gelir: (3,4) vektörünün bileşenleri 3 ve 4 dik kenarlar, uzunluğu hipotenüs = 5.

### 24.3. Kosinüs benzerliği

İki vektörün **yönlerinin** ne kadar benzer olduğunu, büyüklüklerinden bağımsız ölçer. İç Çarpım dersindeki  $\vec{u} \cdot \vec{v} = |\vec{u}||\vec{v}| \cos \theta$  formülünü  $\cos \theta$  için çözersin:

$$\cos \theta = \frac{\vec{u} \cdot \vec{v}}{|\vec{u}||\vec{v}|}$$

Değeri  $-1$  ile  $1$  arasındadır:

- $1 \rightarrow$  aynı yön (en benzer),
- $0 \rightarrow$  dik (ilişkisiz),
- $-1 \rightarrow$  zıt yön (en farklı).

Örnek:  $(3, 4)$  ve  $(4, 3)$  için iç çarpım  $= 24$ , uzunluklar  $5$  ve  $5$ , kosinüs benzerliği  $= \frac{24}{25} = 0,96$  — neredeyse aynı yön.

! Büyüklük mü, yön mü

**Uzunluk** vektörün ne kadar büyük olduğunu söyler; **kosinüs benzerliği** iki vektörün ne kadar aynı yöne baktığını — büyüklükten bağımsız olarak. Kosinüs benzerliği daima  $[-1, 1]$  aralığındadır.

! İki tuzak

- Uzunluk için karekök şart:  $\vec{v} \cdot \vec{v}$  uzunluğun **karesidir**, uzunluğun kendisi değil.
- Kosinüs benzerliği **yönü** ölçer, mesafeyi değil. Birbirinden çok uzak iki nokta bile aynı yöne bakıyorsa kosinüs benzerlikleri  $1$  olabilir.

### 24.4. Örnek

**Örnek 1 (uzunluk).**  $|(6, 8)| = \sqrt{36 + 64} = \sqrt{100} = 10$ .

**Örnek 2 (kosinüs).**  $(2, 0)$  ile  $(4, 0)$ : iç çarpım  $8$ , uzunluklar  $2$  ve  $4$ ;  $\cos \theta = \frac{8}{2 \cdot 4} = 1$  — uzunlukları farklı olsa da tam aynı yön.

💡 ML köprüsü

Kosinüs benzerliği, **embedding ve RAG sistemlerinin çekirdeğidir**:

- Bir RAG sistemi, sorguyu ve belgeleri birer **embedding vektörüne** çevirir, sonra sorguya en yakın belgeleri **kosinüs benzerliğine** göre sıralayıp getirir. Bir vektör veritabanının (örneğin senin Qdrant bilgi tabanının) yaptığı tam olarak budur — yüksek boyutlu vektörler arasında yön benzerliği araması.
- Neden mesafe değil de kosinüs? Çünkü anlam yönde kodlanır: aynı konudaki iki belge, uzunlukları ne olursa olsun aynı yöne bakar.
- Embeddingler çoğu zaman birim vektöre normalize edilir; o zaman kosinüs benzerliği doğrudan iç çarpıma eşit olur (payda  $1$ ).

## 24. Vektör Uzunluğu ve Kosinüs Benzerliği

Böylece tüm yol kapanıyor: **sayı** → **fonksiyon** → **vektör** → **iç çarpım** → **benzerlik** → **getirme**.  
Foundation'ın son tuğlası, senin sistemlerinin temel işlemiyle birleşti.

### 24.5. Alıştırmalar

1.  $|(3, 4)| = ?$
2.  $|(6, 8)| = ?$
3.  $|(1, 2, 2)| = ?$
4.  $(3, 4)$  vektörünün birim vektörü nedir?
5.  $(1, 0)$  ile  $(0, 1)$  arasındaki kosinüs benzerliği nedir? (ne anlama gelir?)
6.  $(2, 0)$  ile  $(4, 0)$  arasındaki kosinüs benzerliği nedir?

**i** Çözümler (önce kendin dene, sonra aç)

1.  $\sqrt{3^2 + 4^2} = \sqrt{25} = 5$
2.  $\sqrt{6^2 + 8^2} = \sqrt{100} = 10$
3.  $\sqrt{1 + 4 + 4} = \sqrt{9} = 3$
4.  $\frac{(3, 4)}{5} = \left(\frac{3}{5}, \frac{4}{5}\right) = (0,6, 0,8)$
5. İç çarpım = 0, uzunluklar 1 ve 1,  $\cos \theta = 0 \rightarrow$  dik, ilişkisiz.
6. İç çarpım = 8, uzunluklar 2 ve 4,  $\cos \theta = \frac{8}{8} = 1 \rightarrow$  aynı yön (uzunluk farkı önemsiz).

---

*Phase 0 — Temel Matematik tamamlandı. Buradan sonra: lineer cebir, kalkülüs ve olasılık sütunları — hepsi bu temelin üstüne kurulur.*

## 25. Machine Learning Bağlantısı

Temelden modele — tüm parçaların birleştiği yer

Bu kursun finali. Her derste bir “ML köprüsü” gördün; burada onları tek resimde birleştiriyoruz: kurduğun temel, nasıl çalışan bir makine öğrenmesi modeline dönüşür? Yeni alıştırma yok — büyük harita.

### 25.1. Model bir fonksiyondur

Bir model, girdi alır (özellikler bir vektör  $\vec{x}$ ) ve çıktı üretir (tahmin  $\hat{y}$ ). Yani model, sonuçta bir **fonksiyondur** (Bölüm 2). Eğitmek, veriye en iyi uyan fonksiyonu bulmaktır.

### 25.2. Bir nöron: dört bölümün buluştuğu nokta

Tek bir yapay nöronun hesabı, bütün foundation’ı tek satırda toplar:

$$\hat{y} = f(\vec{w} \cdot \vec{x} + b)$$

Parçalara ayır:

- $\vec{x}$  → girdi **vektörü** (Bölüm 4).
- $\vec{w} \cdot \vec{x}$  → **iç çarpım** =  $\sum_i w_i x_i$  (Bölüm 3  $\Sigma$  + Bölüm 4 iç çarpım).
- $+b$  → bias, yani  $wx + b$ ’deki **kesişim** (Bölüm 2 doğrusal fonksiyon).
- $f(\cdot)$  → aktivasyon, çoğu zaman  $e^{-x}$  içeren **sigmoid** (Bölüm 2 üstel).

Yani tek bir nöron, dört bölümün de kavramını kullanır.

### 25.3. Öğrenme: hatayı azaltmak

- **Kayıp (loss):** modelin ne kadar yanıldığı.  $MSE = \frac{1}{n} \sum_i (y_i - \hat{y}_i)^2$  — kare hataların ortalaması ( $\Sigma$  + kare).
- Doğrusal regresyonda (aktivasyonsuz,  $\hat{y} = \vec{w} \cdot \vec{x} + b$ ) kayıp, ağırlıkların **konveks bir çukurudur** (parabol); en dipte tek bir en iyi parametre durur. Nöronda sigmoid gibi doğrusal-olmayan bir aktivasyon olunca yüzey genelde konveks değildir, ama fikir aynı: en düşük kayba inmek.
- **Gradyan inişi:**  $\vec{w} \leftarrow \vec{w} - \eta \nabla L$  — gradyanı skalerle çarp, ağırlıktan çıkar (Bölüm 4 skaler çarpım + vektör çıkarma). Adım adım çukurun dibine yuvarlanır.

## 25. Machine Learning Bağlantısı

- **Olabilirlik**: olasılıkların **çarpımıdır** ( $\prod$ ); logaritma alınca toplama döner ( $\sum \log$ ), sayısal olarak kararlı log-olabilirlik / log-loss çıkar (çarpım notasyonu + logaritma +  $\Sigma$ ).

### 25.4. Derin ağ: fonksiyon bileşkesi

Bir sinir ağı katmanları üst üste bindirir: çıktı =  $f_n(\dots f_1(\vec{x}))$  — **fonksiyon bileşkesi** (Bölüm 2). Eğitimdeki geri yayılım (backprop), bu bileşkenin türevini **zincir kuralıyla** alır.

### 25.5. Benzerlik ve getirme: embedding / RAG

- Bir **embedding**, anlamı kodlayan yüksek boyutlu bir **vektördür** (Bölüm 4).
- **Kosinüs benzerliği**  $\frac{\vec{u} \cdot \vec{v}}{|\vec{u}| |\vec{v}|}$  iki embedding'in ne kadar ilgili olduğunu sıralar (norm + iç çarpım).
- Bir RAG sistemi (örneğin senin Qdrant tabanının) tam olarak bunu yapar: sorgu ve belgeleri vektöre çevirip kosinüs benzerliğiyle en yakınları getirir.

### 25.6. Foundation → ML haritası

Temel konu	ML'deki karşılığı
Değişken / denklem (bilinmeyeni izole etme)	parametre öğrenme, optimizasyon
Doğrusal fonksiyon $wx + b$	lineer regresyon, tek nöron
Üstel / logaritma	sigmoid, softmax, log-loss
Parabol (konveks çukur)	kayıp yüzeyi, minimuma iniş
Toplam sembolü $\sum$	ortalama, MSE, iç çarpım
Çarpım $\prod$ + logaritma	olabilirlik → log-olabilirlik
Fonksiyon bileşkesi	derin ağ katmanları, backprop
Diziler / sayma	veri indekisleme, olasılık temeli
Vektör + iç çarpım	embedding, nöron hesabı $\vec{w} \cdot \vec{x}$
Norm + kosinüs benzerliği	embedding araması, RAG getirme

#### ! Tek cümlede

Bir model bir fonksiyondur; eğitim, onun parametrelerini optimize etmektir. Bu kursta öğrendiğin her parça, o mozaiğin bir taşı — ve hepsi tek bir nöronun  $\hat{y} = f(\vec{w} \cdot \vec{x} + b)$  satırında buluşuyor.

#### 💡 Buradan sonra

Üç sütun bu temelin üstüne kurulur ve her parçayı derinleştirir:

- **Lineer cebir** — vektörlerden matrislere; bir katman = matris-vektör çarpımı.
- **Kalkülüs** — türev ve gradyan; gradyan inişinin ve zincir kuralının motoru.
- **Olasılık ve istatistik** — sayma ve dağılımlar; kaybın ve belirsizliğin dili.

Artık bu sütunları okuyacak temelin var.

## 25.7. Alıştırmalar

1. Bir nöronun  $\vec{w} \cdot \vec{x}$  hesabı hangi iki temel konuyu birleştirir?
2. Sigmoid aktivasyonu hangi temel fonksiyonu içerir?
3. Gradyan inişi güncellemesi  $\vec{w} \leftarrow \vec{w} - \eta \nabla L$  hangi iki vektör işlemini kullanır?
4. Doğrusal regresyonda MSE kaybı neden parabol biçimindedir ve bu optimizasyona ne kazandırır?
5. Bir RAG sisteminde belgeler sorguya göre neye bakılarak sıralanır?

**i** Çözümler (önce kendin dene, sonra aç)

1. Toplam sembolü  $\Sigma$  ile vektör iç çarpımı ( $\vec{w} \cdot \vec{x} = \sum_i w_i x_i$ ).
2. Üstel fonksiyonu —  $\sigma(x) = \frac{1}{1+e^{-x}}$ .
3. Skaler çarpım ( $\eta \cdot \nabla L$ ) ve vektör çıkarma.
4. Doğrusal regresyonda kayıp konveks (tek minimumlu) bir çukur olur; bu, tek bir en iyi noktaya inmeyi ve optimizasyonu kolaylaştırır. (Aktivasyonlu/derin modellerde yüzey konveks olmayabilir, ama gradyan inişi sezgisi geçerli kalır.)
5. Kosinüs benzerliğine göre (sorgu ve belge embedding vektörleri arasında).

---

*Phase 0 — Temel Matematik tamamen tamamlandı. Sıradaki: lineer cebir, kalkülüs ve olasılık sütunları.*

