Calculus — 3Blue1Brown'ın Özünden

ML Builder için Türkçe Notlar

Phase 1

2026-05-29

İçindekiler

Önsöz

0.1 Bu kitap nedir?

Bu, **3Blue1Brown — Essence of Calculus** dizisinin (Grant Sanderson, 12 bölüm) Türkçe ders notlarıdır. Hedef, izlerken paralel okunabilecek; sonradan tek başına da yeterli olabilecek bir referans seti üretmek.

Her bölüm bir "Builder Notu" katmanı taşır: kavramın **makine öğrenmesi** ile köprüsü. Türev → gradient → backprop; integral → beklenen değer → Monte Carlo; FTC → forward/backward dualitesi. Calculus'u "tek başına matematik" olarak değil, ML'i türeten alet kutusu olarak okuyoruz.

> **Kaynak**
>
> - **Video dizisi:** 3Blue1Brown — Essence of Calculus (YouTube)
> - **Yazar:** Grant Sanderson — 3blue1brown.com
> - **Çeviri ve genişletme:** Phase 1 (TR + ML köprüleri)

0.2 Nasıl Okumalı

Sıralı oku. Her ders bir öncekinin **dilini** kullanır. Atlamak istersen, en azından *Ders 1 (Calculus'un Özü)* ve *Ders 2 (Türevin Paradoksu)* zorunlu — bu ikisi tüm seriyi taşıyor.

> **Pratik bir tavsiye**
>
> Her bölüm sonundaki **egzersizleri** atlama. Özellikle Python egzersizleri (Riemann toplamı, finite difference, Taylor serisi) calculus sezgisini parmaklarına yerleştirir. ML'de aynı kodu yıllarca farklı kılıklarda yazacaksın.

0.3 12 Ders

#	Ders	Ana Fikir
1	Calculus'un Özü	İntegral + türev + FTC, dairenin alanından çıkar
2	Türevin Paradoksu	Anlık değişim oranı; $d(x^2)/dx = 2x$'in geometrisi
3	Geometriyle Türev Formülleri	Trigonometri, x^n türevi
4	Zincir Kuralı ve Çarpım Kuralı	Backprop'un matematiksel kalbi
5	Euler Sayısı e^x'in Özelliği	$d(e^x)/dx = e^x$ — neden?
6	Kapalı (Implicit) Türev	Eğriler ve teğet doğrular

#	Ders	Ana Fikir
7	Limitler, L'Hôpital, Epsilon-Delta	"Yaklaşıktan kesine" rigorize
8	İntegrasyon ve Temel Teorem	Antiderivative
9	Alan ile Eğim Arasındaki İlişki	FTC'nin geometrisi
10	Yüksek Mertebeden Türevler	İvme, jerk, optimizasyon
11	Taylor Serileri	Sonsuz polinom yaklaşımı
12	Calculus'ta Sana Öğretmedikleri	Riemann'ın ötesi

0.4 Notasyon

- **Türev:** $f'(x)$ veya $\frac{df}{dx}$ — ikisi de aynı şey
- **İkinci türev:** $f''(x)$ veya $\frac{d^2f}{dx^2}$
- **İntegral:** $\int_a^b f(x)\,dx$ — a'dan b'ye eğri altı alan
- **Limit:** $\lim\limits_{x \to a} f(x)$

Tüm matematik KaTeX ile render ediliyor.

❗ Bir tek şey

Calculus iki büyük fikre dayanır: **integral** (küçük niceliklerin toplamı = eğri altı alan) ve **türev** (anlık değişim oranı). FTC bu ikisinin birbirinin tersi olduğunu söyler. Geri kalan her şey bunun varyasyonu.

1 Calculus’un Özü

Daireden integrale + türeve + Calculus’un Temel Teoremine

ℹ Bölüm bilgisi

- **Grant’ın videosu:** YouTube — Chapter 1: The Essence of Calculus (≈17 dk)
- **Kaynak:** 3Blue1Brown — Essence of Calculus
- **Okuma süresi:** ≈22 dk

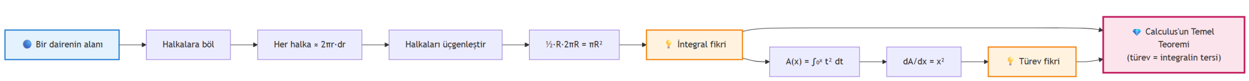
1.1 Bu Derste Ne Var?

Calculus genelde bir yığın kural ve formülle gelir: türev formülleri, çarpım kuralı, zincir kuralı, kapalı türev, integralle türevin zıt olması, Taylor serileri… ve bunlar ezberlenecek şeyler gibi sunulur. Grant’ın hedefi bambaşka: dersin sonunda *“bunu **ben de icat edebilirdim**”* hissiyle kalmanı.

Bu ilk bölümde tek bir geometri parçasını — **bir dairenin alanını** — derinlemesine düşünerek calculus’un üç büyük fikrine birden tökezliyoruz. πr^2 formülünü biliyorsun; ama nereden geldiğini güzel bir yolla görmek, seni doğrudan calculus’un kalbine götürüyor.

Üç büyük fikir:

1. **İntegral** — bir eğrinin altındaki alan; “çok sayıda küçük nicelik”in toplamı.
2. **Türev** — anlık değişim oranı; bir fonksiyonun girdideki küçük dürtüşe ne kadar duyarlı olduğu.
3. **Bu ikisinin ters olması** — Calculus’un Temel Teoremi (FTC): türev ve integral birbirinin tersi.



Şekil 1.1: Bu bölümün kavram haritası — bir dairenin alanından FTC’ye

> *“my goal is for you to come away feeling like you could have invented calculus yourself.”* — Grant, 0:54

Builder Notu — ML Köprüleri

- **Türev = gradient + backprop’un temeli.** Bir loss fonksiyonunun her ağırlığa duyarlılığı ($dLoss/dw$) bir türevdir; gradient descent bu oranın ters yönünde adım atar.
- **İntegral = beklenen değer / Monte Carlo.** $E[f(X)] = \int f(x)\,p(x)\,dx$ — sürekli bir toplama.

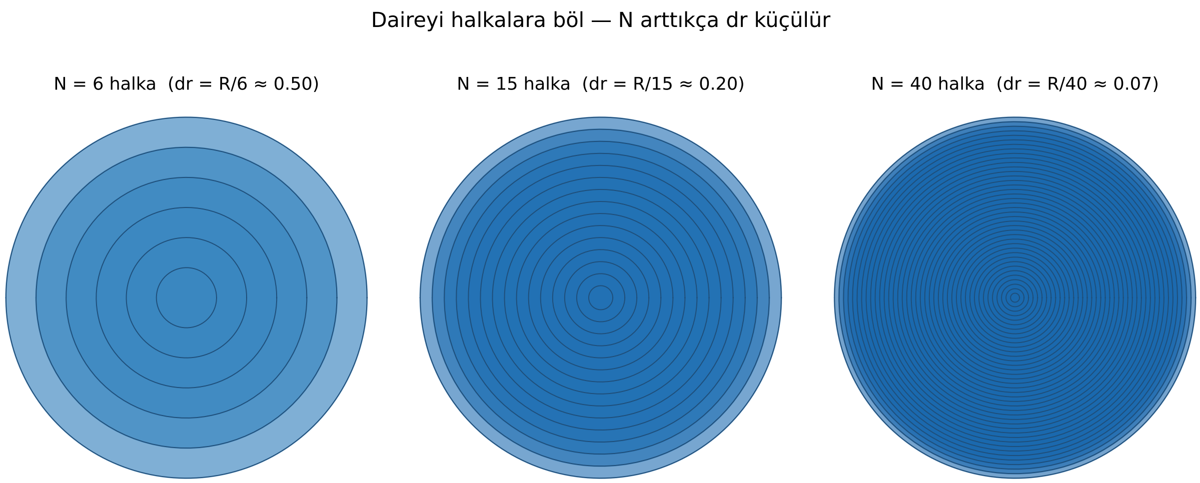
Çok örneğin ortalaması bu integrali tahmin eder.
- **Yaklaşıktan kesine geçiş** ($dr \to 0$) **= limit.** Sürekli optimizasyonun, gradient flow’un ve neural ODE’lerin matematiksel zemini.
- **Türev↔integral terslik (FTC)** → RL’de kümülatif ödül (value function) ile anlık ödül arasındaki ilişki; forward/backward dualitesi.
- **“Kendin keşfedebilirdin” = builder zihniyeti.** Formülü ezberleme, nereden geldiğini gör — her ML tekniğine böyle bak.

1.2 Tek Bir Soru: Bir Dairenin Alanı

Hikaye basit başlıyor: sen ve bir daire — diyelim yarıçapı 3. Alanını bulmaya çalışıyorsun. Bir sürü kâğıt harcayıp alanı farklı şekillerde parçalayıp yeniden dizdikten sonra, umut verici bir fikir deniyorsun: daireyi **eşmerkezli (iç içe) halkalara** dilimlemek.

Bu neden umut verici? Çünkü dairenin **simetrisine saygı gösteriyor** — ve matematik, simetrisine saygı gösterdiğinde seni ödüllendirme eğilimindedir.

> *“math has a tendency to reward you when you respect its symmetries.”* — Grant, 2:24



Şekil 1.2: Daireyi eşmerkezli halkalara böl: her halkanın çevresi $2\pi r$, kalınlığı dr. Halka sayısı arttıkça kalınlık küçülür.

Halkalardan birini al: iç yarıçapı r olsun, 0 ile 3 arasında bir yerde. Plan şu: **(1)** her halkanın alanı için temiz bir ifade bulmak, **(2)** hepsini düzgünce toplamanın bir yolunu bulmak. Bu iki parça bir araya gelince, tüm dairenin alanı ortaya çıkacak.

Builder Notu — Simetri ve Invariance

Simetriye saygı — yani problemi “doğru koordinatlarda” parçalamak — ML’de **invariance/equivariance** olarak karşımıza çıkar. CNN’ler öteleme değişmezliğinden güç alır (kediyi nerede görürsen gör, kedidir); görev simetrisine uygun bir mimari/parametreleme seçmek, öğrenmeyi dramatik biçimde kolaylaştırır.

Grant'ın "doğru dilimleme" sezgisi tam da budur: problemin yapısına uygun ayrıştırma, çözümü kendiliğinden getirir.

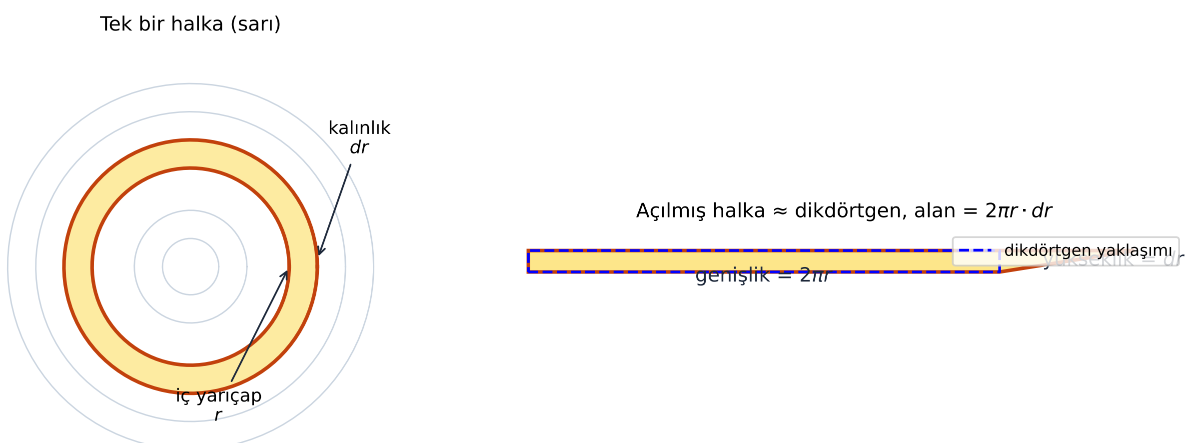
1.3 Halkayı Açmak: $2\pi r \cdot dr$

Bir halkayı al ve düzleştir; düz bir şeride dönüştür. Bu yeni şeklin tam ne olduğunu düşünebilirsin, ama basitlik için onu bir **dikdörtgen** olarak yaklaştıralım. Dikdörtgenin genişliği, orijinal halkanın çevresidir: $2\pi r$ (zaten π'nin tanımı bu). Kalınlığı ise daireyi ne kadar ince dilimlediğine bağlı — şu an keyfî bir seçim. Standart calculus notasyonuyla bu kalınlığa dr diyelim: yarıçaptaki sonsuz küçük bir fark, örneğin $0{,}1$.

> *"let's call that thickness dr, for a tiny difference in the radius from one ring to the next."* — Grant, 3:20

Böylece açılmış halkanın alanı, yaklaşık olarak genişlik çarpı kalınlık:

$$A_{\text{halka}} \approx 2\pi r \, dr$$



Şekil 1.3: Bir halkayı 'aç' — neredeyse bir dikdörtgen olur. Genişliği $2\pi r$, yüksekliği dr. $dr \to 0$ iken hata kaybolur.

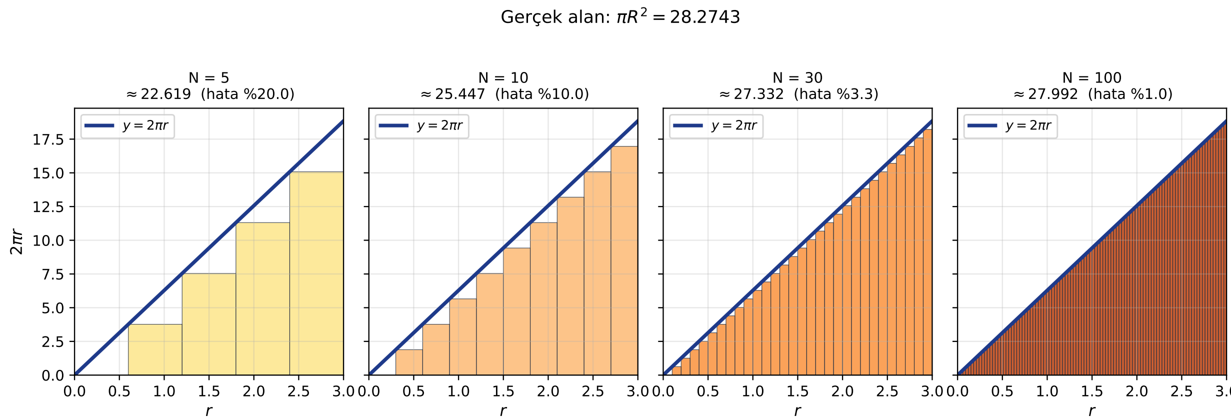
Bu tam doğru değil — şerit gerçek bir dikdörtgen değil, hafifçe yamuk. Ama dr küçüldükçe yaklaşım **giderek daha iyi** olur: şeridin üst ve alt kenar uzunlukları ($2\pi r$ ile $2\pi(r + dr)$) birbirine yaklaşır, aradaki fark $2\pi \cdot dr$ ile orantılı biçimde sıfıra gider.

Builder Notu — Yerel Doğrusallaştırma

"Eğri parçayı küçük ölçekte düz say" — bu **yerel doğrusallaştırma** (local linearization), calculus'un en sık tekrar eden hilesidir. Türev tam olarak budur: bir fonksiyonu bir nokta civarında en iyi doğru ile değiştirmek. ML'de bir loss yüzeyini gradient (birinci derece terim) ile yerel olarak düzleştiririz; her gradient descent adımı, "yeterince küçük bir komşulukta yüzey neredeyse düzdür" varsayımına dayanır.

1.4 Yaklaşıktan Kesine: Üçgenin Alanı

Şimdi bu dikdörtgenleri, $r = 0$'dan $r = 3$'e kadar bir eksen boyunca yan yana dik diz. Her birinin kalınlığı dr, yüksekliği ise o yarıçaptaki $2\pi r$ değeri. Yükseklikler $2\pi r$ olduğundan, dikdörtgenlerin uçları **eğimi** 2π **olan bir doğrunun** üstünde oturur: $y = 2\pi r$ grafiği.



Şekil 1.4: Dikdörtgenler arttıkça (yani $dr \to 0$), toplam alan $y = 2\pi r$ doğrusu altındaki üçgene yakınsar. Bu, integral fikrinin kendisidir.

Şimdi güzel kısım: dr küçüldükçe dikdörtgen sayısı patlar, ama hepsinin toplam alanı yalnızca **bu doğrunun altındaki alana** benzer. O bölge bir üçgendir — tabanı 3, yüksekliği $2\pi \cdot 3$.

$$\frac{1}{2} \cdot 3 \cdot (2\pi \cdot 3) = 9\pi = \pi \cdot 3^2$$

Yarıçap genel olarak R olsaydı, aynı üçgen argümanı:

$$\frac{1}{2} \cdot R \cdot (2\pi R) = \pi R^2$$

İşte dairenin alan formülü — ezberlenecek bir şey değil, halkaları toplamanın doğal sonucu.

İşin **kalbi** şu ince geçişte: problem, her biri $2\pi r \cdot dr$ gibi görünen çok sayıda küçük sayının toplamıyla **yaklaşık** çözülebiliyordu. dr küçüldükçe (daireyi giderek daha ince halkalara böldükçe) bu toplam, grafik altındaki **kesin** alana yakınsıyor. Yaklaşıktan kesine bu geçiş, calculus'un ne olduğuna dair en derin fikirdir.

> *"the way that we transitioned from something approximate to something precise … cuts deep to what calculus is all about."* — Grant, 7:08

Builder Notu — Riemann → Monte Carlo

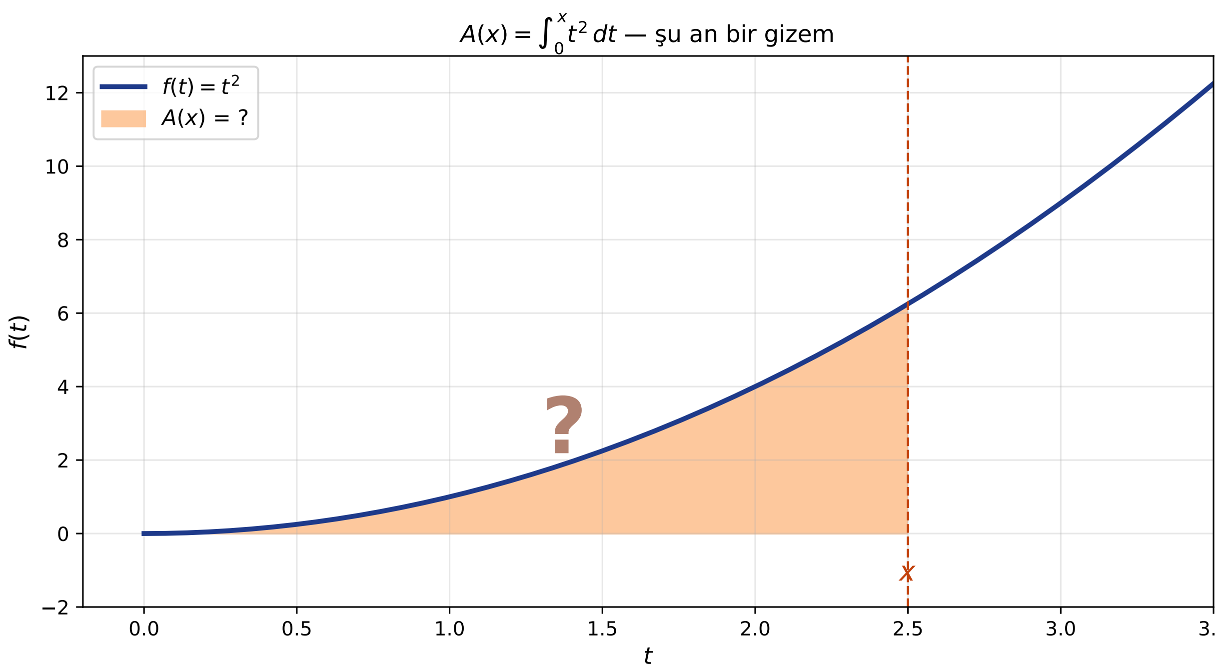
Bu tam olarak bir **Riemann toplamı → integral** geçişidir. ML'de beklenen değer $E[f(X)] = \int f(x)\,p(x)\,dx$, **Monte Carlo** ile tahmin edilir: çok sayıda örneğin ortalaması = küçük katkıların toplamı; örnek sayısı $N \to \infty$ iken gerçek integrale yakınsar (tıpkı $dr \to 0$ gibi). Policy gradient, ELBO ve difüzyon kayıpları hep bu "küçük niceliklerin toplamı → integral" iskeletine dayanır.

1.5 İntegral: Başka Eğrilerin Altındaki Alan

Daire problemi tek başına güzel; ama asıl kazanç, **yöntemin** kendisi. Matematikte ve bilimde pek çok zor problem, çok sayıda küçük niceliğin toplamı olarak yaklaştırılabilir — ve bu toplamlar çoğu zaman **bir grafiğin altındaki alan** sorusuna dönüşür.

Örnek: bir arabanın kat ettiği mesafeyi, her andaki hızından bulmak istiyorsun. Zamanı küçük dt aralıklarına böl; her aralıkta mesafe $\approx v(t) \cdot dt$. Hepsini topla → kat edilen toplam mesafe = **hız grafiğinin altındaki alan**. Daire ile birebir aynı yapı.

Şimdi unutamayacağın bir soru: **parabol**, yani x^2 grafiği. 0 ile x arasında bu eğrinin altındaki alan nedir? Sol ucu 0'da sabitleyip sağ ucu serbest bırakalım. x'e bağlı bir alan fonksiyonu $A(x)$ bulabilir misin? Böyle bir $A(x)$'e x^2**'nin integrali** denir.



Şekil 1.5: Gizem fonksiyonu $A(x) = f(t) = t^2$ eğrisinin altındaki, 0'dan x'e kadar olan alan. Şu an ne olduğunu bilmiyoruz.

Calculus bunu bulmanın araçlarını içerir; ama şu an $A(x)$ bizim için bir gizem — yalnızca x^2 altındaki alanı verdiğini biliyoruz, ne olduğunu değil.

> *"finding this area, this integral function, is genuinely hard."* — Grant, 11:14

Zor bir soruyla karşılaşınca Grant'ın tavsiyesi: doğrudan cevaba saldırma; fikirle **oyna**, $A(x)$ ile x^2 arasındaki ilişkiye aşinalık kur.

> *"a good policy is to not try too hard to get at the answer directly … play around with the idea."*
> — Grant, 11:22

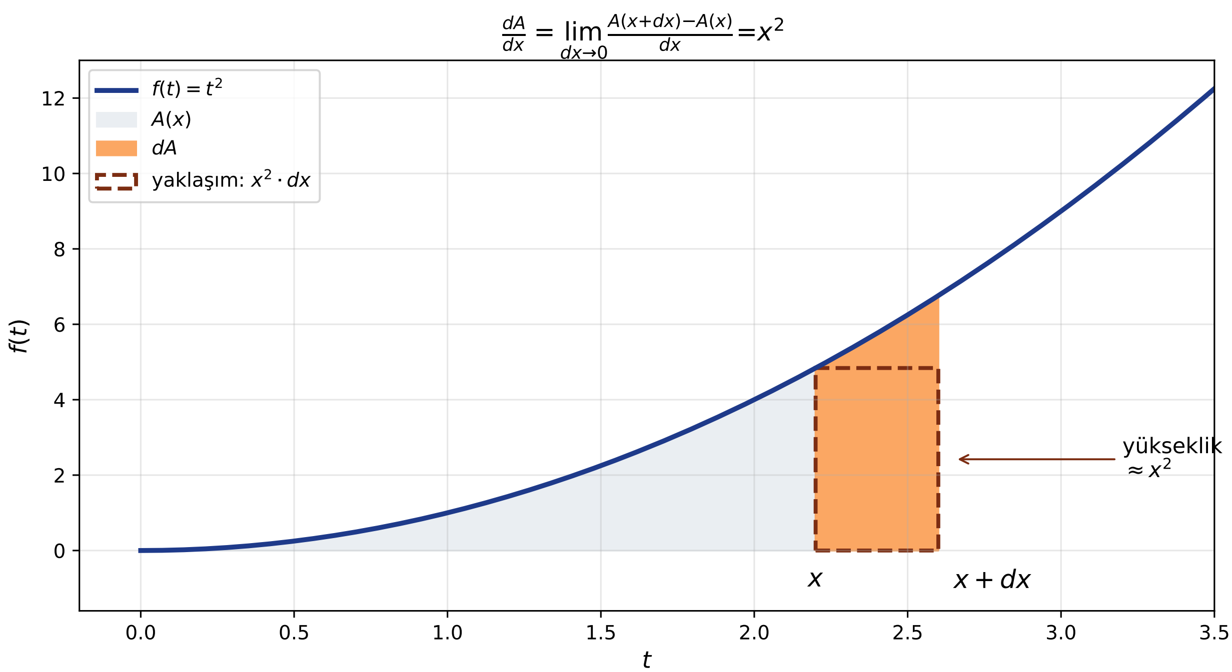
Builder Notu — Biriktirme

İntegral bir **biriktirmedir** (accumulation). "$v \cdot dt$'leri topla → mesafe", sayısal entegrasyonun (Euler yöntemi) ve onun sürekli limitinin tam olarak nasıl çalıştığıdır: bir neural ODE'de durum $= \int$ dinamik dt. Yine beklenen değer $E[f(X)] = \int f \cdot p\, dx$ de bir integraldir — modellerde "ortalama davranışı" hesaplamak, küçük katkıları biriktirmek demektir.

1.6 Türev: dA/dx Oranı

Gizem fonksiyonu $A(x)$ ile oynamaya devam edelim. x'i küçük bir dx kadar **dürt**. Alanda ortaya çıkan değişim, ince bir dilimdir — buna dA diyelim. Bu dilim, yüksekliği x^2 ve genişliği dx olan bir dikdörtgenle iyi yaklaştırılabilir. Yani $dA \approx x^2 \cdot dx$; yeniden düzenlersek $dA/dx \approx x^2$. dx küçüldükçe bu yaklaşım daha da iyi olur.

$$\frac{dA}{dx} \approx \frac{A(x+dx) - A(x)}{dx} \approx x^2$$



Şekil 1.6: x'i dx kadar dürt: ortaya çıkan ince dilim ≈ yüksekliği x^2, genişliği dx olan bir dikdörtgen. $dA/dx \to x^2$ olduğunu görmek bu kadar.

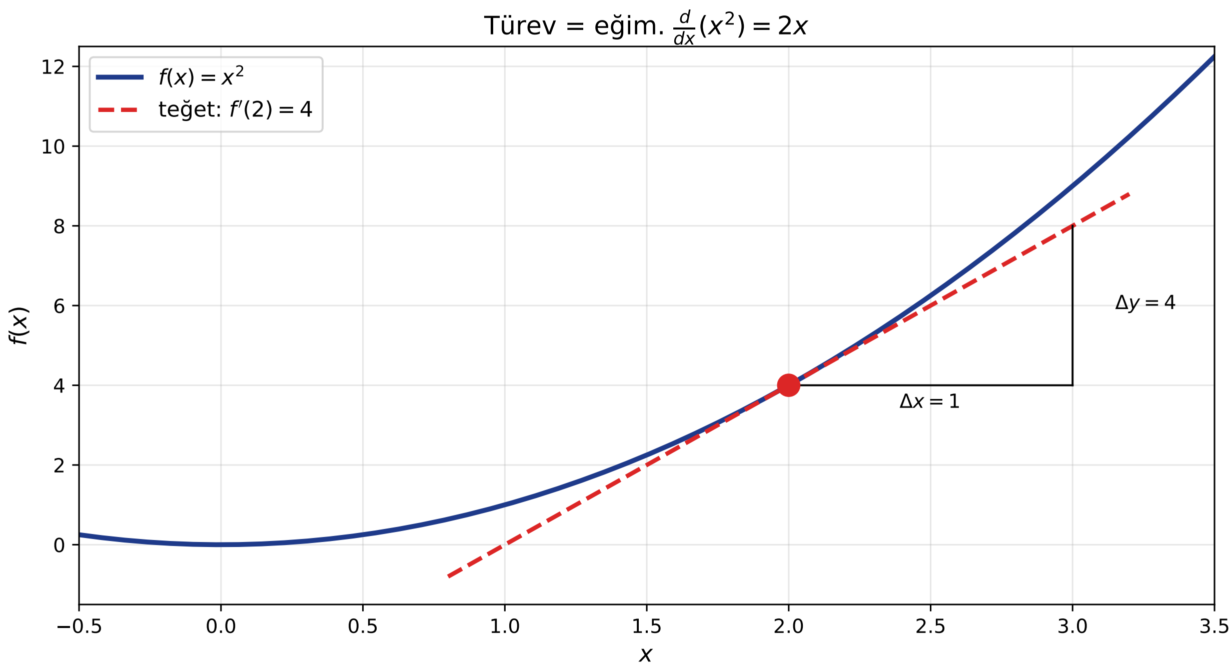
Dikkat: $A(x)$'in ne olduğunu hâlâ bilmiyoruz — ama sağlaması gereken bir **özellik** bulduk. Somut örnek: 3 ile 3,001 noktalarında $[A(3{,}001) - A(3)]/0{,}001$ oranı, yaklaşık $3^2 = 9$ olmalı. Ve bu her x için geçerli, yalnızca 3 için değil.

Aslında özel olan x^2 değil: **bir grafiğin altındaki alan** olarak tanımlı her fonksiyonun bu özelliği vardır — A'ya verilen küçük bir dürtüşün, onu tetikleyen dx'e oranı, o noktadaki grafiğin yüksekliğine eşittir. Bu oranın ($dx \to 0$ iken yaklaştığı değer) bir adı var: A'nın **türevi**.

$$\frac{dA}{dx} = \lim_{dx \to 0} \frac{A(x+dx) - A(x)}{dx}$$

Gevşek bir ifadeyle türev, bir fonksiyonun girdisindeki küçük değişimlere ne kadar **duyarlı** olduğunun ölçüsüdür.

> *"the derivative … loosely speaking it's a measure of how sensitive a function is to small changes in its input."* — Grant, 14:31



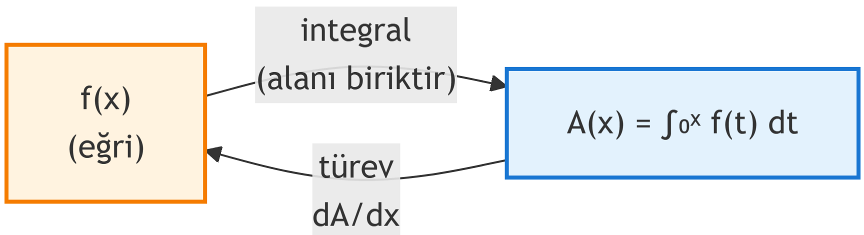
Şekil 1.7: Türev = teğet doğrunun eğimi. $f(x) = x^2$ üzerinde $x_0 = 2$'de türev $f'(2) = 4$, yani teğetin eğimi 4.

> **Builder Notu — Gradient ve Autodiff**
>
> İşte büyük olan. dA/dx, "x'i azıcık değiştirince çıktı ne kadar değişir" oranıdır — yani **gradient'in** tek değişkenli hâli. Eğitimde $dLoss/dw$ her ağırlığın loss'a duyarlılığını verir; gradient descent bu oranın ters yönünde adım atar. "Girdiyi dürt, çıktının değişimini ölç" sezgisi, **otomatik türevin (autodiff)** çekirdeğidir; backprop bu oranları zincir kuralıyla verimli hesaplar (Ders 4'te göreceğiz). Sonlu fark (finite difference) sezgisi de budur: dx'i çok küçük seçip $[f(x+dx) - f(x)]/dx$ ile türevi sayısal olarak yoklamak.

1.7 Temel Teorem: Türev ile İntegral Terstir

Elimizde şu var: gizem fonksiyonu $A(x)$ (x^2 altındaki alan) ve onun türevinin x^2 olduğu bilgisi. A'yı bulmak için **tersten çöz**: türevi x^2 olan fonksiyon hangisidir? Türev hesaplamada yeterince ustalaşınca, türevi verilen bir fonksiyonu geri mühendislikle bulabilirsin.

İntegrallerle türevler arasındaki bu gidiş-geliş — bir grafik altındaki alan fonksiyonunun türevinin, grafiği tanımlayan fonksiyonu geri vermesi — **Calculus'un Temel Teoremi** olarak bilinir. İki büyük fikri birbirine bağlar ve bir anlamda her birinin diğerinin **tersi** olduğunu gösterir.



Şekil 1.8: Calculus'un Temel Teoremi (FTC): türev ve integral, birbirinin tersidir.

> *"this back and forth between integrals and derivatives … is called the fundamental theorem of calculus … each one is an inverse of the other."* — Grant, 15:31

> Builder Notu — Forward/Backward Dualitesi
>
> Temel Teorem, **biriktirme ile oran** arasındaki derin bağdır. Pekiştirmeli öğrenmede (RL) value function V — zamana yayılmış indirimli kümülatif ödül, bir tür "integral" — ile anlık ödül (onun "türevi") Bellman denklemiyle birbirine bağlıdır; bu, FTC'nin ayrık akrabasıdır. Forward (biriktir) ve backward (türevini al / geri yay) dualitesi hem otomatik türevde hem de dinamik programlamada tekrar tekrar karşına çıkar.

1.8 Bu Dersin Özeti

1. Calculus, "ezberlenecek formüller" değil; doğru resmi çizersen kendin keşfedebileceğin bir avuç fikirdir.
2. Bir dairenin alanını eşmerkezli halkalara dilimleyerek bulduk: her halka $\approx 2\pi r$ genişliğinde, dr kalınlığında bir dikdörtgen, alanı $2\pi r \cdot dr$.
3. Halka alanlarını toplamak = $y = 2\pi r$ doğrusunun altındaki üçgenin alanı = $\frac{1}{2} \cdot 3 \cdot (2\pi \cdot 3) = \pi \cdot 3^2$. Bu, **integral** fikrinin somut hâli.
4. Asıl derin nokta: dr küçüldükçe bu yaklaşık toplam, grafik altındaki **kesin** alana yakınsar. Yaklaşıktan kesine geçiş, calculus'un kalbidir.
5. x^2 eğrisinin altındaki alan $A(x)$ bir gizem; ama $dA/dx = x^2$ olduğunu çıkarabiliyoruz. Bu oran **türev**.
6. Türev = bir fonksiyonun girdideki küçük değişime duyarlılığı ($dx \to 0$ iken dA/dx).
7. A'nın türevi orijinal grafiği geri verir → **Calculus'un Temel Teoremi**: türev ve integral birbirinin tersidir.

> ! Tek bir cümle
>
> Calculus iki büyük fikir üzerine kuruludur — **integral** (küçük niceliklerin toplamı = eğri altı alan) ve **türev** (anlık değişim oranı) — ve bu ikisi birbirinin tersidir; üçünü de tek bir daireyi dilimleyerek

kendin keşfedebilirdin.

1.9 Kontrol Soruları

Soru 1: Yarıçapı R = 5 olan bir daire için halka-dilimleme argümanını uygula. Alanı bul.

Cevap: Halka alanı $\approx 2\pi r \cdot dr$, $r \in [0, 5]$. Bu dikdörtgenleri yan yana dizince eğimi 2π olan $y = 2\pi r$ doğrusunun altındaki üçgeni elde ederiz: taban 5, yükseklik $2\pi \cdot 5 = 10\pi$. Alan $= \frac{1}{2} \cdot 5 \cdot 10\pi = 25\pi = \pi \cdot 5^2$. Genel R için aynı argüman πR^2 verir — yani formül, halkaları toplamanın doğrudan sonucudur.

Soru 2: dr küçüldükçe 'halka ≈ dikdörtgen' yaklaşımı neden daha iyi oluyor?

Cevap: Açılmış halkanın iç kenarı $2\pi r$, dış kenarı $2\pi(r + dr)$ uzunluğundadır; aradaki fark $2\pi \cdot dr$. $dr \to 0$ iken bu iki kenar birbirine yaklaşır, şekil gerçek bir dikdörtgene yakınsar ve "dikdörtgen" yaklaşımının hatası dr ile orantılı biçimde sıfıra gider. Daha ince dilimleme = daha küçük hata.

Soru 3: v(t) hızıyla giden bir araç. Neden kat edilen mesafe = hız grafiğinin altındaki alan?

Cevap: Küçük bir dt aralığında hız neredeyse sabittir, dolayısıyla o aralıkta kat edilen yol $\approx v(t) \cdot dt$. Tüm dt'lerdeki bu küçük yolları toplamak, hız grafiğinin altındaki ince dikdörtgenleri toplamaktır; $dt \to 0$ iken toplam, eğrinin altındaki alana yakınsar. Bu, daire problemiyle birebir aynı yapıdır: küçük niceliklerin toplamı = bir grafiğin altındaki alan = integral.

Soru 4: (Builder) dA/dx = x^2 ilişkisi gradient descent ile nasıl bağlanır?

Cevap: dA/dx, "x'i azıcık değiştirince A ne kadar değişir" oranıdır = türev, yani gradient'in tek değişkenli hâli. Eğitimde $dLoss/dw$ her ağırlığın loss'a duyarlılığını ölçer; gradient descent bu oranın ters yönünde küçük bir adım atarak loss'u azaltır. "Girdiyi dürt, çıktının değişimini ölç" sezgisi otomatik türevin (autodiff) çekirdeğidir; backprop bu oranları zincir kuralıyla katman katman verimli hesaplar (Ders 4).

1.10 Egzersizler

Egzersiz 1. Daire argümanını bir **kürenin hacmine** uyarla: küreyi, yarıçapı $r \in [0, R]$ olan ince küresel kabuklara böl. Her kabuğun yüzey alanı $4\pi r^2$, kalınlığı dr, dolayısıyla hacmi $\approx 4\pi r^2 \cdot dr$. Bu küçük hacimleri toplamak hangi grafiğin (hangi fonksiyonun) altındaki alana karşılık gelir? (Sonucun $\frac{4}{3}\pi R^3$ çıkması gerekir — neden?)

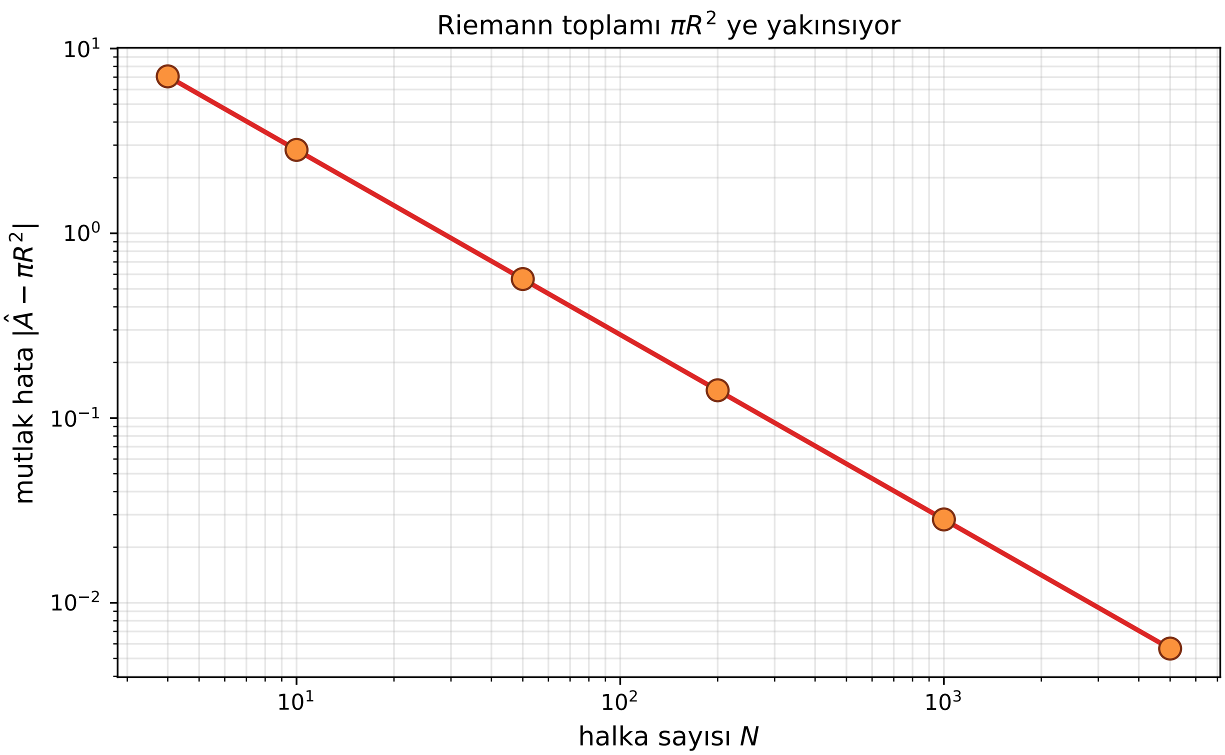
Egzersiz 2. Bir $A(x)$ fonksiyonunun türevinin x^3 olduğunu bildiğini varsay (yani $dA/dx = x^3$). 3 ile 3,01 noktalarında $[A(3{,}01) - A(3)]/0{,}01$ oranının yaklaşık hangi sayıya eşit olmasını beklersin? Cevabını $dx \to 0$ limiti açısından gerekçelendir.

Egzersiz 3. *(Yapısal)* Bir üçgenin alanının neden $\frac{1}{2} \cdot \text{taban} \cdot \text{yükseklik}$ olduğunu, daire argümanındaki $y = 2\pi r$ doğrusunun altında kalan bölgeyi düşünerek açıkla. (İpucu: bu doğru orijinden geçiyor; tabanı 3, yüksekliği $2\pi \cdot 3$ olan dik üçgen, kenarları 3 ve $2\pi \cdot 3$ olan dikdörtgenin tam yarısıdır.)

Egzersiz 4. *(Python — görsel doğrulama)* Daireyi N halkaya bölüp alanı Riemann toplamıyla yaklaştır ve πR^2 ile karşılaştır. N büyüdükçe mutlak hatanın küçüldüğünü hem sayısal hem grafiksel göster.

```
N=    4  yaklasik=21.20575  hata=7.068583
N=   10  yaklasik=25.44690  hata=2.827433
N=   50  yaklasik=27.70885  hata=0.565487
N=  200  yaklasik=28.13296  hata=0.141372
N= 1000  yaklasik=28.24606  hata=0.028274
N= 5000  yaklasik=28.26868  hata=0.005655

Gercek pi*R^2 = 28.274334
```



Şekil 1.9: Riemann toplamı πR^2'ye yakınsıyor — N büyüdükçe mutlak hata $\sim 1/N$ hızıyla küçülür (log-log eğri ≈ doğrusal eğim −1).

Egzersiz 5. *(Sonraki dersin habercisi)* Bu kez alanı değil, **fonksiyonun kendisini** ele al: $f(x) = x^2$. $[f(x+dx) - f(x)]/dx$ oranını dx cinsinden aç ve sonucu *"$2x+$ (bir terim)"* biçiminde yaz. $dx \to 0$ iken hangi terim kaybolur? Bulduğun sonuç, Ders 2'de "türevin paradoksu" olarak göreceğimiz $d(x^2)/dx = 2x$ ifadesidir.

1.11 Sonraki Ders İçin Hazırlık

Ders 2: Türevin Paradoksu

Ders 2'de türevi yakından inceliyoruz. "Anlık değişim oranı" aslında ince bir paradoks barındırır: tek bir **anda** hiçbir şey değişmez (değişim için iki an gerekir), ama yine de anlamlı bir "anlık hız" tanımlarız. Grant bu paradoksu $dx \to 0$ limitiyle çözer ve $d(x^2)/dx = 2x$ sonucunu, bir kareyi büyüterek tamamen **geometrik** olarak türetir.

Ana konular:

- Türevin paradoksu: "anlık değişim oranı" tam olarak ne demek?
- $d(x^2)/dx = 2x$'in geometrik türetimi (ve neden ihmal edilen dx^2 terimi sıfıra gider).
- Türev = bir noktadaki teğet doğrunun eğimi.

> ⚠ Ders 2 öncesi yapılacak
>
> - Egzersizleri çöz — özellikle **4** (Riemann simülasyonu) ve **5** ($d(x^2)/dx$ önizlemesi).
> - "Yaklaşıktan kesine" geçişin ($dr \to 0$) neden calculus'un kalbi olduğunu kendi cümlenle yaz.
> - Ana cümleyi tekrar oku: *"Calculus iki büyük fikir üzerine kuruludur — integral ve türev — ve bu ikisi birbirinin tersidir."*

1.12 Anahtar Kavramlar (Cheat Sheet)

Kavram	Tanım	Grant'ta
İntegral	Bir eğrinin altındaki alan; küçük niceliklerin toplamının limiti	8m14
Türev	dA/dx; bir fonksiyonun girdideki küçük değişime duyarlılığı	14m16
Halka alanı ($2\pi r \cdot dr$)	Açılmış halkanın yaklaşık dikdörtgen alanı (çevre × kalınlık)	3m20
πR^2 **türetimi**	$y = 2\pi r$ doğrusu altındaki üçgen: $\frac{1}{2} \cdot R \cdot 2\pi R$	6m28
Yaklaşıktan kesine	$dr \to 0$ iken toplam, grafik altındaki tam alana yakınsar	7m08
$A(x)$**: integral fonksiyonu**	x^2 altındaki, 0'dan x'e kadar olan alan	10m37
dA/dx **= grafiğin yüksekliği**	Alan fonksiyonunun türevi = eğrinin o noktadaki değeri	13m50
Temel Teorem (FTC)	Türev ve integral birbirinin ters işlemidir	15m31

Kavram	Tanım	Grant'ta
"Kendin keşfet"	Grant'ın pedagojik hedefi: formülü değil, kaynağını anla	0m54
Simetriye saygı	Doğru parçalama (eşmerkezli halkalar) problemi çözer	2m24

1.13 ML Bağlantıları Özeti

7 köprü

1. **Türev (dA/dx)** → gradient: loss'un parametreye duyarlılığı; gradient descent bu yönün tersine adım atar.
2. **Oran/dürtme sezgisi** → otomatik türev (autodiff) ve backprop: "girdiyi dürt, çıktının değişimini ölç"ün mekanikleştirilmesi (Ders 4'te zincir kuralı).
3. **İntegral (küçük niceliklerin toplamı)** → beklenen değer $E[f(X)] = \int f \cdot p\,dx$ ve Monte Carlo: çok örneğin ortalaması, $N \to \infty$ iken yakınsar ($dr \to 0$'ın aynası).
4. **Yaklaşıktan kesine / $dr \to 0$** → limit: sürekli optimizasyon, gradient flow, neural ODE (durum $= \int$ dinamik dt).
5. **Temel Teorem (türev ↔ integral terslik)** → RL'de value function (kümülatif ödül = "integral") ile anlık ödül ("türev") arasındaki Bellman bağı; forward/backward dualitesi.
6. **Simetriye saygı (doğru parçalama)** → invariance/equivariance: CNN'de öteleme değişmezliği; göreve uygun ayrıştırma öğrenmeyi kolaylaştırır.
7. **"Kendin keşfedebilirdin"** → builder zihniyeti: her ML tekniğinin nereden geldiğini (türetimini) anlamak, kara kutu ezberden üstündür.

! Tek bir şey alıp gideceksen

Calculus'un tamamı iki fikre dayanır — **integral** (küçük parçaları topla = eğri altı alan) ve **türev** (anlık değişim oranı) — ve Temel Teorem bunların birbirinin tersi olduğunu söyler. Hiçbirini ezberlemene gerek yok; doğru resmi çizip dr ile dx'i küçülttüğünde hepsi kendiliğinden ortaya çıkar.

2 Türevin Paradoksu

Anlık değişim oranı, dt → 0 ve teğet doğrunun eğimi

ℹ Bölüm bilgisi

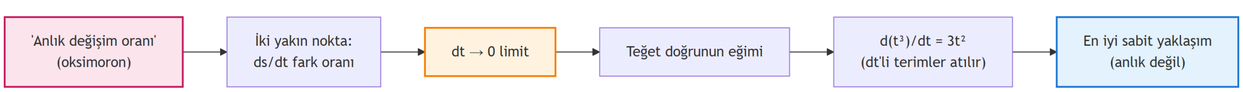
- **Grant'ın videosu:** YouTube — Chapter 2: The paradox of the derivative (≈17 dk)
- **Kaynak:** 3Blue1Brown — Essence of Calculus
- **Okuma süresi:** ≈22 dk

2.1 Bu Derste Ne Var?

Ders 1'de calculus'un üç fikrine — integral, türev, terslik — yukarıdan baktık. Bu derste **türevi** yakından tanımlıyoruz. Ama bir tuzak var: türev genelde *"anlık değişim oranı"* diye tanımlanır — oysa Grant'a göre bu ifade bir **oksimorondur**. Değişim iki nokta arasında olur; tek bir ana kilitlenince değişime yer kalmaz. Calculus'un kurucularının zekası, bu paradoksu dt'yi **sıfıra yaklaştırarak** (ama sıfır yapmadan) zarifçe atlatmakta.

Üç ana fikir:

1. **Türev =** ds/dt **oranının,** $dt \to 0$ **iken yaklaştığı değer.** Belirli bir dt için değil, limitinde.
2. **Geometrik anlam:** bir noktadaki **teğet doğrunun eğimi**.
3. $d(t^3)/dt = 3t^2$ — ve "dt'li terimleri yok say" hilesi, calculus'u kullanışlı yapan şeydir.



Şekil 2.1: Paradoksun çözümünün yol haritası: oksimorondan limite, limitten teğet eğimine.

> *"it's common for people to say that the derivative measures an instantaneous rate of change, but … that phrase is actually an oxymoron."* — Grant, 0:35

Builder Notu — ML Köprüleri

- **Türev = fark oranının limiti** → sayısal türev ve **gradient checking**: backprop gradyanını, $[f(w+\varepsilon) - f(w)]/\varepsilon$ sonlu farkıyla doğrularsın.
- **Teğet = en iyi yerel doğrusal yaklaşım** → birinci-derece Taylor; gradient, bir noktada fonksiyonun **en iyi lineer modeli**dir.

- **"dt^2 terimini yok say" hilesi** → forward-mode autodiff'in **dual sayıları** ($a+b\varepsilon$, $\varepsilon^2 = 0$): yüksek dereceli terimleri tam olarak düşürür. Türevin cebirsel mekaniği birebir budur.
- **"Anlık değil, en iyi sabit yaklaşım"** → pratikte hep sonlu dt (Euler adımı, learning rate); sürekli türev bunun limiti.
- **Kuvvet kuralı** $d(t^n)/dt = n \cdot t^{n-1}$ → otomatik türev kural tablosunun ilk girdisi.

2.2 Hedef ve Paradoks: "Anlık Değişim Oranı" Bir Oksimoron

Amaç basit: türevin ne olduğunu açıklamak. Ama konu incelikli ve dikkatsiz olursan paradokslarla dolu. Bu yüzden ikincil hedef, bu paradoksları görmek ve onlardan nasıl kaçınacağını anlamak.

Sık duyarsın: *"türev, anlık değişim oranını ölçer."* Oysa düşününce bu ifade kendi içinde çelişkili. **Değişim**, ayrı zaman noktaları arasında olan bir şeydir; kendini tek bir ana körleştirdiğinde, değişime hiç yer kalmaz. Yine de "anlık hız" demek istediğimiz gerçek bir sezgi var — ve calculus'un kurucuları bu sezgiyi, mantıklı bir matematik parçasıyla (türevle) yakaladılar.

> *"if that feels strange and paradoxical, good! You're grappling with the same conflicts that the fathers of calculus did."* — Grant, 4:14

Builder Notu — Ayrıklaştırma

Bu paradoks ML'de **ayrıklaştırma (discretization)** olarak karşına çıkar. Bir optimizasyon adımı ya da bir ODE çözücü, "anlık" türevi asla doğrudan kullanmaz; hep sonlu bir adım (dt, learning rate) alır. Sürekli türev, bu sonlu adımların $dt \to 0$ limitidir. "Anlık oran" matematiksel bir idealizasyon; pratikte daima yaklaşık, sonlu bir fark hesaplarsın.

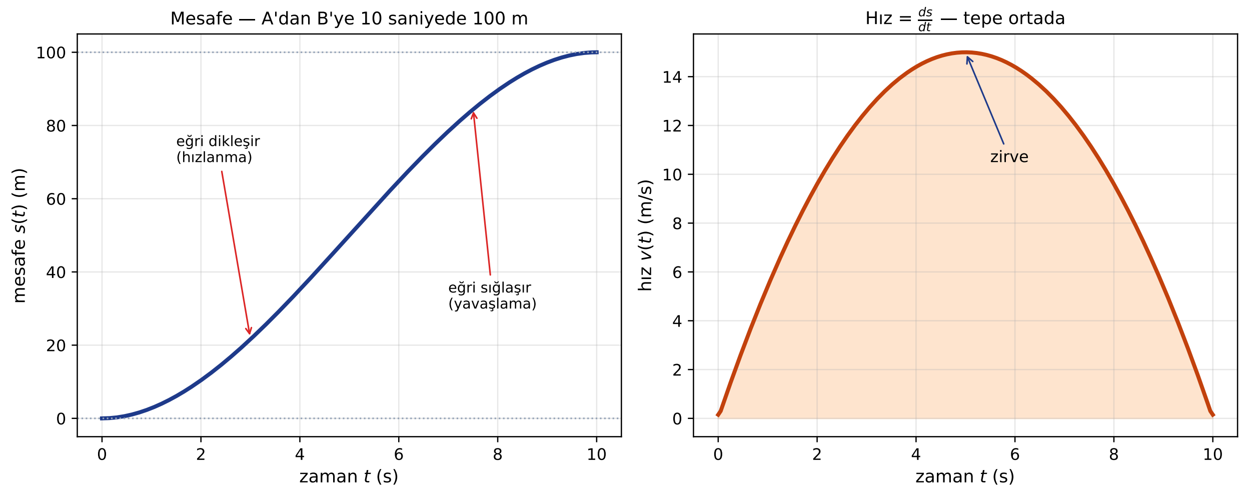
2.3 Araba Örneği: Mesafe ve Hız

Merkezî örneğimiz: bir araba A noktasından başlıyor, hızlanıyor, sonra yavaşlayıp 100 metre ötedeki B noktasında duruyor — tüm bunlar 10 saniyede oluyor.

Bu hareketi grafikleyebiliriz: dikey eksen kat edilen **mesafe**, yatay eksen **zaman**. Her t anında grafiğin yüksekliği, arabayı o ana kadar ne kadar yol gittiğini söyler. Bu mesafe fonksiyonuna $s(t)$ diyelim (d harfi calculus'ta zaten başka bir işte çalışıyor).

Başlangıçta eğri **sığ**: araba yavaş, ilk saniyede az yol alır. Orta bölümde hızlanır, her saniye daha çok yol — grafiğin eğimi **diklesir**. Sona doğru yavaşlayınca eğri yine sığlaşır.

Aynı hareketin **hızını** (m/s) zamana karşı çizersek, bir tepe (bump) elde ederiz: başta küçük, ortada maksimum, sonda yine sıfıra iner. Bu iki eğri birbiriyle ilişkili — mesafe fonksiyonunu değiştirirsen hız fonksiyonu da değişir. Anlamak istediğimiz tam olarak bu ilişki: hız, mesafe-zaman fonksiyonuna nasıl bağlı?



Şekil 2.2: Aynı hareketin iki yüzü: mesafe $s(t)$ ve hız $v(t) = ds/dt$. Mesafe grafiği dikleşince hız zirveye çıkar; sığlaşınca sıfıra döner.

Builder Notu — Kümülatif ↔ Anlık

$s(t)$ (biriken nicelik) ile hız (onun değişim oranı) ilişkisi, Ders 1'deki integral↔türev çiftinin somut hâli. ML'de bunun eşdeğeri: **kümülatif** bir nicelik (toplam ödül, biriken kayıp) ile onun **anlık** değişimi (adım ödülü, gradyan) arasındaki bağ. Birini bilince diğerini türev/integral ile geçersin.

2.4 Tek Anda Hız Neden Anlamsız?

"Hız" kelimesinin burada tam ne demek olduğunu eleştirel düşünelim. Sezgisel olarak bir andaki hız, arabanın o anda hız göstergesinde (speedometer) gösterdiği şeydir; ve mesafe fonksiyonu dik olduğunda hızın yüksek olması mantıklı görünür.

Ama tuhaf olan şu: **tek bir anda hız anlamsızdır.** Sana bir arabanın tek bir anlık fotoğrafını gösterip "ne kadar hızlı gidiyor?" diye sorsam, söyleyemezsin. İhtiyacın olan şey, karşılaştıracak **iki ayrı zaman noktası**: mesafedeki değişimi zamandaki değişime bölersin. Hız zaten budur — birim zamanda kat edilen mesafe.

İşte paradoks: tek tek zaman noktalarına bir hız atamak istiyoruz, ama hızı hesaplamak iki ayrı zaman noktasını karşılaştırmayı gerektiriyor.

Gerçek dünyada araba bunu nasıl atlatır? 3. saniyede hız göstergesi, arabanın çok küçük bir zamanda gittiği yolu ölçer — diyelim 3 ile 3,01 saniye arası — ve bu küçük mesafeyi küçük zamana (0,01 s) böler. Yani fiziksel araba paradoksu **yan geçer**: tek bir anda değil, çok küçük bir zaman aralığında hız hesaplar.

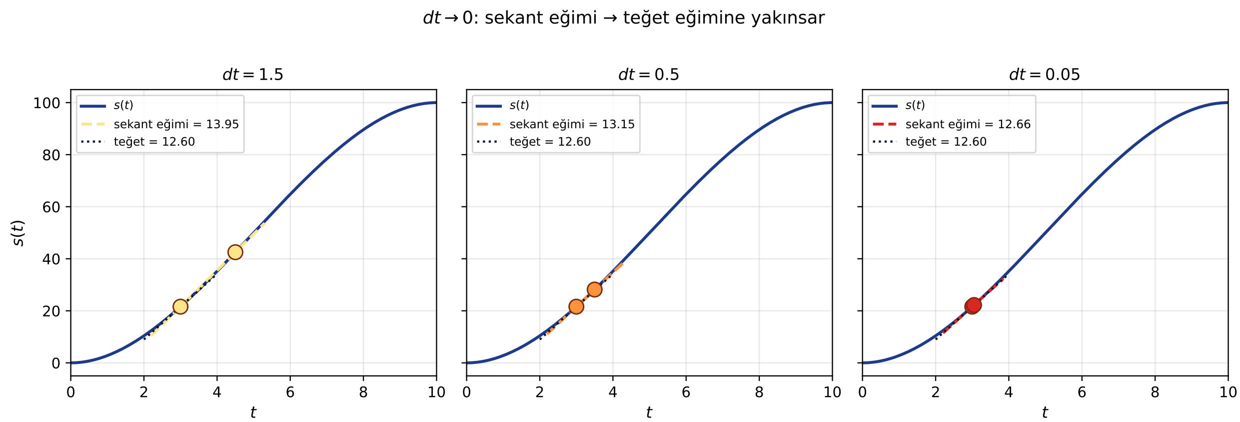
> *"a physical car just side-steps the paradox … it computes speed during a very small amount of time."* — Grant, 5:08

Builder Notu — Finite Difference

Bu "iki yakın nokta al, oranı hesapla" fikri, **sayısal türevin** (finite difference) ta kendisidir. Bir gradyanı elle doğrulamak istediğinde (gradient checking) tam bunu yaparsın: $[L(w+\varepsilon) - L(w)]/\varepsilon$. Otomatik türev (autodiff) ise bu oranı sonlu ε ile değil, cebirsel limitle hesaplar — sonraki bölümde göreceğimiz gibi bu hem daha hızlı hem daha doğrudur.

2.5 ds/dt: İki Yakın Nokta Arasındaki Eğim

Bu küçük zaman farkına dt diyelim (0,01 gibi düşün), ve onun yol açtığı küçük mesafe farkına ds. O hâlde bir andaki hız, yaklaşık olarak ds/dt — küçük mesafe değişiminin küçük zaman değişimine oranı.



Şekil 2.3: Sekant doğrusu (ds/dt fark oranı) $dt \to 0$ iken teğet doğrusuna yakınsar. Mesafe grafiğinin $t = 3$ noktasındaki türevi, teğetin eğimidir.

Grafiksel olarak: $t = 3$ civarında mesafe-zaman grafiğine yakınlaş. dt sağa doğru küçük bir adım (zaman yatay eksende), ds ise grafiğin yüksekliğindeki buna karşılık gelen değişim. Yani ds/dt, grafikteki birbirine çok yakın iki nokta arasındaki **yükselme/yatay (rise/run) eğimidir**.

$$\frac{ds}{dt} = \frac{s(t+dt) - s(t)}{dt}$$

$t = 3$'te özel bir şey yok; bunu her t için yapabiliriz, böylece ds/dt'yi t'nin bir fonksiyonu — hız fonksiyonu — olarak görürüz.

İşte kritik nokta: saf matematikteki türev, belirli bir dt için bu ds/dt oranı **değildir**. Türev, dt giderek küçülürken (0'a yaklaşırken) bu oranın **yaklaştığı** değerdir.

$$\frac{ds}{dt} = \lim_{dt \to 0} \frac{s(t+dt) - s(t)}{dt}$$

Bunun çok güzel bir görsel anlamı var: $dt \to 0$ iken iki nokta birbirine yaklaşır ve onlardan geçen doğrunun eğimi, grafiğe o tek noktada **teğet olan doğrunun eğimine** yakınsar.

> *"the true honest-to-goodness pure math derivative … is equal to the slope of a line tangent to the graph at a single point."* — Grant, 8:33

İnce ama hayati nokta: türev, dt sonsuz küçük olduğunda olan şey **değildir** (öyle bir şey yok); dt'ye 0 koymak da **değildir** (sıfıra bölme olurdu). dt daima sonlu, sıfır-olmayan bir değerdir — yalnızca 0'a yaklaşır.

> *"this dt is always a finitely small non-zero value, it's just that it approaches 0."* — Grant, 9:03

Bu yüzden bu eğimi "anlık değişim oranı" değil, **"bir nokta civarında değişim oranının en iyi sabit yaklaşımı"** olarak düşünmek en sağlıklısıdır.

> Builder Notu — Yerel Lineer Model
>
> Teğet doğru = fonksiyonun o noktadaki **en iyi doğrusal yaklaşımı** (birinci dereceden Taylor). ML'de gradient tam bu rolü oynar: yüksek boyutlu kayıp yüzeyini, bulunduğun noktada bir **hiper-düzlemle** (lineer model) yaklaştırır. Gradient descent her adımda bu yerel doğrusal modele güvenir; adım çok büyük olursa yaklaşım bozulur — bu yüzden learning rate küçük tutulur.

2.6 Worked Example: $d(t^3)/dt = 3t^2$

Garip ama gerçek: "dt küçülürken oran neye yaklaşır?" diye sormak hesabı zorlaştırmaz, **kolaylaştırır**. Görelim.

Mesafe fonksiyonun tam olarak t^3 olsun (1 saniyede $1^3 = 1$ m, 2 saniyede $2^3 = 8$ m, …). Hızı, yani ds/dt'yi, belirli bir anda — diyelim $t = 2$'de — hesaplamak isteyelim. Şimdilik dt'yi somut bir büyüklük olarak tutalım; birazdan 0'a göndereceğiz. 2 ile $2 + dt$ arasındaki küçük mesafe değişimi $s(2 + dt) - s(2)$, bölü dt:

$$\frac{s(2+dt) - s(2)}{dt} = \frac{(2+dt)^3 - 2^3}{dt}$$

Üstteki ifadeyi cebirsel olarak açalım:

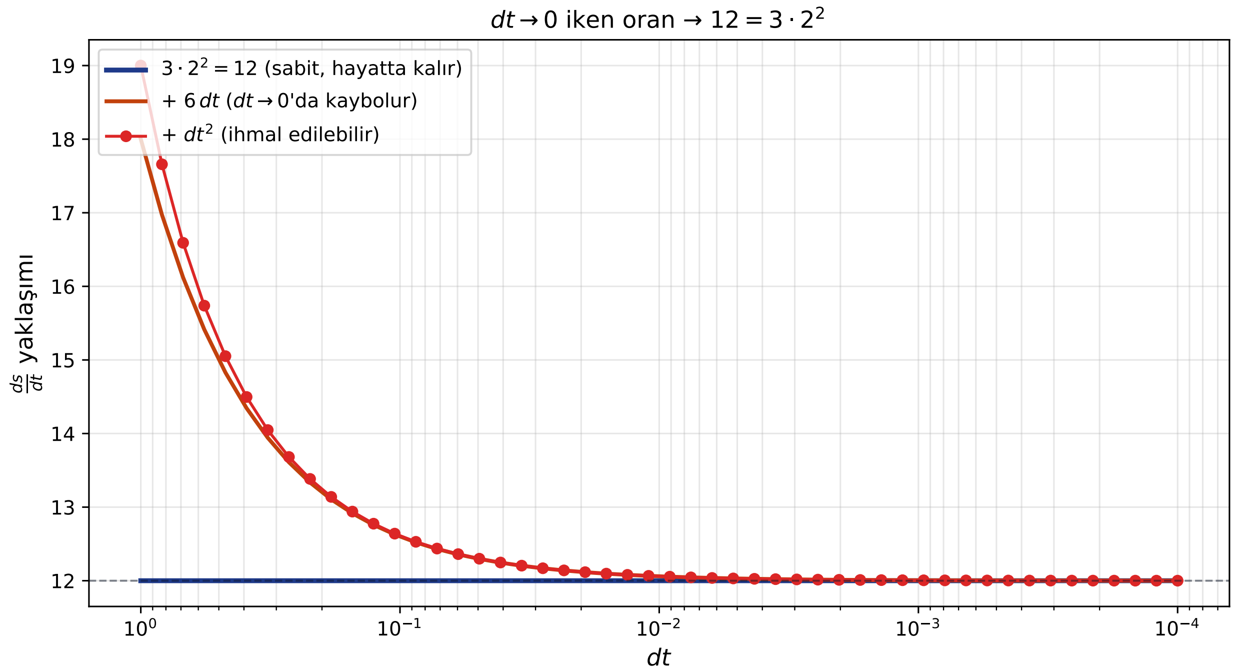
$$(2 + dt)^3 = 2^3 + 3 \cdot 2^2\, dt + 3 \cdot 2\, dt^2 + dt^3$$

Karmaşık görünüyor ama sadeleşiyor. 2^3 terimleri (açılımdaki ile çıkarılan) **birbirini götürür**. Geriye kalan her terimde bir dt var; paydadaki dt ile sadeleşince:

$$\frac{ds}{dt} = 3 \cdot 2^2 + 3 \cdot 2\, dt + dt^2 \xrightarrow{dt \to 0} 3 \cdot 2^2 = 12$$

$dt \to 0$ sorduğumuzda, içinde dt olan terimleri **tamamen yok sayabiliriz**. Geriye tertemiz $3 \cdot 2^2 = 12$ kalır: bu, $t = 2$ noktasındaki teğet doğrunun eğimidir. $t = 2$'de özel bir şey yok; genel olarak:

$$\frac{d}{dt}\, t^3 = 3t^2$$



Şekil 2.4: $(2+dt)^3$ açılımının terimleri: ana terim $3 \cdot 2^2 = 12$ (mavi), dt ile orantılı düzeltme (turuncu), dt^2 ile orantılı çok küçük artık (kırmızı). $dt \to 0$ iken yalnızca 12 hayatta kalır.

> *"the derivative of t cubed as a function of t is 3 times t squared."* — Grant, 13:10

İşte calculus'un neden bu kadar işe yaradığının kalbi: belirli bir dt için ifade bir karmaşaydı; ama oranın $dt \to 0$ iken **yaklaştığı** değere bakınca, o karmaşanın çoğunu yok sayabiliyoruz.

> *"that right there is kind of the heart of why calculus becomes useful."* — Grant, 14:13

Builder Notu — Dual Sayılar

$dt \to 0$ iken dt^2 ve dt^3 terimlerini atmak tesadüf değil — **forward-mode autodiff** bunu birebir mekanikleştirir. **Dual sayılar** $a + b\varepsilon$ (burada $\varepsilon^2 = 0$ tanımlıdır) ile çalışırsın: $(2+\varepsilon)^3 = 8 + 12\varepsilon + 6\varepsilon^2 + \varepsilon^3 = 8 + 12\varepsilon$, çünkü ε^2 ve sonrası sıfır. ε'un katsayısı (12) tam olarak türevdir. Yani Grant'ın "dt'li yüksek terimleri yok say" hilesi, modern otomatik türevin cebirsel temelidir; PyTorch/JAX bunu sembolik açılım yapmadan, sayısal fark hatası olmadan yapar.

2.7 Paradoksun Çözümü: En İyi Sabit Yaklaşım

Somut bir türev ($3t^2$) elimizde olunca, "anlık hız" yanılsamasına fazla inanırsak çıkan paradoksu görebiliriz. t^3 mesafe fonksiyonuyla giden arabayı düşün ve başlangıç anına, $t = 0$'a bak. Araba o anda hareket ediyor mu?

Bir yandan: hızı türevle hesaplarız, $3t^2$, ve $t = 0$ için bu 0 çıkar. Teğet doğru tam yatay; yani "anlık hız" 0, demek ki hareket etmiyor.

$$3t^2\Big|_{t=0} = 3 \cdot 0^2 = 0$$

Öte yandan: eğer $t = 0$'da hareket etmiyorsa, ne zaman hareket etmeye başlıyor? Bir an dur ve düşün.

> *"do you see the paradox? The issue is that the question makes no sense."* — Grant, 15:24

Çözüm: soru anlamsız, çünkü "bir anda değişim" fikrine atıfta bulunuyor ama öyle bir şey yok — türevin ölçtüğü o değil. Bir mesafe fonksiyonunun türevinin 0 olması demek, arabanın o nokta civarındaki hızının **en iyi sabit yaklaşımının** 0 m/s olması demektir. Gerçek bir zaman aralığına bakarsan (0 ile 0,1 s arası) araba **hareket eder**: $0{,}001$ m gider, yani ortalama hız $0{,}01$ m/s. Daha küçük dürtüşler için bu oran 0'a yaklaşır — ama bu, arabanın durağan olduğu anlamına gelmez; 0 sabit hızıyla yaklaştırmak sadece bir yaklaşımdır.

Bu yüzden birisi türeve "anlık değişim oranı" dediğinde — özünde çelişkili bu ifade — onu **"değişim oranının en iyi sabit yaklaşımı"** için kavramsal bir kısaltma olarak düşün.

> *"think of that as a conceptual shorthand for the best constant approximation for rate of change."*
> — Grant, 16:33

Builder Notu — Kritik Noktalar

"Türev = en iyi sabit/lineer yaklaşım" bakışı, optimizasyonun temelidir. Gradient descent, kayıp yüzeyini her noktada bir teğet düzlemle (birinci derece) yaklaştırıp o yönde adım atar. Türevin 0 olması (kritik nokta) ise "yerel olarak düz" demektir — minimum, maksimum veya eyer noktası. $t = 0$'daki düz teğet, bir loss minimumundaki sıfır gradyanın birebir analoğudur: gradyan 0 olması, modelin "durduğu" anlamına gelmez, yalnızca o noktadaki en iyi lineer yaklaşımın sabit olduğunu söyler.

2.8 Bu Dersin Özeti

1. Türev sık sık "anlık değişim oranı" diye anılır, ama bu bir oksimorondur: değişim iki ayrı nokta gerektirir, tek bir anda değişime yer yoktur.
2. Araba örneği: mesafe $s(t)$, hız = ds/dt. Mesafe grafiği dikleştiğinde hız yüksektir.
3. Tek anda hız anlamsızdır; gerçek araba paradoksu yan geçer — t ile $t + dt$ arasındaki mesafeyi ölçüp dt'ye böler.
4. ds/dt, iki yakın nokta arasındaki eğimdir; türev ise $dt \to 0$ iken bu oranın yaklaştığı değer = o noktadaki **teğet doğrunun eğimi**.
5. dt ne sonsuz küçüktür ne de 0'dır; yalnızca 0'a yaklaşır. (0 koymak sıfıra bölme olurdu.)
6. $d(t^3)/dt = 3t^2$: $(2 + dt)^3$ açılımında dt'li terimler $dt \to 0$'da kaybolur. Bu sadeleşme, calculus'un neden işe yaradığının kalbidir.
7. Türev, "en iyi sabit yaklaşım"dır; $t = 0$'da $3 \cdot 0^2 = 0$ olması, arabanın gerçekten durduğu anlamına gelmez.

! Tek bir cümle

Türev, ds/dt fark oranının $dt \to 0$ iken yaklaştığı değerdir — geometrik olarak teğet doğrunun eğimi, kavramsal olarak "anlık" değil, bir nokta civarında değişim oranının en iyi sabit yaklaşımı.

2.9 Kontrol Soruları

i Soru 1: s(t) = t^2 için t = 3'te türevi, açılım yöntemiyle (dt → 0) bul.

Cevap: $[s(3+dt)-s(3)]/dt = [(3+dt)^2-3^2]/dt = [9+6\,dt+dt^2-9]/dt = (6\,dt+dt^2)/dt = 6+dt$. $dt \to 0$ iken bu 6'ya gider. Yani $d(t^2)/dt = 2t$ ve $t = 3$'te değer $2 \cdot 3 = 6$. (Genel kuvvet kuralı: $d(t^n)/dt = n \cdot t^{n-1}$.)

i Soru 2: 'Türev, dt'ye 0 koymaktır' demek neden yanlış?

Cevap: dt'ye 0 koyarsan pay da payda da 0 olur: $0/0$ belirsizdir, tanımsız. Türev, oranın $dt = 0$'daki **değeri** değil, $dt \to 0$ iken **yaklaştığı limittir**. dt daima sonlu ve sıfır-olmayan kalır; biz yalnızca giderek küçülen dürtüşlerde oranın hangi sayıya yaklaştığına bakarız. Bu ayrım, paradokstan kaçmanın anahtarıdır.

i Soru 3: Bir fonksiyonun bir noktadaki türevi 0 ise, fonksiyon o noktada 'durağan/sabit' midir?

Cevap: Hayır. Türevin 0 olması, yalnızca o nokta civarındaki en iyi sabit (lineer) yaklaşımın eğiminin 0 olduğunu söyler. Fonksiyon hâlâ değişiyor olabilir — t^3 fonksiyonu $t = 0$'da türevi 0'dır ama araba durağan değildir; 0,1 s'de 0,001 m gider. Türevin 0 olduğu nokta bir **kritik noktadır**: yerel minimum, maksimum ya da eyer noktası olabilir.

i Soru 4: (Builder) Forward-mode autodiff, dual sayılarla d(t^3)/dt'yi t = 2'de nasıl verir?

Cevap: Dual sayı $2+\varepsilon$ al ($\varepsilon^2 = 0$ tanımlı). $(2+\varepsilon)^3 = 8+12\varepsilon+6\varepsilon^2+\varepsilon^3 = 8+12\varepsilon$ (çünkü ε^2 ve sonrası sıfır). ε'un katsayısı 12, tam olarak türevdir ($3 \cdot 2^2 = 12$). Bu, Grant'ın "dt^2 ve sonrası terimleri at" hilesinin birebir cebirsel mekaniğidir; PyTorch/JAX türevleri ne sembolik açılımla ne de sayısal fark hatasıyla, bu dual-sayı yapısıyla (ileri mod) hesaplar.

2.10 Egzersizler

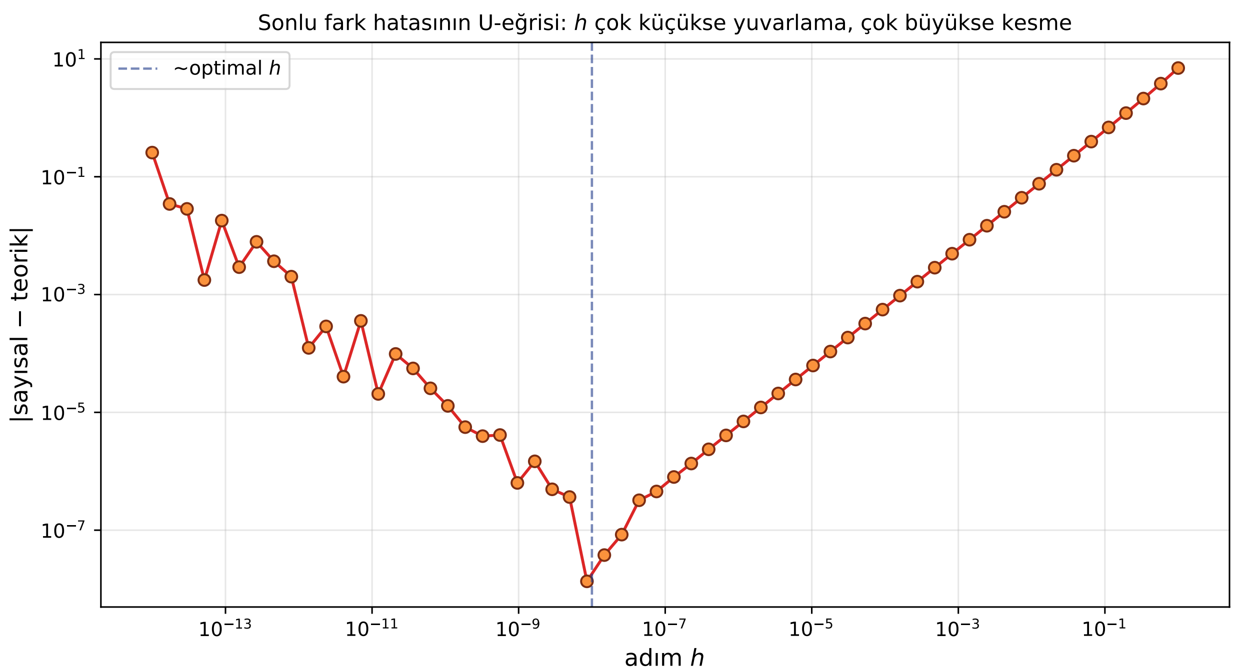
Egzersiz 1. Türevin limit tanımını kullanarak $d(t^2)/dt$'yi **genel** t için türet: $[(t+dt)^2 - t^2]/dt$ ifadesini sadeleştir ve $dt \to 0$ limitini al. Sonucun $2t$ çıktığını göster.

Egzersiz 2. $(t+dt)^4$ açılımını yaz (binom). $d(t^4)/dt$'yi, dt'li terimleri $dt \to 0$'da atarak hesapla ve kuvvet kuralının ($4t^3$) doğrulandığını göster.

Egzersiz 3. *(Türev her zaman var mı?)* $f(t) = |t|$ fonksiyonunu $t = 0$'da incele. $[f(0+dt) - f(0)]/dt$ oranını $dt > 0$ ve $dt < 0$ için ayrı ayrı hesapla. Sağdan ve soldan farklı değerler ($+1$ ve -1) çıktığı için limit yoktur → f, $t = 0$'da **türevlenemez** (bir köşe). Sonuç: her fonksiyon her noktada türevlenebilir değildir.

Egzersiz 4. *(Python — görsel doğrulama)* $s(t) = t^3$ için sayısal türevi $[s(t+h) - s(t)]/h$ ile hesapla; h küçüldükçe $t = 2$'de $3 \cdot 2^2 = 12$'ye yakınsadığını göster. h çok küçülünce kayan-nokta yuvarlama hatasının nasıl büyüdüğüne dikkat et (autodiff'in sonlu farktan neden üstün olduğunun kanıtı).

```
teorik turev 3*t^2 = 12.0
h=   1e+00  yaklasik=19.00000000  hata=7.000e+00
h=   5e-01  yaklasik=15.25000000  hata=3.250e+00
h=   1e-01  yaklasik=12.61000000  hata=6.100e-01
h=   1e-02  yaklasik=12.06010000  hata=6.010e-02
h=   1e-04  yaklasik=12.00060001  hata=6.000e-04
h=   1e-06  yaklasik=12.00000600  hata=6.002e-06
h=   1e-09  yaklasik=12.00000099  hata=9.929e-07
h=   1e-12  yaklasik=12.00106681  hata=1.067e-03
```



Şekil 2.5: Sonlu fark hatasının ünlü U-eğrisi: h büyükken kesme hatası (truncation), h çok küçükken yuvarlama hatası (floating-point) baskın. Tatlı nokta ~10^{-5} civarında.

Egzersiz 5. *(Sonraki dersin habercisi)* $d(t^2)/dt = 2t$ ve $d(t^3)/dt = 3t^2$ sonuçlarını cebir yapmadan düşün: kenarı t olan bir karenin alanı t^2, kenarı dt büyürse alan ne kadar artar? Kenarı t olan bir küpün hacmi t^3, kenar dt büyürse hacim ne kadar artar? Ders 3 bu türevleri tam olarak böyle — kareyi ve küpü büyüterek, **geometrik** olarak — türetecek.

2.11 Sonraki Ders İçin Hazırlık

Ders 3: Geometriyle Türev Formülleri

Ders 2'de bir türevi ($t^3 \to 3t^2$) cebirsel olarak, gözle görülür bir karmaşadan geçerek türettik. Ders 3'te Grant aynı formülleri **görsel** olarak türetiyor: x^2 bir karenin alanı, x^3 bir küpün hacmi olarak düşünülünce, türev *"kenarı azıcık büyütünce alan/hacim ne kadar artar?"* sorusuna dönüşür ve formüller diyagramdan kendiliğinden çıkar.

Ana konular:

- $d(x^2)/dx = 2x$'in kare-büyütme diyagramıyla türetimi.
- $d(x^3)/dx = 3x^2$ ve genel kuvvet kuralı, geometrik olarak.
- $1/x$ ve $\sin(x)$ gibi fonksiyonların türevleri için görsel sezgiler.

2.12 Anahtar Kavramlar (Cheat Sheet)

Kavram	Tanım	Grant'ta
Türevin paradoksu	"Anlık değişim oranı" oksimorondur; değişim iki nokta ister	0m35
Mesafe fonksiyonu $s(t)$	Zamana karşı kat edilen yol; grafiğin eğimi = hız	1m46
ds/dt **(fark oranı)**	İki yakın nokta arasındaki rise/run eğimi	5m51
Türev = limit	$dt \to 0$ iken ds/dt oranının yaklaştığı değer	8m02
Teğet doğru eğimi	Türevin geometrik anlamı (tek noktada)	8m33
dt **sonlu, sıfır değil**	Türev, dt'ye 0 koymak değil; 0'a yaklaşmaktır	9m03
$d(t^3)/dt = 3t^2$	$(2+dt)^3$ aç, dt'li terimleri at → $3 \cdot 2^2$	13m10
Kuvvet kuralı	$d(t^n)/dt = n \cdot t^{n-1}$	13m04
En iyi sabit yaklaşım	"Anlık oran" yerine türevin doğru kavramı	16m33

2.13 ML Bağlantıları Özeti

7 köprü

1. **Türev = fark oranının limiti** → sayısal türev ve gradient checking ($[L(w+\varepsilon) - L(w)]/\varepsilon$ ile gradyan doğrulama).
2. **Teğet = en iyi lineer yaklaşım** → gradient, kayıp yüzeyinin yerel lineer modeli; learning rate, bu yaklaşımın geçerli kaldığı komşuluğu belirler.
3. **"dt^2 terimini at"** → forward-mode autodiff dual sayıları ($\varepsilon^2 = 0$); türevin cebirsel mekaniği, sembolik açılım veya sayısal hata olmadan.
4. **Kuvvet kuralı** $d(t^n)/dt = n \cdot t^{n-1}$ → autodiff kural tablosunun temel girdileri (her primitif işlemin yerel türevi tanımlı).
5. **Türev 0 = kritik nokta** → loss yüzeyinde minimum/maksimum/eyer; sıfır gradyan "yerel olarak düz" demektir, "durağan" değil.
6. **"Anlık değil, sonlu dt"** → ayrıklaştırma: Euler çözücü adımı, SGD adımı; sürekli türev bunların

$dt \to 0$ limiti.

7. **Türevlenemezlik ($|t|$ köşesi)** → ReLU'nun 0'daki köşesi; pratikte alt-gradyan (subgradient) kullanılır ve tek bir noktadaki kırılma optimizasyonu nadiren bozar.

! Tek bir şey alıp gideceksen

Türev, "bir anda değişim" değildir — öyle bir şey yok. Türev, ds/dt fark oranının $dt \to 0$ iken yaklaştığı limittir: geometrik olarak teğet doğrunun eğimi, kavramsal olarak bir nokta civarındaki değişim oranının en iyi sabit yaklaşımı. Paradoks, soruyu yanlış sormaktan doğar.

3 Geometriyle Türev Formülleri

Kareyi, küpü, su birikintisini ve çemberi büyüt — formül ortaya çıkar

Bölüm bilgisi

- **Grant'ın videosu:** YouTube — Chapter 3: Derivative formulas through geometry (≈17 dk)
- **Kaynak:** 3Blue1Brown — Essence of Calculus
- **Okuma süresi:** ≈24 dk

3.1 Bu Derste Ne Var?

Ders 2'de türevin **ne olduğunu** (fark oranının limiti, teğet eğimi) tanımladık. Bu derste türev **formüllerini** hesaplamayı öğreniyoruz — ama ezberlenecek bir kurallar listesi olarak değil, **geometrik** olarak. Grant; x^2, x^3, x^n, $1/x$ ve sin gibi fonksiyonların türevlerini, küçük bir dürtmenin alana/hacme/yüksekliğe etkisini çizerek türetir. Sırrı: **küçük dürtmeler (tiny nudges)** hep işin kalbinde.

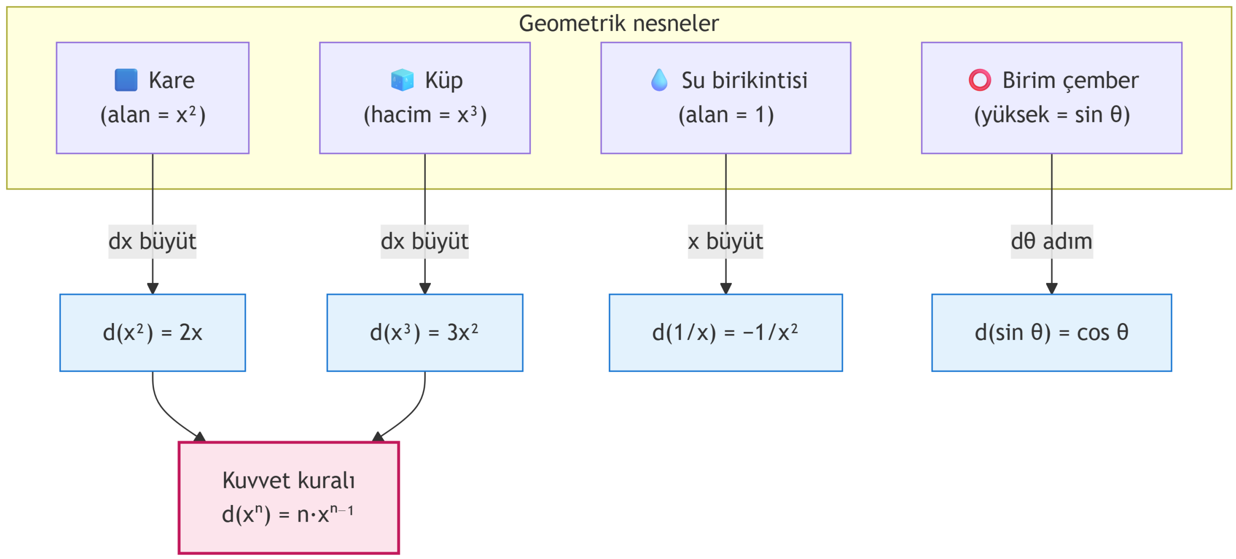
Üç ana fikir:

1. **Kuvvet kuralı** $d(x^n)/dx = n \cdot x^{n-1}$ — bir kareyi/küpü büyütmekten doğal olarak çıkar.
2. $(dx)^2$ **ihmal edilir** — "küçük bir değişimin karesi" ihmal edilebilir derecede küçüktür.
3. $d(\sin\theta)/d\theta = \cos\theta$ — birim çemberde benzer üçgenle, grafiğe bakmadan.

> *"never forget that tiny nudges are at the heart of derivatives."* — Grant, 1:32

Builder Notu — ML Köprüleri

- **Her temel fonksiyonun türev kuralı = autodiff'in "primitif türev tablosu".** x^2, x^n, $1/x$, sin… her biri framework'te kayıtlı bir yerel türev kuralıdır; backprop bunları zincirler.
- **"$(dx)^2$ ihmal"** → birinci-derece (lineer) yaklaşım; yine dual sayılar ve Jacobian-vector product'ın temeli.
- $d(1/x) = -1/x^2$ → normalizasyon gradyanları: $1/\sigma$ (batch/layer norm), softmax paydası $1/\Sigma$ türevleri hep bu biçimde.
- sin / cos **türevleri** → sinüzoidal **positional encoding** ve Fourier özniteliklerinin gradyanları; rotary embedding (RoPE).
- **Geometrik bakış** — "bu fonksiyon neyi temsil ediyor?" sorusu, bir işlemin gradyanını yapısal olarak anlamanın en sağlam yoludur.



Şekil 3.1: Bu bölümün geometrik temalı kavram haritası: her fonksiyon, kendi geometrisini büyüt → türev çıkar.

3.2 Neden Soyut Türevler? Küçük Dürtmeler Kalbi

Türevin ne demek olduğunu gördük; sıradaki adım onları **hesaplamak**: sana açık formüllü bir fonksiyon versem, türevinin formülünü bulabilmelisin. Peki neden calculus öğrencisinin vaktinin çoğu, somut hız problemleri yerine **soyut fonksiyonların** türevleriyle boğuşarak geçer?

Çünkü gerçek dünyadaki olguların çoğu — calculus ile analiz etmek istediğimiz şeyler — polinomlar, trigonometrik fonksiyonlar, üsteller ve benzeri **saf fonksiyonlarla** modellenir. Bu saf fonksiyonların değişim oranlarında akıcılık kazanırsan, somut durumlardaki değişimleri konuşmak için bir dil edinmiş olursun.

Ama bu süreç kolayca "bir kurallar listesi ezberlemek" gibi hissettirir — ve o his geldiğinde, türevlerin aslında **bir niceliğe verilen küçük bir değişimin, başka bir nicelikte yol açtığı küçük değişimle** ilişkisi olduğunu unutmak da kolaydır. Bu derste kuralları sezgisel ve geometrik düşüneceğiz; sakın küçük dürtmelerin türevin kalbinde olduğunu unutma.

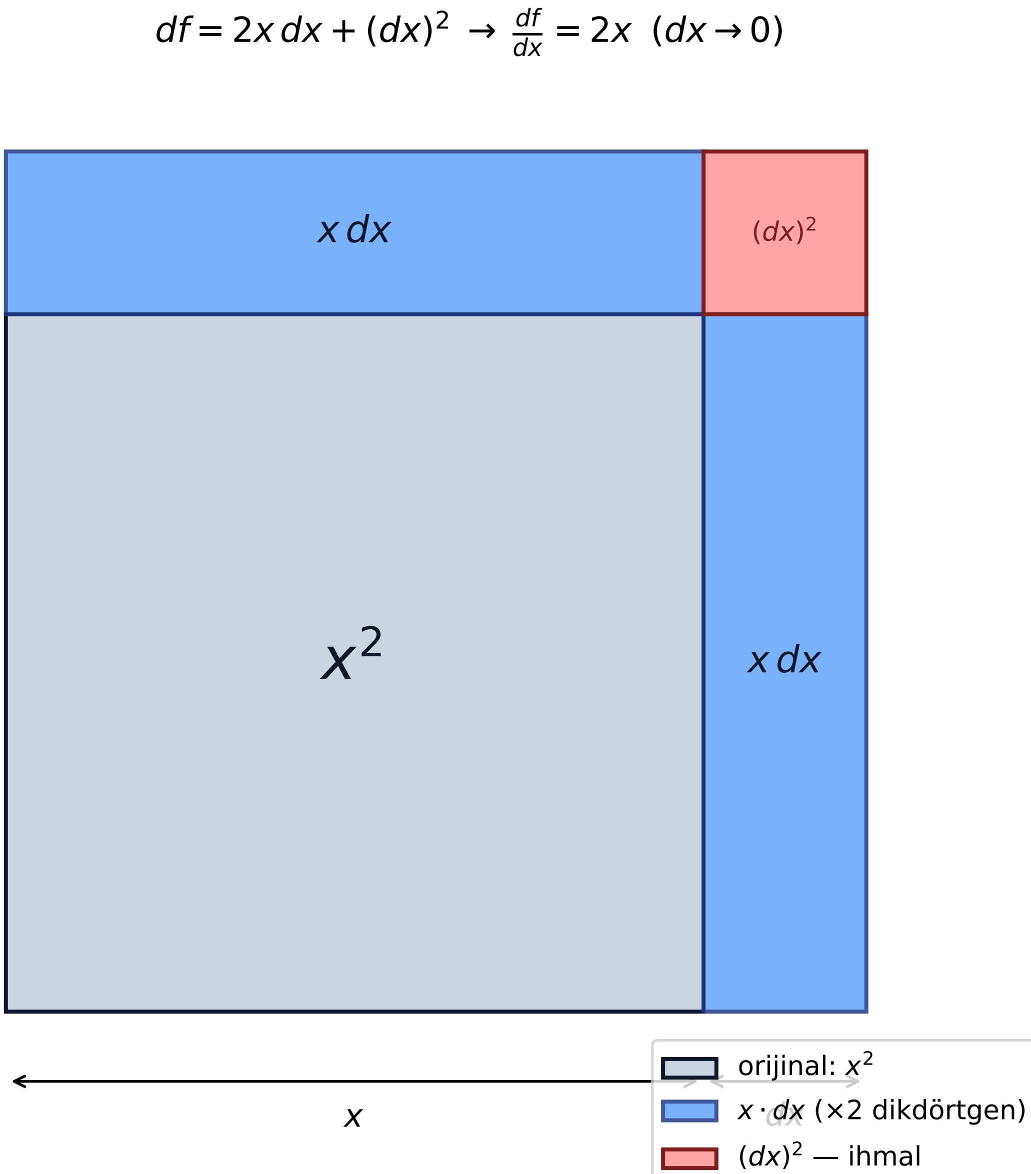
💡 Builder Notu — Primitif Kural Kaydı

Bu "saf fonksiyonların türev kurallarında akıcılık" tam olarak bir derin öğrenme framework'ünün yaptığı şeydir: PyTorch/JAX, bilinen her primitif işlem (üs alma, çarpma, exp, sin, log…) için yerel türevi bir tabloda tutar; karmaşık bir modelin gradyanını, bu temel kuralları **zincir kuralıyla** birleştirerek otomatik çıkarır. Yani bu ders, autodiff'in "kural kaydının" matematiksel kaynağıdır.

3.3 $d(x^2)/dx = 2x$ — Kareyi Büyütmek

$f(x) = x^2$ fonksiyonunu al. Türevini soruyorsam: bir x değerini (örneğin $x = 2$) alıp, ondan dx kadar büyük bir değerle karşılaştırınca, fonksiyon değeri df ne kadar değişir — ve özellikle df/dx oranı nedir?

Grafiğe bakarsak, df/dx teğet doğrunun eğimidir ve x büyüdükçe eğim artar ($x = 0$'da yatay, $x = 1$'de eğimli, $x = 2$'de daha dik). Ama grafiğe bakmak, kesin formülü bulmanın iyi yolu değil. Bunun için x^2'nin **ne anlama geldiğine** daha doğrudan bakalım: kenar uzunluğu x olan bir **kare** düşün.



Şekil 3.2: Kenarı x olan kareyi dx büyüt: alan değişimi df = 2 dikdörtgen ($2x \cdot dx$, mavi) + minik kare (dx^2, kırmızı, ihmal).

x'i küçük bir dx kadar büyütürsen, karenin alanındaki değişim df'tir. Diyagramda üç yeni alan parçası belirir: **iki ince dikdörtgen** ve **minik bir kare**. İki dikdörtgenin her birinin kenarları x ve dx, yani toplam $2 \cdot x \cdot dx$ yeni alan. Minik karenin alanı ise $(dx)^2$. Örneğin $x = 3$, $dx = 0{,}01$ ise: ince dikdörtgenler $2 \cdot 3 \cdot 0{,}01 = 0{,}06$ (dx'in ~ 6 katı), ama minik kare yalnızca $0{,}0001$ — ihmal edilebilir.

> *"a good rule of thumb is that you can ignore anything that includes a dx raised to a power greater than one … a tiny change squared is a negligible change."* — Grant, 3:57

Geriye df'in dx'in bir katı olması kalır; o kat da türevdir:

$$\frac{df}{dx} = \frac{2x\,dx + (dx)^2}{dx} = 2x + dx \;\to\; 2x$$

$x = 3$'te oran $2 \cdot 3 = 6$, $x = 5$'te $2 \cdot 5 = 10$ birim alan / birim uzunluk.

Builder Notu — Lineer vs Eğrilik

İki dikdörtgen (mertebe dx) "korunur", minik kare (mertebe dx^2) "atılır" — bu, türevin **birinci-derece (lineer)** bir nesne olmasının geometrik yüzüdür. ML'de gradient de tam böyle: parametredeki küçük değişimin kayba **lineer** katkısını yakalar; ikinci derece etkiler (eğrilik) Hessian'a kalır. dx^2 atmak, "yerel olarak fonksiyon düzdür" yaklaşımının ta kendisidir.

3.4 $d(x^3)/dx = 3x^2$ — Küpü Büyütmek

Şimdi $f(x) = x^3$. Ders 2'de cebirsel yaptığımız şeyin geometrik hâli bu. x^3'ü, kenar uzunluğu x olan gerçek bir **küpün hacmi** olarak düşünebiliriz. x'i dx kadar büyütünce ortaya çıkan hacim artışı, kenarı $x + dx$ olan kübün, kenarı x olan orijinal kübe ait olmayan kısmıdır.

Bu yeni hacmin neredeyse tamamı **üç kare yüzden** gelir ($dx \to 0$ iken bu üç yüz, yeni hacmin %100'üne yaklaşır). Her ince yüzün hacmi $x^2 \cdot dx$ (yüzün alanı çarpı dx kalınlık), yani toplam $3x^2 \cdot dx$. Kenarlardaki ince çubuklar ve köşedeki minik küp ise $(dx)^2$ ve $(dx)^3$ ile orantılı — dx'e bölününce hayatta kalmazlar, güvenle atılır.

$$df \approx 3x^2\,dx \qquad \frac{df}{dx} = 3x^2$$

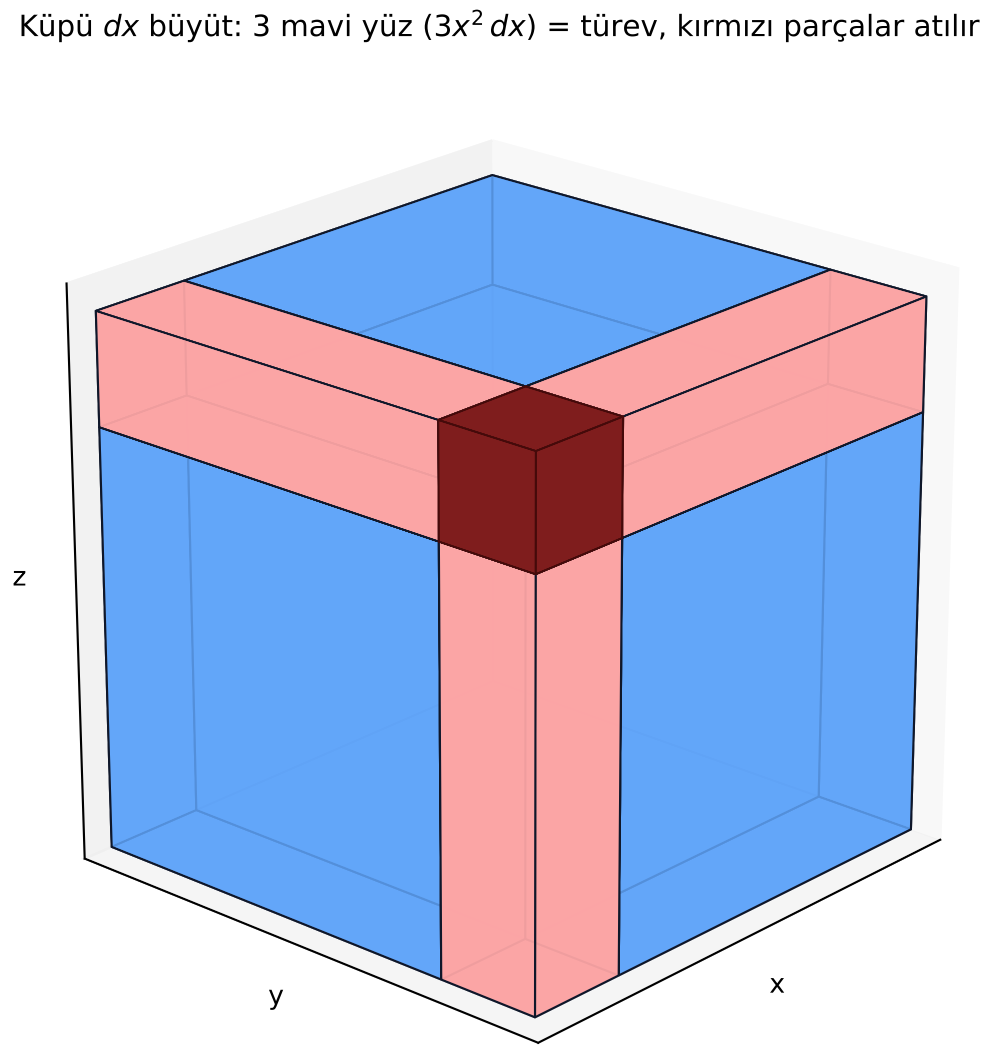
Builder Notu — Boyut → Katsayı

$x^2 \to 2$ dikdörtgen, $x^3 \to 3$ kare yüz. Bu "boyut sayısı kadar yüzey katkısı" örüntüsü, kuvvet kuralındaki katsayının (n) nereden geldiğini gösterir. ML açısından önemli olan ders: bir fonksiyonun gradyanını, onun **geometrik/yapısal anlamından** okumak, formülü ezberlemekten hem daha güvenilir hem genelleştirilebilirdir.

3.5 Kuvvet Kuralı: $d(x^n)/dx = n \cdot x^{n-1}$

Pratikte x^2 için her seferinde kareyi, x^3 için küpü düşünmezsin; ikisi de tanınabilir bir örüntüye uyar. x^4'ün türevi $4x^3$, x^5'in türevi $5x^4$, ve genel olarak:

$$\frac{d}{dx}\,x^n = n\,x^{n-1}$$



Şekil 3.3: Küpü dx büyüt: üç ince yüz ($3 \cdot x^2 \cdot dx$, mavi) ana katkı; üç çubuk ($3 \cdot x \cdot dx^2$) ve köşedeki minik küp (dx^3) ihmal edilir.

Buna **kuvvet kuralı** denir. Sembolik olarak: üs öne iner, geriye kendinden bir eksiğini bırakır. Ama neden çalıştığını $n = 2, 3$'ün ötesinde de görelim. x'i $x + dx$'e ittiğinde, $x + dx$'in n kopyasını çarpman gerekir:

$$(x + dx)^n = x^n + n\, x^{n-1}\, dx + \cdots$$

İlk terim x^n (orijinal karenin alanı / kübün hacmi gibi). Sonraki terimler için: n parantezden **yalnızca birinden** dx, geri kalanından x seçersin — bunu yapmanın n yolu var, her biri $x^{n-1} \cdot dx$ verir, toplam $n \cdot x^{n-1} \cdot dx$. Açılımdaki diğer tüm terimler $(dx)^2$ ve üzeri içerir; dx'e bölününce hayatta kalmazlar. Geriye $n \cdot x^{n-1}$ kalır.

> *"the derivative of x to the n for any power n is n times x to the n minus 1 … the power rule."* — Grant, 7:28

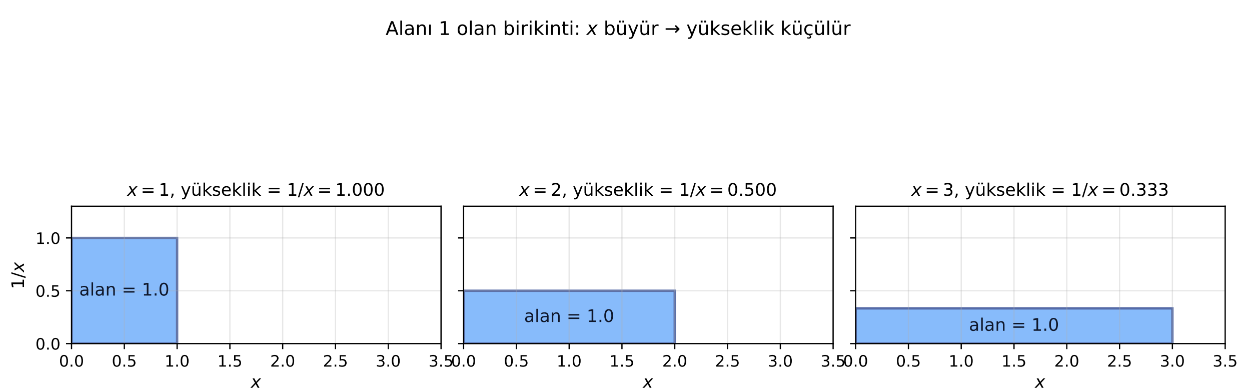
Builder Notu — Birinci Derece

Kuvvet kuralı, autodiff'in en temel kayıtlı kuralıdır. Ama asıl ML dersi açılımda gizli: $(x + dx)^n$'in yalnızca **birinci-derece** terimini tutmak, forward-mode autodiff'in dual sayılarıyla ($\varepsilon^2 = 0$) birebir aynıdır. Yüksek-dereceli terimleri taşımak ikinci-derece bilgi (eğrilik/Hessian) gerektirir; çoğu eğitim yalnızca birinci dereceyle (gradient) çalışır çünkü ucuz ve ölçeklenebilir.

3.6 $d(1/x)/dx = -1/x^2$ — Su Birikintisi

$f(x) = 1/x$'i ele al. Bir yandan kuvvet kuralını körü körüne uygulayabilirsin: $1/x = x^{-1}$, üs (-1) öne iner ve bir eksiği (-2) kalır $\to -x^{-2} = -1/x^2$. Ama bunu geometrik olarak da görelim.

$1/x$, "x ile çarpınca 1 eden sayı" demek. Şöyle hayal et: alanı 1 olan dikdörtgen bir su birikintisi; genişliği x ise, yüksekliği $1/x$ olmak zorunda (çünkü alan = 1). x'i 2'ye uzatırsan yükseklik $1/2$'ye iner; 3'e çıkarırsan $1/3$'e sıkışır.



Şekil 3.4: Alanı sabit 1 olan dikdörtgen su birikintisi: genişlik x artarsa, yükseklik $1/x$ azalır. $d(1/x)$, üstten kaybedilen ile sağda kazanılan alanları eşitleyerek $-1/x^2$ verir.

Şimdi x'i dx kadar büyüt. Birikintinin alanı 1 sabit kalsın diye yükseklik ne kadar değişmeli? Genişliği dx artırmak sağda yeni alan ekler ($(1/x) \cdot dx$ kadar); bunu dengelemek için yükseklik $d(1/x)$ kadar **azalmalı** (negatif), öyle ki üstten kaybedilen alan ($x \cdot |d(1/x)|$) sağda kazanılanı götürsün:

$$\frac{1}{x}\,dx = -x \cdot d\left(\frac{1}{x}\right) \qquad \frac{d}{dx}\left(\frac{1}{x}\right) = -\frac{1}{x^2}$$

Körü körüne kuvvet kuralının verdiğiyle birebir aynı — ama şimdi **neden** negatif ve neden $1/x^2$ olduğunu görüyorsun. (Aynı muhakemeyle $\sqrt{x}$'in türevini de bulmayı dene.)

> **Builder Notu — Normalizasyon**
>
> $d(1/x) = -1/x^2$ işareti ve büyüklüğü ML'de her yerde: bir kaybı $1/\sigma$ ile ölçeklerken (batch/layer norm) veya softmax paydası $1/\Sigma$ türevini alırken, "payda büyürse çıktı küçülür, hem de kare oranında" davranışı tam budur. Ters-orantılı bir niceliğin gradyanı her zaman negatif ve karesel sönümlüdür — normalizasyon katmanlarının geri yayılımının özü.

3.7 $d(\sin\theta)/d\theta = \cos\theta$ — Birim Çember

Son olarak trigonometrik bir fonksiyon: sin. Birim çemberi (orijin merkezli, yarıçapı 1) hatırla. Bir θ değeri için, en sağ noktadan başlayıp çember üzerinde θ kadar **yay uzunluğu** yürürsün; yarıçap 1 olduğundan açı da tam θ radyandır. $\sin(\theta)$, o noktanın x-ekseni üzerindeki **yüksekliğidir**. θ arttıkça yükseklik -1 ile 1 arasında inip çıkar — sin grafiği o klasik dalga.

Grafiğe bakarak türevin şeklini sezebiliriz: $\theta = 0$'da eğim pozitif (sin artıyor), tepede sıfıra iner, sonra bir süre negatif, sonra yine sıfır… Tepe ve çukurlar cosine ile birebir hizalanır, dolayısıyla türevin $\cos\theta$ olduğunu tahmin edersin. Doğru — ama neden tam olarak cosine, sadece "benzer şekilli yeni bir fonksiyon" değil?

Kesin neden, grafiğe değil fonksiyonun temsil ettiği şeye bakmakta. Çember üzerindeki noktaya yakınlaş ve çevre boyunca küçük bir $d\theta$ adımı at. Bu kadar yakında çember neredeyse düz bir doğru gibi; küçük bir **dik üçgen** düşün: hipotenüsü çevre boyunca atılan $d\theta$ adımı, dikey kenarı ise yükseklikteki değişim, yani $d(\sin\theta)$. Bu minik üçgen, açısı θ ve hipotenüsü yarıçap (uzunluk 1) olan **büyük üçgene benzerdir**.

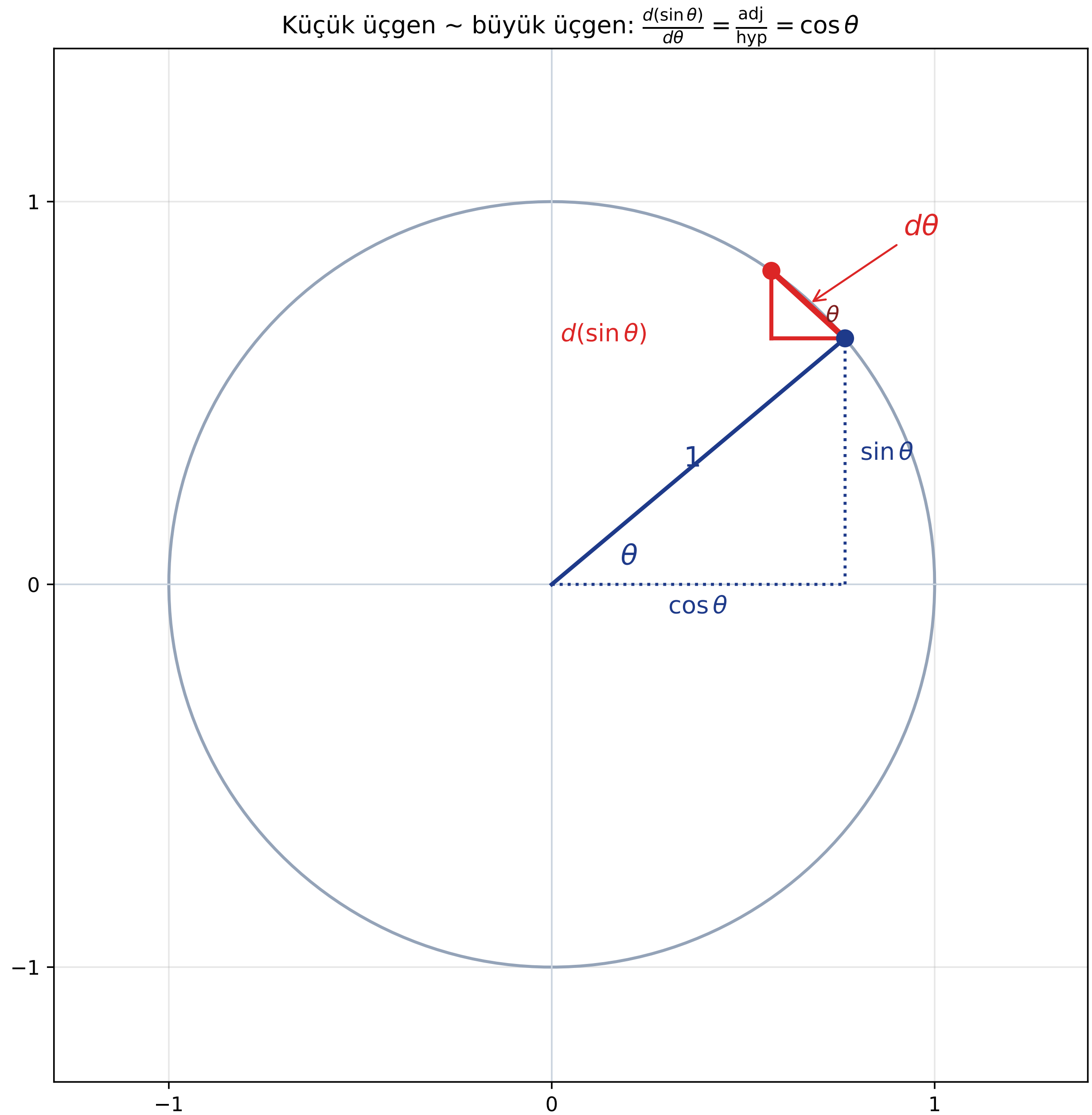
$$\frac{d(\sin\theta)}{d\theta} = \frac{\text{adjacent}}{\text{hypotenuse}} = \cos\theta$$

θ'ya komşu kenar bölü hipotenüs — bu zaten cosine'ın tanımı. Demek ki:

$$\frac{d}{d\theta}\sin(\theta) = \cos(\theta)$$

"adjacent divided by hypotenuse, that's exactly what the cosine of theta means." — Grant, 16:24

> **Builder Notu — Fourier ve RoPE**
>
> sin'in türevinin cos olması, sinüzoidlerin türev altında kendi içlerinde dönmesi demektir ($\sin \to \cos \to -\sin \to -\cos \to \sin$). Bu kapalılık, **Fourier analizi** ve transformer'lardaki **sinüzoidal positional encoding**'in temelidir: konum bilgisini farklı frekanslı $\sin / \cos$ çiftleriyle kodlarsın, ve türev/kaydırma işlemleri bu tabanda temiz (lineer) kalır. Rotary positional embedding (RoPE) de bu döngüsel yapıyı doğrudan kullanır.



Şekil 3.5: Birim çember üzerinde θ noktası ile $d\theta$ adımı: küçük üçgen büyük üçgene benzer. Türev = komşu kenar / hipotenüs = $\cos\theta$.

3.8 Bu Dersin Özeti

1. Türev formülleri ezberlenecek kurallar değil; “girdiye küçük dürtme → çıktıda küçük değişim” geometrisinden çıkar.
2. $d(x^2)/dx = 2x$: kareyi dx büyüt → iki $x \cdot dx$ çubuğu; minik kare (dx^2) atılır.
3. $d(x^3)/dx = 3x^2$: küpü dx büyüt → üç $x^2 \cdot dx$ yüzü; kenar/köşe parçaları atılır.
4. Kuvvet kuralı $d(x^n)/dx = n \cdot x^{n-1}$: $(x + dx)^n$ açılımında dx’i n paranteziden birinden seçmenin n yolu → $n \cdot x^{n-1} \cdot dx$.
5. $(dx)^2$ ve daha yüksek kuvvetler her zaman atılır — türev özünde **birinci-derece** bir nesnedir.
6. $d(1/x)/dx = -1/x^2$: alanı sabit su birikintisi; körü körüne kuvvet kuralıyla ($x^{-1} \to -x^{-2}$) aynı sonuç.
7. $d(\sin\theta)/d\theta = \cos\theta$: birim çemberde benzer üçgen; komşu kenar / hipotenüs = $\cos\theta$.

! Tek bir cümle

Her türev formülü, fonksiyonun temsil ettiği geometrik nesneyi (kare, küp, su birikintisi, çember) dx kadar büyütüp “birinci-derece” değişimi okumaktan çıkar; $(dx)^2$ atılır, çünkü türev özünde yerel bir lineer yaklaşımdır.

3.9 Kontrol Soruları

i Soru 1: d(x^4)/dx’i, (x+dx)^4 açılımındaki birinci-derece terimi düşünerek bul.

Cevap: $(x + dx)^4 = x^4 + 4x^3 \cdot dx + (dx^2$ ve daha yüksek terimler$)$. İlk terim x^4 orijinal değer; dx’i dört parantezden birinden seçmenin 4 yolu var, her biri $x^3 \cdot dx$ → toplam $4x^3 \cdot dx$. Geri kalan terimler dx^2 içerir, dx’e bölününce kaybolur. Sonuç: $d(x^4)/dx = 4x^3$ (kuvvet kuralıyla uyumlu).

i Soru 2: d(x^2)/dx’te minik kareyi ((dx)^2) neden atıyoruz ama iki dikdörtgeni (2x·dx) tutuyoruz?

Cevap: İkisi de df’e (alan değişimine) katkı verir. Ama df/dx oranına bakınca: $2x \cdot dx$ terimi dx’e bölününce $2x$ (sabit) kalır; $(dx)^2$ terimi ise dx’e bölününce dx olur ve $dx \to 0$’da sıfıra gider. Yani limitte yalnızca **birinci-derece** (tek dx’li) terim hayatta kalır; $(dx)^2$ “ikinci-derece” katkıdır ve türevde görünmez.

i Soru 3: d(√x)/dx’i kuvvet kuralıyla bul ve x = 0’da ne olduğunu söyle.

Cevap: $\sqrt{x} = x^{1/2}$. Kuvvet kuralı: üs $(1/2)$ öne iner, bir eksiği $(-1/2)$ kalır → $(1/2) \cdot x^{-1/2} = 1/(2\sqrt{x})$. x büyüdükçe türev küçülür (eğri yatıklaşır). $x = 0$’da $1/(2 \cdot 0)$ tanımsızdır — grafiğin orada **dikey teğeti** vardır (eğim sonsuza gider).

Soru 4: (Builder) sin'in türevinin cos olması, positional encoding'de neden işe yarar?

Cevap: sin ve cos türev (ve kaydırma) altında birbirine dönüşür: $\sin \to \cos \to -\sin \to -\cos \to \sin$. Bu kapalılık sayesinde, konumu farklı frekanslı sin / cos çiftleriyle kodlarsan, bir konum kaymasını sabit bir lineer dönüşüm (rotasyon) temsil eder. Transformer'ın göreli konumu kolayca öğrenmesini sağlayan budur; rotary positional embedding (RoPE) bu döngüsel/rotasyonel yapıyı doğrudan kullanır.

3.10 Egzersizler

Egzersiz 1. $(x + dx)^5$ açılımının birinci-derece terimini bularak $d(x^5)/dx$'i hesapla; kuvvet kuralının ($5x^4$) doğrulandığını göster.

Egzersiz 2. Kareyi büyütme diyagramında $x = 10$, $dx = 0{,}1$ al. İki ince dikdörtgenin toplam alanını ($2 \cdot x \cdot dx$) ve minik karenin alanını ($(dx)^2$) sayısal hesapla. Minik kare, dikdörtgenlerin yüzde kaçı? dx'i $0{,}01$'e düşürünce bu oran ne olur?

Egzersiz 3. $d(\cos\theta)/d\theta$'yı birim çemberde benzer üçgenle türet. (İpucu: $d\theta$ adımının yataydaki bileşenine ve işaretine bak; sonucun $-\sin\theta$ çıkması gerekir.)

Egzersiz 4. *(Python — görsel doğrulama)* SymPy ile temel türevleri sembolik al, sonra sin ile cos'u üst üste çizip türev ilişkisini gör.

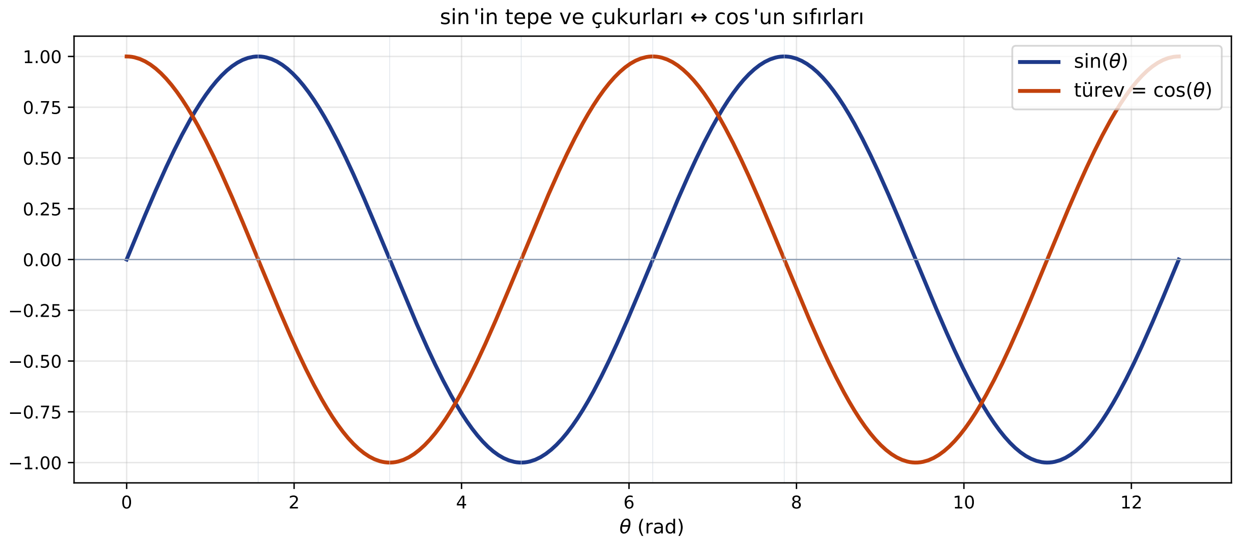
```
x**2 -> 2*x
x**3 -> 3*x**2
x**5 -> 5*x**4
1/x -> -1/x**2
sqrt(x) -> 1/(2*sqrt(x))
sin(x) -> cos(x)
```

Egzersiz 5. *(Sonraki dersin habercisi)* $f(x) = \sin(x^2)$ gibi bir **bileşke** fonksiyonu düşün: dış fonksiyon sin (türevi cos), iç fonksiyon x^2 (türevi $2x$). Bu ikisini nasıl birleştirip f'in türevini bulursun? Tahminini yaz — Ders 4, bileşke fonksiyonlar için **zincir kuralını** anlatacak.

3.11 Sonraki Ders İçin Hazırlık

Ders 4: Zincir Kuralı ve Çarpım Kuralı (Görsel)

Bu derste tek tek fonksiyonların türevlerini öğrendik. Ders 4, bunları **birleştirmeyi** ele alıyor: toplam ($f + g$), çarpım ($f \cdot g$) ve bileşke ($f(g(x))$) fonksiyonların türevleri. Her birini yine geometrik/sezgisel olarak — çarpımı bir dikdörtgenin alanı, bileşkeyi ardışık dürtmeler olarak — göreceğiz. Zincir kuralı, ML için en kritik olanı: backprop'un tam kalbidir.



Şekil 3.6: sin'in türevi cos: sin'in tepe ve çukurlarında cos sıfıra iner, ve tersi. Türev geometrisi grafiklerden net görülür.

3.12 Anahtar Kavramlar (Cheat Sheet)

Kavram	Tanım	Grant'ta
$d(x^2)/dx = 2x$	Kareyi dx büyüt: iki $x \cdot dx$ çubuğu (minik kare atılır)	2m35
$d(x^3)/dx = 3x^2$	Küpü dx büyüt: üç $x^2 \cdot dx$ yüzü	4m42
$(dx)^2$ **ihmali**	Küçük değişimin karesi negligible; türev birinci-derecedir	3m57
Kuvvet kuralı	$d(x^n)/dx = n \cdot x^{n-1}$; $(x+dx)^n$ açılımından	7m28
$d(1/x)/dx = -1/x^2$	Alanı sabit su birikintisi (genişlik x, yükseklik $1/x$)	10m08
$d(\sqrt{x})/dx = 1/(2\sqrt{x})$	$x^{1/2}$ için kuvvet kuralı	12m25
$d(\sin\theta)/d\theta = \cos\theta$	Birim çemberde benzer üçgen: adjacent / hypotenuse	12m38
Geometrik türetim	"Fonksiyon neyi temsil ediyor?" → diyagramı büyüt	2m30
Tiny nudge sezgisi	Türevin kalbi: küçük dürtme → küçük değişim	1m32

3.13 ML Bağlantıları Özeti

7 köprü

1. **Temel türev tablosu** → autodiff'in primitif kural kaydı: her işlemin (üs, çarpım, exp, sin, log) yerel türevi tanımlı.
2. $(dx)^2$ **ihmali** → birinci-derece/lineer yaklaşım; dual sayılar ($\varepsilon^2 = 0$); gradient eğriliği (Hessian) taşımaz.
3. **Kuvvet kuralı katsayısı** → $n \cdot x^{n-1}$; "n yüz/çubuk" geometrisi, gradyan büyüklüğünün ölçeğini belirler.
4. $d(1/x) = -1/x^2$ → normalizasyon gradyanları ($1/\sigma$, softmax paydası $1/\Sigma$): negatif ve karesel sönümlü.
5. sin / cos **kapalılığı** → Fourier öznitelikleri, sinüzoidal positional encoding, rotary embedding (RoPE).
6. **"Fonksiyon neyi temsil ediyor" geometrik bakışı** → bir operatörün gradyanını ezberlemeden, yapısal olarak okumak.
7. **(Sonraki) zincir kuralı** → backprop; bu dersin temel kuralları, bileşke fonksiyonlarda zincirlenecek.

! Tek bir şey alıp gideceksen

Türev formülleri gökten inmez. x^2'yi bir kare, x^3'ü bir küp, $1/x$'i alanı sabit bir su birikintisi, $\sin\theta$'yı bir çember yüksekliği olarak görüp dx kadar büyüt — formül, birinci-derece değişimi okumaktan kendiliğinden çıkar. $(dx)^2$ hep atılır, çünkü türev yerel bir lineer yaklaşımdır.

4 Zincir Kuralı ve Çarpım Kuralı

Toplam + çarpım + bileşke — backprop'un matematiksel kalbi

Bölüm bilgisi

- **Grant'ın videosu:** YouTube — Chapter 4: Visualizing the chain rule and product rule (≈16 dk)
- **Kaynak:** 3Blue1Brown — Essence of Calculus
- **Okuma süresi:** ≈24 dk

4.1 Bu Derste Ne Var?

Ders 3'te tek tek fonksiyonların (x^2, x^3, $1/x$, sin) türevlerini öğrendik. Ama gerçek dünyayı modelleyen fonksiyonlar genelde bunların bir **karışımıdır**. Bu derste fonksiyonları birleştirmenin üç temel yolunu ve her birinin türev kuralını — yine küçük dürtmelerle, geometrik olarak — görüyoruz. En önemlisi **zincir kuralı**: makine öğrenmesindeki backprop'un tam kalbi.

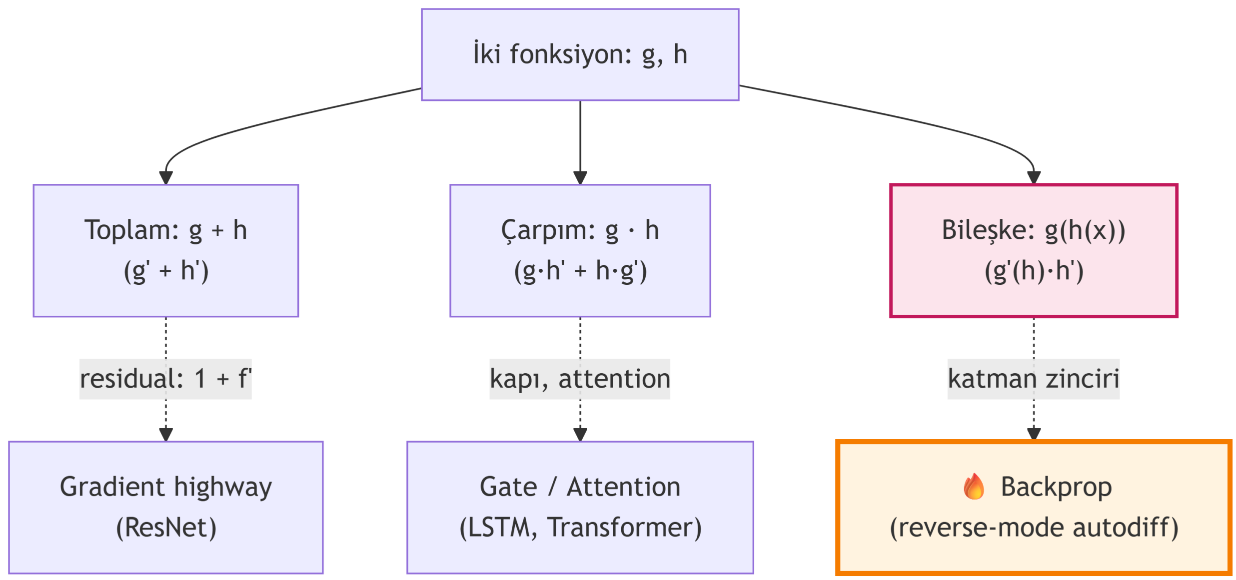
Üç ana fikir:

1. **Toplam kuralı:** $d(g+h) = g' + h'$ (en kolayı).
2. **Çarpım kuralı:** $d(g \cdot h) = g \cdot h' + h \cdot g'$ (ayarlanabilir kutunun alanı).
3. **Zincir kuralı:** $d(g(h(x))) = g'(h(x)) \cdot h'(x)$ (üç sayı doğrusu; dış türev çarpı iç türev).

> *"these are natural patterns, things that you too could have discovered just by patiently thinking through what a derivative actually means."* — Grant, 15:31

Builder Notu — ML Köprüleri

- **Zincir kuralı = backprop.** Bir sinir ağı = iç içe fonksiyonlar (katmanlar); gradyan, zincir kuralıyla çıkıştan girdiye doğru **geriye akar**. Bu dersin en kritik ML bağlantısı — derin öğrenme bu kural üzerine kuruludur.
- **Toplam kuralı** → çoklu kayıpların (multi-task loss = $\sum L_i$) gradyanı toplanır; **residual bağlantı** $d(x + f(x)) = 1 + f'$ — "gradient highway" (ResNet'in derin ağları eğitebilmesinin nedeni).
- **Çarpım kuralı → kapı (gate)** mekanizmaları (LSTM/GRU'da $h \cdot g$), attention'ın ağırlıklı toplamı, element-wise çarpım gradyanları.
- **"Katmanları soy"** → computational graph: her düğüm bir primitif işlem; autodiff bu grafi ters yönde gezerek gradyanı toplar.
- dh**'lerin sadeleşmesi** → reverse-mode autodiff: ara türevler (Jacobian'lar) çarpılarak zincirlenir;



Şekil 4.1: Üç birleştirme yolu ve backprop bağlantısı. Zincir kuralı, derin ağlardaki tüm öğrenmenin matematiksel motoru.

> "sadece notasyon hilesi değil", küçük dürtmelerin gerçek aktarımı.

4.2 Üç Birleştirme Yolu: Topla, Çarp, Bileşke

Türevini bildiğimiz basit fonksiyonlardan karmaşık fonksiyonlar kurmanın aslında yalnızca **üç temel yolu** vardır:

- **Toplama:** $g(x) + h(x)$
- **Çarpma:** $g(x) \cdot h(x)$
- **Bileşke (composition):** birini diğerinin içine koymak, $g(h(x))$

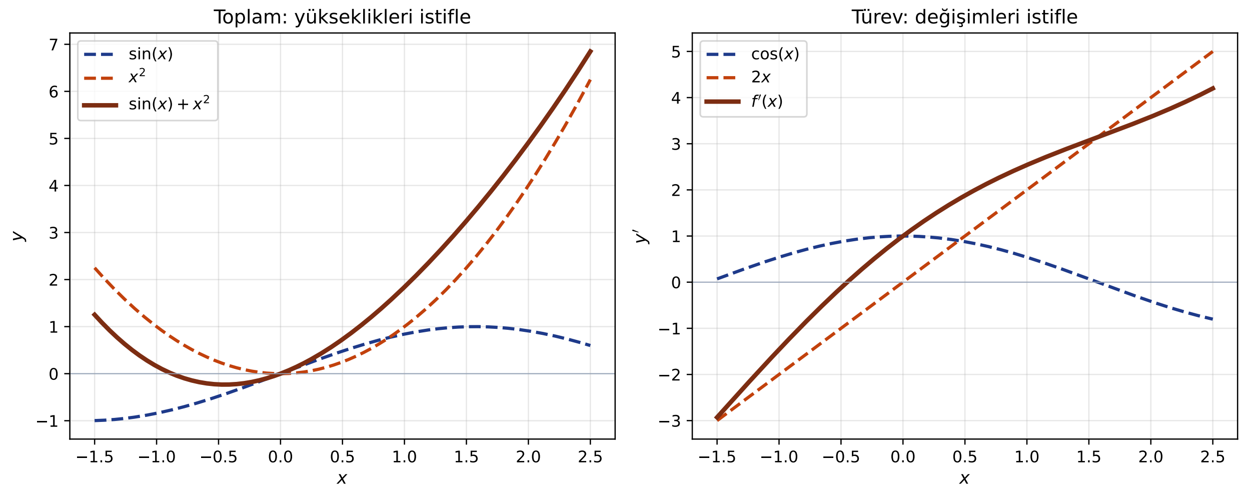
Çıkarma aslında "ikinciyi -1 ile çarpıp toplamak"tır; bölme ise "$1/x$ ile bileşke alıp çarpmak". Yani ne kadar canavarsı görünürse görünsün, karşılaştığın fonksiyonların çoğu bu üç birleştirmenin katman katman istiflenmesidir. Türevin bu üç türle nasıl oynadığını bilirsen, en karmaşık ifadeyi bile adım adım **katmanlarını soyarak** türetebilirsin.

> 💡 Builder Notu — Computational Graph
>
> "Karmaşık bir ifadeyi üç temel birleştirmeye ayırıp katman katman soymak", bir **computational graph**'ın tanımıdır. PyTorch/JAX bir modeli tam olarak böyle temsil eder: her düğüm bir toplama, çarpma veya bir primitif fonksiyon (exp, sin, matmul); autodiff bu grafi kurar, sonra zincir + çarpım + toplam kurallarını mekanik olarak uygular. Yani bu üç kural, tüm otomatik türevin tam kümesidir — başka kurala gerek yok.

4.3 Toplam Kuralı: $d(g + h) = g^{'} + h^{'}$

En kolayı: iki fonksiyonun toplamının türevi, türevlerinin toplamıdır. Yine de küçük dürtmelerle düşünmeye değer, çünkü çarpım ve bileşke o kadar düz olmayacak.

Örnek: $f(x) = \sin(x) + x^2$. Her girdide $\sin(x)$ ile x^2'nin değerlerini toplarsın.



Şekil 4.2: Toplam kuralı: $f(x) = \sin(x) + x^2$. Yükseklikler istiflenir; değişimler de istiflenir → türevler de toplanır: $f'(x) = \cos(x) + 2x$.

Girdiyi dx kadar ittiğinde, toplam yükseklikteki değişim df, sin grafiğindeki değişim ile x^2 grafiğindeki değişimin toplamıdır:

$$df = d(\sin x) + d(x^2) \approx (\cos x \cdot dx) + (2x \cdot dx)$$

dx'e bölersek:

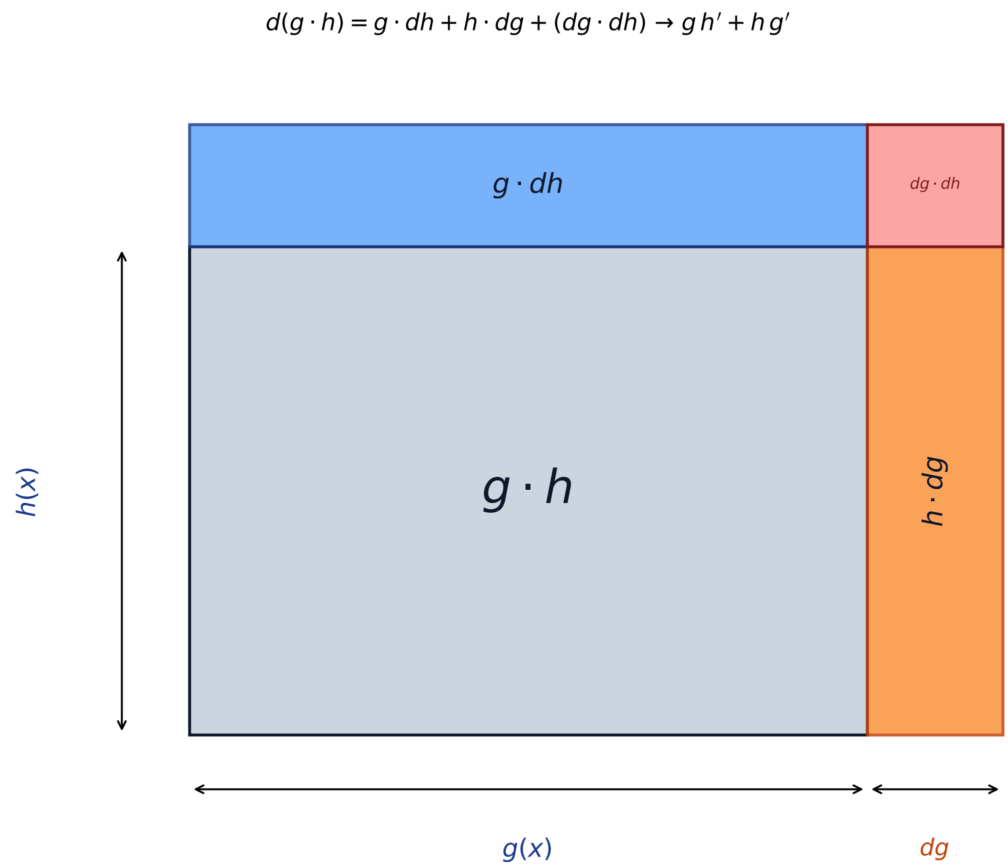
$$\frac{d}{dx}\left(\sin x + x^2\right) = \cos x + 2x$$

Builder Notu — Multi-task & Residual

Toplam kuralı ML'de iki yerde her gün karşına çıkar. Birincisi **çoklu kayıp**: toplam kayıp $L = L_1 + L_2 + \ldots$ ise gradyan da $\nabla L = \nabla L_1 + \nabla L_2 + \ldots$ — her terimi ayrı geri yayıp toplarsın. İkincisi **residual bağlantı**: $d/dx(x + f(x)) = 1 + f'(x)$. Buradaki "+1", gradyanın kimlik yoluyla hiç sönmeden geçebildiği bir "gradient highway" yaratır; ResNet'lerin çok derin ağları eğitebilmesinin matematiksel nedeni budur.

4.4 Çarpım Kuralı: Ayarlanabilir Kutu

Çarpım için grafik değil, **alan** en iyi görseldir. $f(x) = \sin(x) \cdot x^2$'yi, kenarları $\sin(x)$ ve x^2 olan bir kutunun alanı olarak düşün. Bu kenarlar x ile değişir: x'i 0'dan artırınca üst kenar ($\sin x$) önce 1'e kadar büyür, sonra azalır; yan kenar (x^2) sürekli büyür. $f(x)$, bu kutunun alanıdır.



Şekil 4.3: Çarpım kuralı: kenarları $g(x), h(x)$ olan ayarlanabilir kutu. x dürtülünce: alttaki ince dikdörtgen $(g \cdot h')$ + sağdaki ince dikdörtgen $(h \cdot g')$. Köşedeki minik kare ihmal.

x'i dx kadar ittiğinde alan nasıl değişir? Üç yeni parça belirir: **alttaki** ince dikdörtgen (genişliği $\sin x$, yüksekliği $d(x^2)$), **sağdaki** ince dikdörtgen (yüksekliği x^2, genişliği $d(\sin x)$) ve köşedeki minik parça — son ikisinin çarpımı $(dx)^2$ ile orantılı olduğundan ihmal edilir.

$d(x^2) \approx 2x \cdot dx$ ve $d(\sin x) \approx \cos x \cdot dx$ koyup dx'e bölersek:

$$\frac{d}{dx}\left(\sin x \cdot x^2\right) = \sin x \cdot 2x + x^2 \cdot \cos x$$

Burada sin ve x^2'ye özgü hiçbir şey yok; aynı muhakeme herhangi iki g ve h için geçerli:

$$(g \cdot h)' = g \cdot h' + h \cdot g'$$

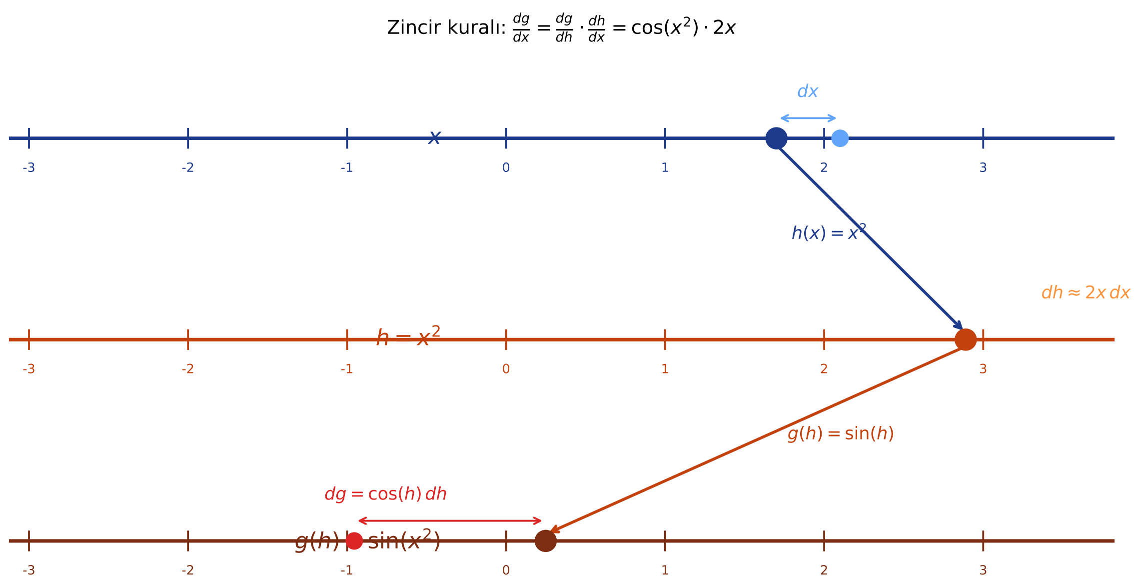
Akılda tutma tekerlemesi: **“sol d sağ + sağ d sol”** — sol fonksiyon çarpı sağın türevi, artı sağ fonksiyon çarpı solun türevi.

> **Builder Notu — Gate ve Attention**
>
> Çarpım kuralı, iki sinyalin çarpıldığı her yerde devrededir. **Kapı (gate) mekanizmaları**: LSTM/GRU’da “unut kapısı çarpı hücre durumu” $(g \cdot h)$ biçiminde; geri yayılımda her iki faktöre de gradyan akar $(g \cdot h' + h \cdot g')$. **Attention** da bir ağırlık (softmax çıktısı) çarpı değer toplamıdır; ağırlık ve değerin ikisi de öğrenilir, ikisine de çarpım kuralıyla gradyan gider.

4.5 Zincir Kuralı: Üç Sayı Doğrusu

Üçüncü ve ML’de en sık karşımıza çıkan birleştirme: birini diğerinin içine koymak (bileşke). Örneğin x^2’yi sin’in içine sokup $\sin(x^2)$ elde et. Türevi ne?

Bunu görselleştirmek için **üç sayı doğrusu** kullanalım: birincisi x değerini, ikincisi x^2 değerini, üçüncüsü $\sin(x^2)$ değerini tutsun. x^2 fonksiyonu seni 1. doğrudan 2.’ye, sin fonksiyonu 2.’den 3.’ye taşır.



Şekil 4.4: Üç sayı doğrusu: $x \to h(x) \to g(h(x))$. Bir dx dürtmesi, $dh = h'(x) \cdot dx$ üzerinden $dg = g'(h(x)) \cdot dh = g'(h) \cdot h' \cdot dx$ olarak aktarılır — zincir kuralı.

x’i 3’e götürürsen 2. değer 9’a, 3. değer $\sin(9)$’a sabitlenir. Türev için x’i dx kadar dürt. Ortadaki değerin değişimi (x^2’nin değişimi) $d(x^2)$’dir; buna kısaca dh diyelim ($h = x^2$). O zaman alttaki değer $\sin(h)$’nin değişimi, $d(\sin h) = \cos(h) \cdot dh$ olur (sin’in türevi cos olduğundan). Şimdi katmanları geri açalım: h yerine x^2 koy, dh yerine $2x \cdot dx$ koy:

$$\frac{d}{dx}\sin(x^2) = \cos(x^2) \cdot 2x$$

Dikkat: **dış fonksiyonun türevi**, hâlâ **değiştirilmemiş iç fonksiyona** uygulanıyor ($\cos(x^2)$, $\cos(x)$ değil), sonra **iç fonksiyonun türeviyle** ($2x$) çarpılıyor. sin ve x^2'ye özgü bir şey yok; herhangi iki g, h için:

$$\frac{d}{dx}g(h(x)) = g'(h(x)) \cdot h'(x)$$

Buna **zincir kuralı** denir. Leibniz gösterimiyle:

$$\frac{dg}{dx} = \frac{dg}{dh} \cdot \frac{dh}{dx}$$

> *"this pattern right here is what we usually call the chain rule."* — Grant, 12:49

Builder Notu — Ara Değerler Saklanır

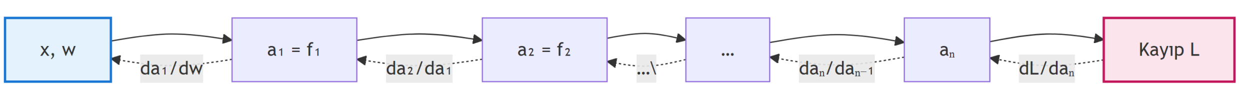
Zincir kuralı tek bir bileşke için "dış türev × iç türev"dir. Derin ağda bu, onlarca katman boyunca tekrarlanır — ve işte tam burada backprop doğar (sonraki bölüm). dg/dh notasyonu, gradyanın "hangi ara değere göre" alındığını izlemek demektir; reverse-mode autodiff'te bu ara değerler (aktivasyonlar) ileri geçişte saklanır, geri geçişte kullanılır.

4.6 Zincir Kuralı = Backprop'un Kalbi

Grant'ın altını çizdiği incelik: $dg/dx = (dg/dh) \cdot (dh/dx)$ ifadesinde dh**'ler sadeleşir** — ve bu sadece bir notasyon hilesi değil, küçük dürtmelerin gerçekten nasıl aktarıldığının yansımasıdır. Üçüncü doğrudaki dürtme, ikinci doğrudaki dürtme üzerinden, o da birinci doğrudaki dürtme (dx) üzerinden meydana geldi.

> *"that cancellation of dh is not just a notational trick, that is a genuine reflection of what's going on with the tiny nudges."* — Grant, 14:14

Bir sinir ağı tam olarak iç içe fonksiyonlardan oluşur: girdi x, ağırlıklar w, katmanlar $a_1, a_2, \ldots, a_n$ ve en sonda kayıp L.



Şekil 4.5: Backprop: zincir kuralı, ileri akışın (girdi → kayıp) ardından kayıptan geriye doğru yerel türevlerin çarpımıyla gradyanı dağıtır.

Bir ağırlığın kaybı nasıl etkilediğini bulmak için zincir kuralını tüm katmanlar boyunca uygularsın:

$$\frac{dL}{dw} = \frac{dL}{da_n} \cdot \frac{da_n}{da_{n-1}} \cdots \frac{da_1}{dw}$$

Her çarpan, bir katmanın **yerel türevidir** (çok değişkenli hâlinde bir Jacobian). **Backpropagation**, bu çarpımı çıktıdan (L) girdiye doğru, **sağdan sola** hesaplar; ara sonuçları yeniden kullanır, bu yüzden milyonlarca parametrenin gradyanını tek bir geri geçişte verimli çıkarır. Yani Grant'ın "dh sadeleşmesi", derin öğrenmenin tüm eğitim mekanizmasının çekirdeğidir.

> Builder Notu — Reverse vs Forward Mode
>
> Neden "ileri" değil de "geri"? Çarpımı soldan (dL/da_n'den) başlatmak, her adımda bir vektör-Jacobian çarpımı yapmana izin verir — bu, tam Jacobian matrislerini kurmaktan çok daha ucuzdur (reverse-mode autodiff). Skaler bir kayıptan milyonlarca parametreye gradyan akıtmak istediğinde geri mod kazanır; tersi durumda (tek girdiden çok çıktıya) ileri mod. Modern derin öğrenme çerçeveleri ikisini de yapar ama eğitim neredeyse hep reverse-mode'dur.

4.7 Katmanları Soymak: Bileşik İfadeler

Üç kuralı (toplam, çarpım, zincir) elinde tutunca, ne kadar canavarsı olursa olsun her ifadeyi adım adım soyabilirsin. Örneğin $f(x) = \sin(x^2) \cdot \cos(x)$. En dıştaki yapı bir **çarpım** ($\sin(x^2)$ çarpı $\cos x$), o yüzden önce çarpım kuralı:

$$f' = \sin(x^2) \cdot \frac{d(\cos x)}{dx} + \cos x \cdot \frac{d(\sin(x^2))}{dx}$$

Şimdi parçaları doldur: $d(\cos x)/dx = -\sin x$ (basit kural), ve $d(\sin(x^2))/dx$ **zincir kuralıyla** $\cos(x^2) \cdot 2x$. Birleştirince:

$$\frac{d}{dx}\left(\sin(x^2) \cdot \cos x\right) = \sin(x^2) \cdot (-\sin x) + \cos x \cdot \cos(x^2) \cdot 2x$$

İşin yöntemi bu: en dış birleştirmeyi tanı, ilgili kuralı uygula, ortaya çıkan alt-türevleri aynı yöntemle çöz.

> Builder Notu — Computational Graph Sırası
>
> "En dıştan başla, içe doğru soy" stratejisi, reverse-mode autodiff'in computational graph'ı geriye gezme sırasının ta kendisidir. Bir framework $f(x) = \sin(x^2) \cdot \cos(x)$'i bir grafa çevirir (çarpım düğümü, içinde $\sin \circ$ kare ve cos dalları), ileri geçişte değerleri saklar, geri geçişte her düğümde yerel türevi uygular ve zincirler. Sen elle yaptığında yorulursun; framework bunu hatasız ve milyonlarca parametre için yapar.

4.8 Bu Dersin Özeti

1. Fonksiyonları birleştirmenin üç temel yolu: toplam, çarpım, bileşke. (Çıkarma ve bölme bunlardan türer.)
2. **Toplam kuralı:** $d(g+h) = g' + h'$. Yükseklikleri istiflersin, değişimler toplanır.

3. **Çarpım kuralı:** $d(g \cdot h) = g \cdot h' + h \cdot g'$. Ayarlanabilir kutunun iki ince dikdörtgeni; "sol d sağ + sağ d sol".
4. **Zincir kuralı:** $d(g(h(x))) = g'(h(x)) \cdot h'(x)$. Dış fonksiyonun türevi (değiştirilmemiş iç fonksiyonda) çarpı iç fonksiyonun türevi.
5. Leibniz gösteriminde $dg/dx = (dg/dh) \cdot (dh/dx)$ — dh'ler sadeleşir; bu, dürtmelerin gerçek aktarımının yansımasıdır.
6. Zincir kuralı tüm katmanlar boyunca uygulanınca **backprop** olur: gradyan, kayıptan girdiye doğru geriye akar.
7. Karmaşık bir ifadeyi, en dış birleştirmeden başlayıp katman katman soyarak türetirsin; akıcılık pratik gerektirir.

! Tek bir cümle

Toplam, çarpım ve zincir kuralları küçük dürtmelerden doğal olarak çıkar; bunların en derini olan zincir kuralı (dış türev × iç türev), iç içe fonksiyonlardan oluşan bir sinir ağında gradyanı geriye akıtan **backprop**'un ta kendisidir.

4.9 Kontrol Soruları

i Soru 1: d/dx [x^3 + cos x] nedir? (Toplam kuralı)

Cevap: Toplam kuralıyla terimleri ayrı ayrı türevle: $d(x^3)/dx = 3x^2$, $d(\cos x)/dx = -\sin x$. Toplam: $3x^2 - \sin x$. Toplam kuralı her terimi bağımsız ele almanı sağlar.

i Soru 2: d/dx [x^2 · sin x] nedir? (Çarpım kuralı)

Cevap: "Sol d sağ + sağ d sol": $x^2 \cdot (\sin x)' + \sin x \cdot (x^2)' = x^2 \cdot \cos x + \sin x \cdot 2x = x^2 \cos x + 2x \sin x$. Ayarlanabilir kutuda alttaki ve sağdaki ince dikdörtgenlerin alanları.

i Soru 3: d/dx [cos(x^3)] nedir? (Zincir kuralı)

Cevap: Dış fonksiyon cos (türevi $-\sin$), iç fonksiyon x^3 (türevi $3x^2$). Zincir kuralı: dış türev (değiştirilmemiş içte) × iç türev = $-\sin(x^3) \cdot 3x^2$. Yaygın hata, iç türevle ($3x^2$) çarpmayı unutmaktır.

i Soru 4: (Builder) Basit bir ağ: L = ${a_2}^2$, a_2 = w·a_1, a_1 = x. dL/dw'yi zincir kuralıyla bul ve backprop'la ilişkilendir.

Cevap: Zincir kuralı: $dL/dw = (dL/da_2) \cdot (da_2/dw)$. $dL/da_2 = 2a_2$; $da_2/dw = a_1 = x$. Yani $dL/dw = 2a_2 \cdot x = 2(wx) \cdot x = 2wx^2$. Backprop bu çarpımı **sağdan sola** hesaplar: önce $dL/da_2 = 2a_2$ (ileri geçişte saklanan a_2 ile), sonra yerel türev x ile çarpar. Gerçek ağlarda her katman böyle bir yerel türev ekler ve zincir uzar.

4.10 Egzersizler

Egzersiz 1. $d/dx[3x^2 + 5\sin x - 1/x]$ türevini bul (toplam + sabit katsayı + $1/x$ kuralı birlikte).

Egzersiz 2. $d/dx[x^3 \cdot \cos x]$ türevini çarpım kuralıyla bul; “sol d sağ + sağ d sol” tekerlemesini uygula.

Egzersiz 3. $d/dx[\sin(\cos x)]$ türevini zincir kuralıyla bul. (İki katlı bileşke: dış sin, iç cos.)

Egzersiz 4. *(Python — sembolik doğrulama)* $\sin(x^2) \cdot \cos(x)$ türevini SymPy ile al; elle bulduğun (bölüm 6’daki) sonuçla karşılaştır.

```
sin(x^2)*cos(x) türevi: 2*x*cos(x)*cos(x**2) - sin(x)*sin(x**2)
sin(cos x) türevi: -sin(x)*cos(cos(x))
```

Egzersiz 5. *(Sonraki dersin habercisi)* e^x fonksiyonu çok özeldir: türevi kendisidir, $d(e^x)/dx = e^x$. Bunu zincir kuralıyla birleştir: $d(e^{3x})/dx$ nedir? Ya da $d(e^{x^2})/dx$? Ders 5, e sayısının neden bu kadar özel olduğunu anlatacak.

4.11 Anahtar Kavramlar (Cheat Sheet)

Kavram	Tanım	Grant’ta
Toplam kuralı	$d(g+h) = g' + h'$; değişimler istiflenir	1m49
Çarpım kuralı	$d(g \cdot h) = g \cdot h' + h \cdot g'$ (“sol d sağ + sağ d sol”)	4m13
Ayarlanabilir kutu	Çarpım = alan; iki ince dikdörtgen (köşe atılır)	4m54
Sabit katsayı	$d(c \cdot f) = c \cdot f'$; sabit dışarı çıkar	8m21
Zincir kuralı	$d(g(h(x))) = g'(h(x)) \cdot h'(x)$; dış türev × iç türev	12m49
Üç sayı doğrusu	Bileşkeyi katmanlı görselleştirme ($x \to h \to g$)	9m13
dh **sadeleşmesi**	$dg/dx = (dg/dh)(dh/dx)$; dürtmenin gerçek aktarımı	14m14
Katmanları soymak	Karmaşık ifade = üç kuralın istiflenmesi	1m28

4.12 ML Bağlantıları Özeti

7 köprü

1. **Zincir kuralı → backprop:** gradyan kayıptan girdiye geriye akar; reverse-mode autodiff bu çarpımı sağdan sola yapar. Bu dersin en kritik bağlantısı.
2. **Toplam kuralı** → multi-task loss gradyanı toplanır ($\nabla \sum L_i = \sum \nabla L_i$); residual $d(x + f(x)) = 1 + f'$ → gradient highway (ResNet).
3. **Çarpım kuralı** → kapı mekanizmaları (LSTM/GRU), attention'ın ağırlık × değer toplamı, element-wise çarpım gradyanları.
4. **Sabit katsayı** → gradyan ölçekleme; learning rate, gradyanı sabitle çarpar.
5. **Üç kural = tam autodiff kümesi** → bir computational graph (toplam, çarpım, primitif düğümler) her modeli temsil eder; başka kurala gerek yok.
6. dh **sadeleşmesi** → ara aktivasyonlar ileri geçişte saklanır, geri geçişte yerel türevlerle çarpılır.
7. **Reverse vs forward mode** → tek skaler kayıptan milyonlarca parametreye gradyan: reverse mode (backprop) kazanır; tersi durumda forward mode.

! Tek bir şey alıp gideceksen

Karmaşık türevler ezber değil — üç doğal kuralla (topla, çarp, bileşke) her ifadenin katmanlarını soyarsın. Bunların en derini zincir kuralıdır: **dış türev çarpı iç türev**. Onu üst üste binmiş katmanlara uyguladığında elde ettiğin şey **backprop**'tur — derin öğrenmenin tüm öğrenme motoru.

5 Euler Sayısı e'nin Özelliği

Türevi kendisine eşit olan yegâne taban

ℹ Bölüm bilgisi

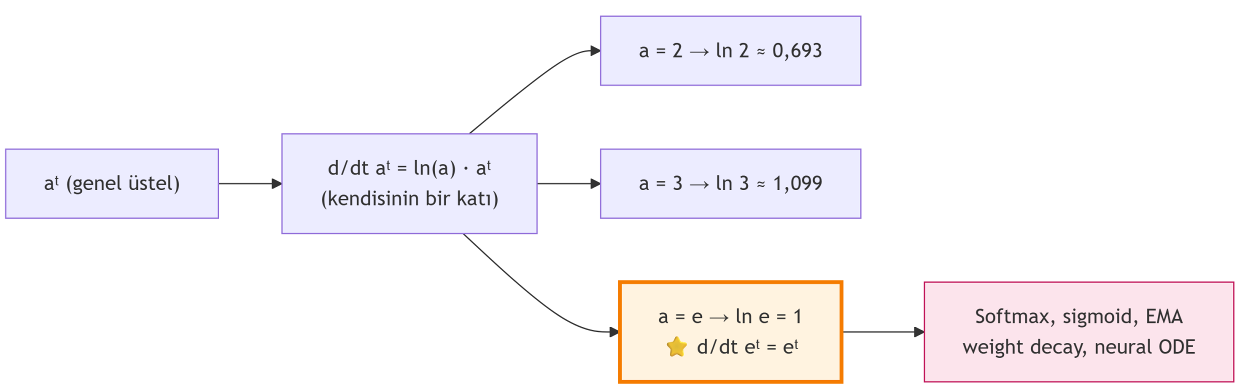
- **Grant'ın videosu:** YouTube — Chapter 5: What's so special about Euler's number e? (≈13 dk)
- **Kaynak:** 3Blue1Brown — Essence of Calculus
- **Okuma süresi:** ≈20 dk

5.1 Bu Derste Ne Var?

Türev formülleri arasında önemli birini atlamıştık: **üsteller**. Bu derste 2^x, 7^x gibi üstellerin türevini ve özellikle e^x'in neden en önemli üstel olduğunu görüyoruz. Kısa cevap: e^x'in türevi **kendisine eşittir** — ve bu, e sayısını matematiğin (ve ML'in) her yerine yerleştirir.

Üç ana fikir:

1. **Bir üstelin türevi, kendisinin bir katıdır.** Üstel özellik $a^{t+dt} = a^t \cdot a^{dt}$ bunu sağlar.
2. **O kat =** $\ln(\textbf{taban})$ (doğal logaritma). 2^x için $\approx 0{,}6931 = \ln 2$; 3^x için $\ln 3$.
3. e, o katın tam **1** olduğu özel tabandır ($e \approx 2{,}71828$); yani $d(e^t)/dt = e^t$.



Şekil 5.1: Üstellerin türev örüntüsü: tüm a^t'lerin türevi $\ln(a) \cdot a^t$. e, $\ln(a) = 1$ olan özel taban.

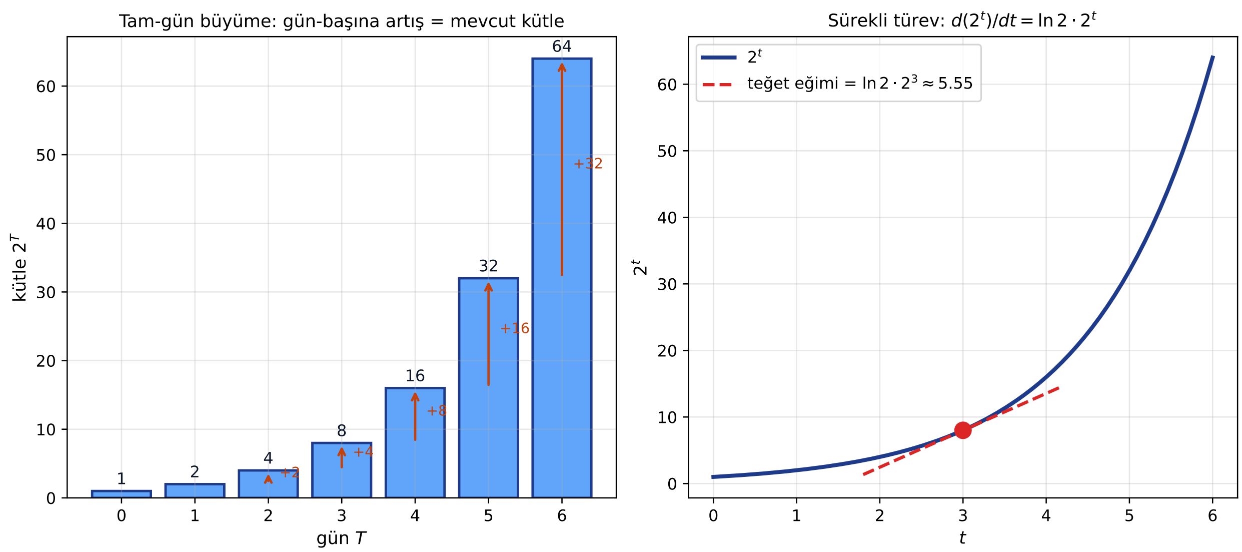
> *"e to the t actually equals its own derivative … the slope of a tangent line to any point on this graph equals the height of that point."* — Grant, 8:36

Builder Notu — ML Köprüleri

- e^x**'in türevi kendisi** → **softmax** (e^{logit}), **sigmoid** ($1/(1+e^{-x})$), **exponential decay** (learning rate schedule, EMA), gradient flow — hepsi bu özelliğe dayanır.
- ln(**taban**) **sabiti** → log-uzay hesapları, **log-likelihood**, **cross-entropy**, logit; çarpımları toplama çeviren köprü.
- **"Oran** $\propto$ **miktar"** → exponential decay her yerde: **weight decay**, lr decay, EMA momentum (β), RL'de **discount** γ^t; sürekli dinamik (neural ODE).
- $e^{c \cdot t}$**,** c **= oran sabiti** → öğrenme oranı çizelgesi $e^{-\lambda t}$, Adam'da β_1/β_2 üstel ortalamaları; c, "ne kadar hızlı büyür/söner" anlamını taşır.

5.2 2^t: Bir Popülasyon, Her Gün İkiye Katlanıyor

Sezgi için 2^t fonksiyonuna odaklanalım. Girdiyi bir **zaman** T (gün cinsinden), çıktıyı (2^t) bir **popülasyon büyüklüğü** olarak düşün — her gün ikiye katlanan, verimli bir yaratık topluluğu.



Şekil 5.2: 2^t: her gün ikiye katlanan popülasyon. Gün-başına büyüme = o günkü kütleye eşit (kabaca; tam türev için bölümün sonuna bak).

$T = 0$'da kütle $2^0 = 1$; $T = 1$'de $2^1 = 2$; ve her gün ikiye katlanarak devam eder. Türev için, kütlenin büyüme oranı dm/dt'yi istiyoruz.

Önce **tam bir günlük** değişime bakalım. 3. ile 4. gün arasında kütle 8'den 16'ya çıkar — günde 8 birim artış, ki bu **gün başındaki popülasyon büyüklüğüne eşit**. Genel olarak tam-günlük büyüme oranı, o günün başındaki popülasyona eşittir.

Bu, "2^t'nin türevi kendisidir" demeye yol açabilir — doğru yönde ama tam doğru değil. Çünkü burada **tam bir gün** üzerinden karşılaştırma yapıyoruz; türev ise giderek **küçülen** değişimleri sorar. Sonraki bölümde bu farkı düzelteceğiz.

Builder Notu — EMA ve Üstel Sönüm

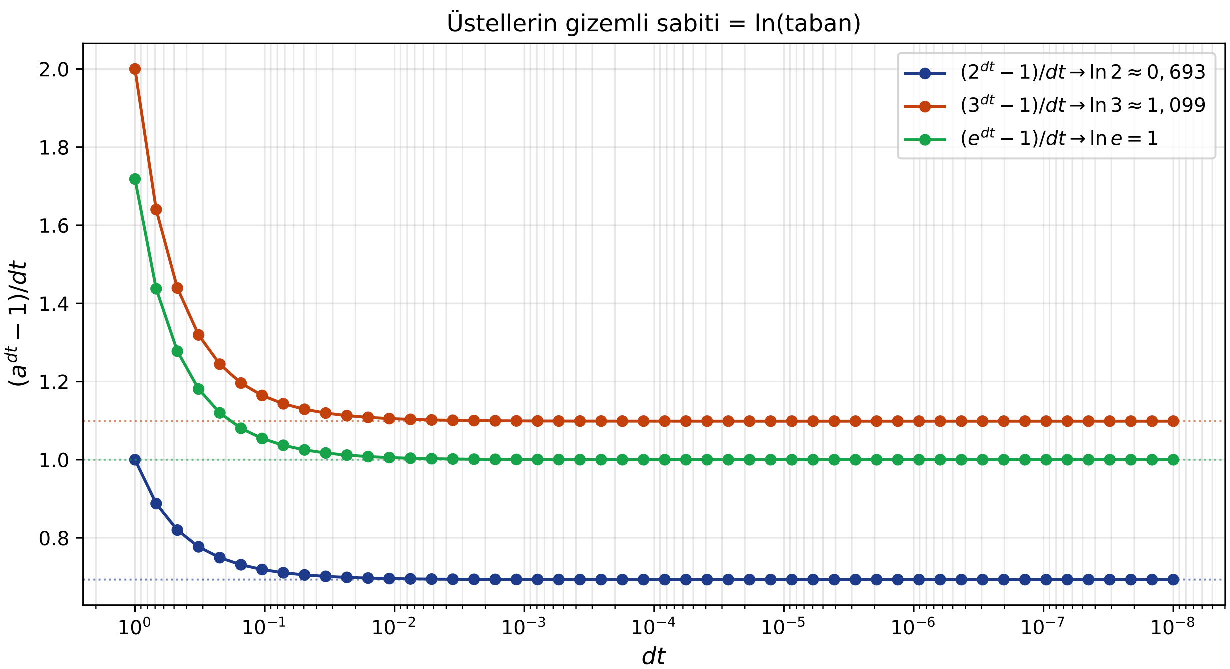
"Değişim oranı, mevcut büyüklüğe eşit/orantılı" — bu, tüm **üstel büyüme/sönüm** olgularının imzasıdır. ML'de bunu en çok **exponential moving average** (EMA) ve **üstel öğrenme oranı sönümü**nde görürsün: bir büyüklük, kendi değerinin sabit bir oranı kadar güncellenir, bu da üstel bir eğri çizer.

5.3 Türev: 2^t Kendisinin Bir Katı (Üstel Özellik)

Tam-gün yerine küçük dt için soralım: $[2^{t+dt} - 2^t]/dt$ nedir? Burada üstellerin **en önemli özelliği** devreye girer: üstte toplama varsa, çıktıyı bir çarpıma ayırabilirsin:

$$\frac{2^{t+dt} - 2^t}{dt} = 2^t \cdot \frac{2^{dt} - 1}{dt}$$

2^t'yi dışarı çarpan olarak aldık. Şimdi kritik gözlem: sağdaki $(2^{dt} - 1)/dt$ terimi, **başladığımız t'den tamamen bağımsız** — içinde yalnızca dt var. Hesap makinesine küçük dt değerleri koyarsan, dt küçüldükçe bu ifade çok belirli bir sayıya yaklaşır: yaklaşık $0{,}6931$.



Şekil 5.3: $(a^{dt} - 1)/dt$, $dt \to 0$ iken $\ln(a)$'ya yakınsar. Üç farklı taban için gözle görülür yakınsama.

Yani diğer fonksiyonların türevlerinin aksine, dt'ye bağlı her şey t'den ayrılıyor. Sonuç: 2^t'nin türevi kendisidir, ama bir **sabitle** çarpılmış:

$$\frac{d}{dt} 2^t = (0{,}6931\ldots)\, 2^t$$

> **Builder Notu — Log-uzay**
>
> Toplamayı (üstteki $t + dt$) çarpmaya ($2^t \cdot 2^{dt}$) çeviren bu özellik, ML'in **log-uzayda** çalışmasının nedenidir: olasılık çarpımları taşma yapar, ama log alınca toplama olur ve stabil kalır. Tersi yönde, e^{toplam} = çarpım özdeşliği softmax'ın paydasını ve enerji-temelli modelleri tanımlar. "Toplamsal ↔ çarpımsal" köprüsü, üstel/logaritmik fonksiyonların ML'deki her yerdeliğinin kökü.

5.4 Gizemli Sabitler: $\ln(\textbf{taban})$

2'de özel bir şey yok. 3^t ile çalışsaydık, üstel özellik yine "türev kendisiyle orantılı" sonucunu verirdi, ama bu kez sabit $\approx 1{,}0986$ olurdu. 8^t için sabit $\approx 2{,}079$ çıkar — ve dikkat: bu, 2 tabanının sabitinin tam **üç katıdır** ($3 \cdot 0{,}6931 = 2{,}079$). Tesadüf değil: $8 = 2^3$ olduğundan sabitler de bu ilişkiyi taşır.

Örüntü şu: oran sabiti, tabanın **doğal logaritmasıdır** (ln):

$$\frac{d}{dt} a^t = \ln(a) \cdot a^t$$

$$\ln 2 \approx 0{,}6931, \qquad \ln 3 \approx 1{,}0986, \qquad \ln 8 = 3 \ln 2 \approx 2{,}079$$

> **Builder Notu — Cross-entropy**
>
> $\ln(\text{taban})$ sabiti, ML'de **logit** ve **log-likelihood**'un temelidir. Bir softmax çıktısının logu, $\ln(\text{taban})$ ilişkileriyle doludur; cross-entropy kaybı $= -\sum \log p$, doğal logaritma üzerine kuruludur. Üstel bir büyümenin "hızını" (ln taban) okumak, log-ölçekli grafiklerde eğimi okumakla aynı şeydir.

5.5 e: Sabitin Tam 1 Olduğu Taban

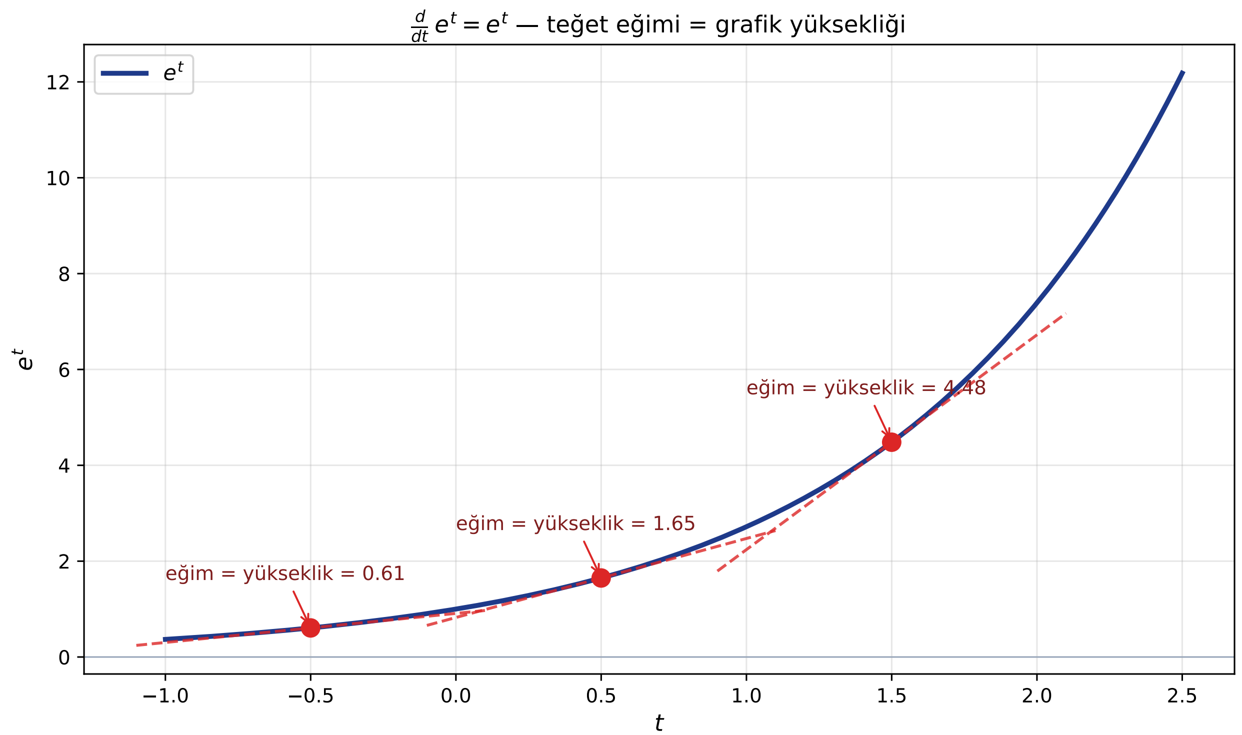
Doğal soru: öyle bir taban var mı ki o orantı sabiti tam **1** olsun — yani üstelin türevi yalnızca kendisine *orantılı* değil, kendisine **eşit** olsun? Var: özel sabit $e \approx 2{,}71828$.

Aslında e burada "tesadüfen ortaya çıkmaz"; bu özellik e'yi **tanımlayan** şeydir. "Neden bunca sayı içinde e?" diye sormak, "neden bunca sayı içinde π, çemberin çevresinin çapına oranı?" diye sormak gibidir.

Tüm üsteller kendi türevleriyle orantılıdır; ama yalnızca e'de o sabit 1'dir:

$$\frac{d}{dt} e^t = e^t \qquad (e \approx 2{,}71828)$$

> **Builder Notu — Neural ODE**
>
> "Türevi kendine eşit" özelliği, e^t'yi diferansiyel denklemlerin doğal çözümü yapar — ve ML'de **sürekli dinamiklerin** dili budur. Bir neural ODE'de gizli durumun zaman evrimi $dy/dt = f(y)$ biçimindedir; en basit hâli ($f(y) = c \cdot y$) doğrudan üstel çözüm verir.



Şekil 5.4: e^t'in her noktasındaki teğet eğimi = o noktanın yüksekliğine eşit. Bu özellik e'yi diğer tüm tabanlardan ayırır.

5.6 e^{ct} Bir Seçimdir: c'nin Anlamı

Zincir kuralıyla (Ders 4) artık her şeyi çözebiliriz. $d(e^{3t})/dt$ nedir? Dış fonksiyon e^x (türevi kendisi), iç fonksiyon $3t$ (türevi 3). Zincir kuralı:

$$\frac{d}{dt} e^{ct} = c\, e^{ct}$$

Şimdi gizemli sabitler tamamen çözülüyor. 2 sayısını $e^{\ln 2}$ olarak yazabiliriz. O hâlde $2^t = e^{(\ln 2)\cdot t}$, ve zincir kuralı türevin sabitini doğrudan ln 2 yapar:

$$2^t = e^{(\ln 2)\, t} \qquad \frac{d}{dt} 2^t = \ln 2 \cdot 2^t$$

İşte $0{,}6931 = \ln 2$ buradan geliyor; aynısı her taban için geçerli (sabit = ln taban). Bu yüzden calculus uygulamalarında üsteller neredeyse hiç "taban üzeri t" yazılmaz; $e^{\textbf{sabit}\cdot t}$ yazılır.

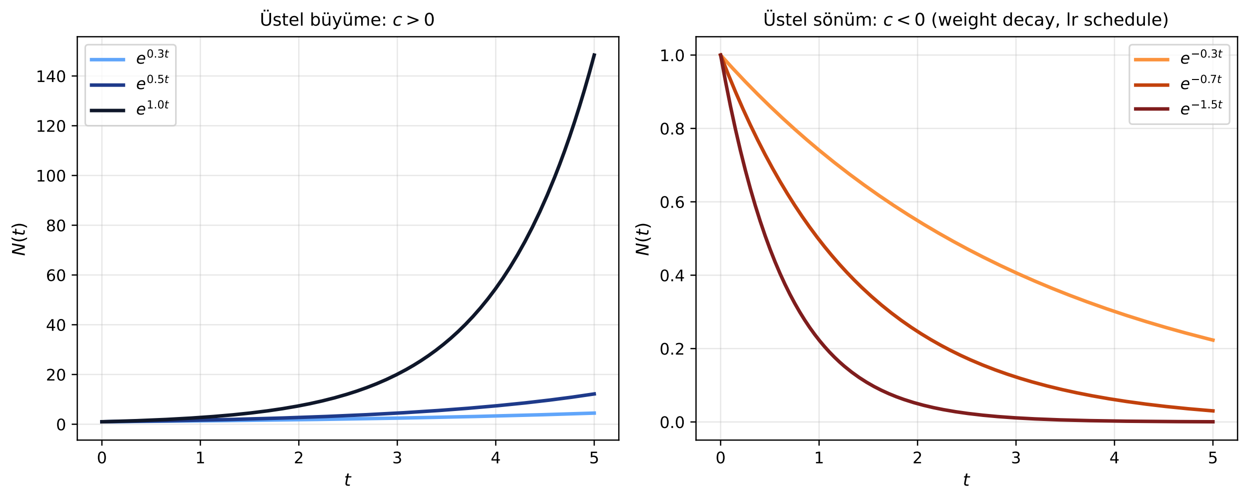
Builder Notu — Hep Aynı Kalıp

e^{ct} yazımı ML'de standarttır çünkü c, "ne kadar hızlı" sorusunun yanıtıdır. Öğrenme oranı sönümü genelde $\mathrm{lr}(t) = \mathrm{lr}_0 \cdot e^{-\lambda t}$ biçimindedir; λ, sönüm hızıdır. EMA güncellemesi (momentum β) ve Adam'ın β_1/β_2'si üstel ortalamalardır. RL'de discount $\gamma^t = e^{(\ln \gamma)\cdot t}$. Hep aynı kalıp: e^{ct}, c = oran.

5.7 Doğanın Üstelleri: Oran $\propto$ Miktar

Neden bu kadar çok şey üstel? Çünkü doğada pek çok olguda, bir niceliğin **değişim oranı, o niceliğin kendisiyle orantılıdır**:

- **Popülasyon:** büyüme oranı, mevcut nüfusla orantılı (kaynak sınırı yoksa).
- **Newton soğuması:** sıcak su soğur, soğuma oranı su ile oda arasındaki sıcaklık **farkıyla** orantılı.
- **Bileşik faiz:** paranın büyüme oranı, o anki para miktarıyla orantılı.



Şekil 5.5: Üstel büyüme ve sönüm: c pozitifse büyür, negatifse söner. $|c|$ büyüklük "hızı" verir.

Bu durumların hepsinde, niceliği zamana karşı tanımlayan fonksiyon bir **üsteldir**. Matematiksel olarak "oran kendisiyle orantılı" demek:

$$\frac{dN}{dt} = c\,N \qquad N(t) = N_0\,e^{ct}$$

Bu fonksiyonu yazmanın çok yolu olsa da, $e^{c \cdot t}$ seçmek doğaldır — çünkü c, tam olarak değişen değişkenin büyüklüğü ile değişim oranı arasındaki **orantı sabitidir**.

> *"it's the same as the proportionality constant between the size of the changing variable and the rate of change."* — Grant, 13:05

Builder Notu — Gradient Flow

$dN/dt = c \cdot N$ denklemi, ML'deki sürekli-zaman düşüncesinin iskeletidir. **Weight decay** bir sönüm denklemidir ($dw/dt = -\lambda w \to w(t) = w_0 e^{-\lambda t}$). **EMA**: yeni ortalama, eskiyle üstel ağırlıklı karışır. Gradient flow, SGD'nin sürekli limiti olarak $d\theta/dt = -\nabla L$ biçiminde yazılır; lineerleştirildiğinde özdeğerler $e^{\lambda t}$ terimleri verir ve eğitimin kararlılığını (patlama/sönme) belirler.

5.8 Bu Dersin Özeti

1. Üstellerin türevi, kendilerinin bir katıdır; bunu üstel özellik $a^{t+dt} = a^t \cdot a^{dt}$ sağlar.
2. O kat, başlangıç zamanı t'den bağımsız bir sabittir: $(a^{dt} - 1)/dt$, $dt \to 0$ iken ln(a)'ya yaklaşır.
3. $d(a^t)/dt = \ln(a) \cdot a^t$. Sayısal: $\ln 2 \approx 0{,}6931$, $\ln 3 \approx 1{,}0986$, $\ln 8 = 3 \cdot \ln 2 \approx 2{,}079$.
4. $e \approx 2{,}71828$, o sabitin tam 1 olduğu özel tabandır: $d(e^t)/dt = e^t$. Bu eşitlik e'yi **tanımlar**.
5. Zincir kuralıyla $d(e^{ct})/dt = c \cdot e^{ct}$; ve $2^t = e^{(\ln 2) \cdot t}$ yazımı gizemli sabitleri tamamen açıklar.
6. $e^{c \cdot t}$ yazmak bir **seçimdir** (e fonksiyona özsel değil); değerini, c'ye "oranın miktara oranı" anlamını vermesi belirler.
7. "Değişim oranı miktarla orantılı" ($dN/dt = c \cdot N$) olan her olgu, üstel bir çözüme ($N_0 \cdot e^{ct}$) sahiptir.

! Tek bir cümle

e^x, türevi tam kendisine eşit olan yegâne üstel tabandır (bu, e'nin tanımıdır); ve "değişim oranı, değişen miktarla orantılı" olan her olgu — popülasyon, soğuma, faiz, sönüm — doğal olarak $e^{c \cdot t}$ biçiminde yazılır, çünkü buradaki c o orantı sabitidir.

5.9 Kontrol Soruları

i Soru 1: d/dt [e^{5t}] nedir?

Cevap: Zincir kuralı: dış fonksiyon e^x (türevi kendisi), iç fonksiyon $5t$ (türevi 5). Sonuç: $e^{5t} \cdot 5 = 5e^{5t}$. Genel olarak $d(e^{ct})/dt = c \cdot e^{ct}$.

i Soru 2: d(a^t)/dt'deki orantı sabiti neden başlangıç zamanı t'den bağımsızdır?

Cevap: Üstel özellik sayesinde: $a^{t+dt} = a^t \cdot a^{dt}$. Fark oranında a^t dışarı çarpan olarak çıkar ve geriye $(a^{dt} - 1)/dt$ kalır — bu ifade yalnızca dt'ye bağlıdır, t hiç görünmez. Bu yüzden türev "kendisi çarpı bir sabit"tir; sabit = ln(a).

i Soru 3: d/dt [e^{-t}] nedir ve ne anlama gelir?

Cevap: $c = -1$ için $d(e^{ct})/dt = c \cdot e^{ct} = -e^{-t}$. Türev daima değerin negatifi: fonksiyon her zaman azalır, üstelik o anki değeriyle orantılı hızda. Bu **üstel sönümdür** — radyoaktif bozunma, weight decay, lr sönümü hep bu biçimde.

i Soru 4: (Builder) Weight decay'de $w(t) = w_0 \cdot e^{-\lambda t}$. λ büyürse ne olur? Yarı-ömür nedir?

Cevap: λ büyüdükçe sönüm hızlanır — ağırlık daha çabuk küçülür. Yarı-ömür, değerin yarıya indiği süredir: $e^{-\lambda t} = 1/2 \to t = \ln 2/\lambda$. Yani λ iki katına çıkarsa yarı-ömür yarıya iner. λ, "regularizasyonun ne kadar agresif olduğunu" doğrudan kontrol eder; EMA momentum β ve Adam'ın β_1/β_2'si de aynı yarı-ömür mantığıyla okunur.

5.10 Egzersizler

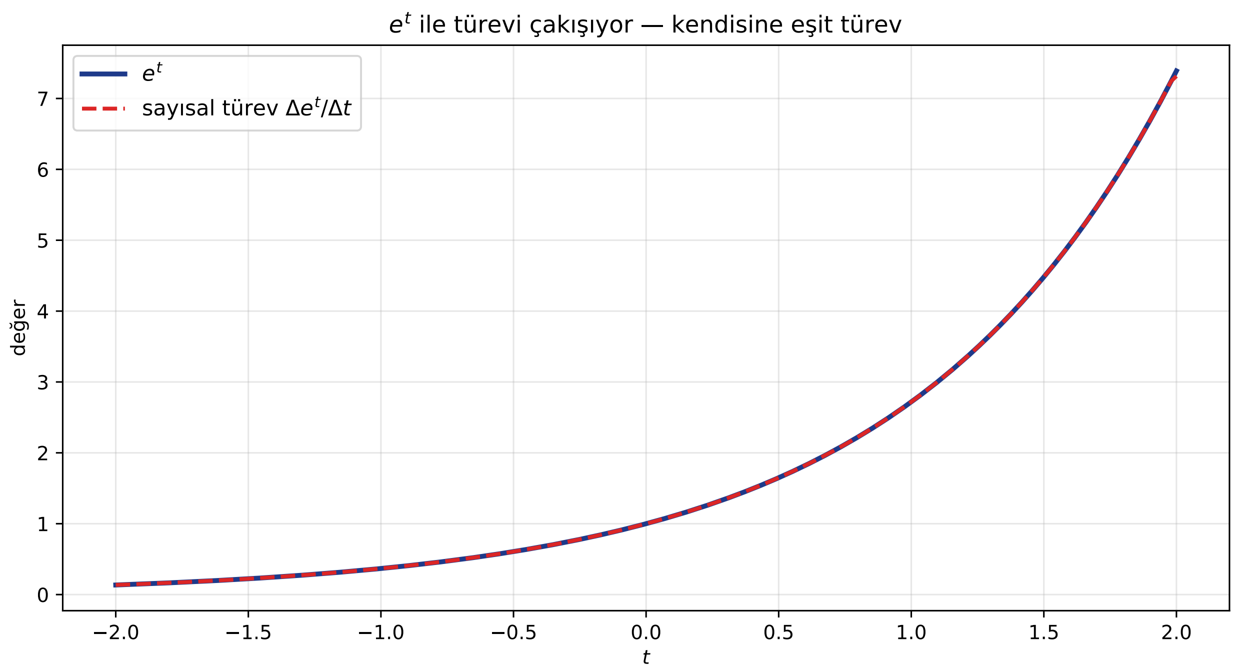
Egzersiz 1. Şu türevleri bul: $d/dt[e^{-2t}]$, $d/dx[e^{x^2}]$ (zincir kuralı), $d/dt[3 \cdot e^t]$.

Egzersiz 2. 5^t'nin türev sabitinin $\ln 5$ olduğunu göster: $5 = e^{\ln 5}$ yazıp zincir kuralını uygula. $\ln 5$'i hesapla ($\approx 1{,}609$) ve $(5^{0{,}001} - 1)/0{,}001$ ile karşılaştır.

Egzersiz 3. *(Newton soğuması)* Soğuma denklemi $dT/dt = -k(T - T_{\text{oda}})$. Çözümün $T(t) = T_{\text{oda}} + (T_0 - T_{\text{oda}}) \cdot e^{-kt}$ olduğunu, bu ifadeyi denkleme yerine koyarak doğrula.

Egzersiz 4. *(Python — sayısal doğrulama)* $(2^{dt} - 1)/dt$'nin $dt \to 0$ iken $\ln 2$'ye yaklaştığını göster; ayrıca e^t ile sayısal türevinin çakıştığını çiz.

```
dt=1e-01  (2^dt - 1)/dt = 0.717735
dt=1e-02  (2^dt - 1)/dt = 0.695555
dt=1e-03  (2^dt - 1)/dt = 0.693387
dt=1e-05  (2^dt - 1)/dt = 0.693150
ln 2 = 0.6931471805599453
```



Şekil 5.6: e^t ile sayısal türevi ($\Delta e^t/\Delta t$) tam çakışır — bir fonksiyonun kendisine eşit türevi olmasının görsel kanıtı.

Egzersiz 5. *(Sonraki dersin habercisi)* Şimdiye kadar hep $y = f(x)$ biçiminde **açık** fonksiyonların türevini aldık. Peki $x^2 + y^2 = 25$ gibi, y'yi x cinsinden açıkça çözmediğin **kapalı** bir ilişkide dy/dx nasıl bulunur? Ders 6, **kapalı türevi** anlatacak.

5.11 Anahtar Kavramlar (Cheat Sheet)

Kavram	Tanım	Grant'ta
Üstel özellik	$a^{t+dt} = a^t \cdot a^{dt}$; toplama → çarpma	4m14
$d(a^t)/dt = \ln(a) \cdot a^t$	Türev = kendisi çarpı ln(taban)	5m53
Gizemli sabit = ln(taban)	$0{,}6931 = \ln 2$; $1{,}0986 = \ln 3$	10m45
$e \approx 2{,}71828$	Orantı sabitinin tam 1 olduğu taban	7m54
$d(e^t)/dt = e^t$	e'yi tanımlayan özellik; teğet eğimi = yükseklik	8m36
$d(e^{ct})/dt = c \cdot e^{ct}$	Zincir kuralı; c = oran sabiti	9m00
$2^t = e^{(\ln 2) \cdot t}$	Her üstel, e tabanında yazılabilir	9m58
Oran $\propto$ miktar	$dN/dt = c \cdot N \to N_0 \cdot e^{ct}$	11m55

5.12 ML Bağlantıları Özeti

7 köprü

1. $d(e^x)/dx = e^x$ → softmax (e^{logit}), sigmoid, gradient flow, neural ODE'lerin doğal çözümü.
2. **Üstel sönüm** → weight decay ($w_0 e^{-\lambda t}$), lr schedule, EMA momentum, RL discount $\gamma^t = e^{(\ln \gamma) \cdot t}$.
3. ln**(taban) sabiti** → logit, log-likelihood, cross-entropy; kayıp eğrilerini log-ölçekte okumak.
4. **Üstel özellik (toplam → çarpım)** → log-olasılık toplamı (taşma önleme), enerji-temelli modeller, softmax paydası.
5. e^{ct}**,** c **= oran** → yarı-ömür/β okuması: EMA, Adam (β_1/β_2), öğrenme oranı çizelgeleri.
6. $dN/dt = c \cdot N$ → sürekli dinamik; gradient flow lineerleştirmesinde özdeğer $e^{\lambda t}$, eğitimin patlama/sönme kararlılığı.
7. e**'nin tanımı (türevi kendisi)** → diferansiyel denklemlerin doğal çözüm tabanı; lineer sistem analizinin dili.

Tek bir şey alıp gideceksen

Üstellerin sihri tek bir özellikten gelir — türevleri kendileriyle orantılıdır, orantı sabiti de ln(taban). e, bu sabitin tam 1 olduğu sayıdır; yani e^x kendi türevine eşittir. Bu yüzden "değişimi kendi büyüklüğüne bağlı" olan her şey (büyüme, sönüm, softmax, EMA, discount) $e^{c \cdot t}$ ile yazılır ve c sana o değişimin hızını söyler.

6 Kapalı (Implicit) Türev

Çember, merdiven ve gradient'in kapısı

ℹ Bölüm bilgisi

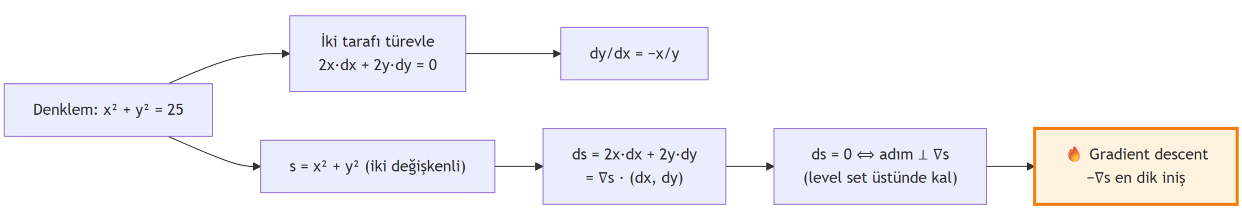
- **Grant'ın videosu:** YouTube — Chapter 6: Implicit differentiation, what's going on here? (≈15 dk)
- **Kaynak:** 3Blue1Brown — Essence of Calculus
- **Okuma süresi:** ≈23 dk

6.1 Bu Derste Ne Var?

Şimdiye kadar hep $y = f(x)$ biçiminde **açık** fonksiyonların türevini aldık. Ama $x^2 + y^2 = 25$ (bir çember) gibi, y'nin x cinsinden tek bir çıktı olmadığı **kapalı** ilişkilerde dy/dx nasıl bulunur? Cevap **kapalı türev**: denklemin iki tarafını da türevleyip dy/dx'i çözersin. Bu işlem ilk bakışta tuhaftır — ama ardındaki sezgi, **çok değişkenli** düşüncenin kapısını aralar.

Üç ana fikir:

1. **Kapalı eğri:** x ve y'nin bir denklemi sağladığı noktalar kümesi (bir fonksiyon grafiği olmak zorunda değil).
2. **Prosedür:** denklemin iki tarafını da türevle ($x^2 \to 2x \cdot dx$, $y^2 \to 2y \cdot dy$), sonra dy/dx'i çöz.
3. **Anlam:** $s = x^2 + y^2$ iki değişkenli bir fonksiyon; $ds = 2x \cdot dx + 2y \cdot dy$. Eğri üzerinde kalmak = s'yi sabit tutmak = $ds = 0$.



Şekil 6.1: Kapalı türevden gradient'e: $ds = \nabla s \cdot (dx, dy)$ iç çarpımı.

> *"this is a little sneak peek into multivariable calculus … the key, as always, is to have a clear image of what tiny nudges are at play and how they depend on each other."* — Grant, 14:37

Builder Notu — ML Köprüleri

- $ds = 2x \cdot dx + 2y \cdot dy$ **aslında** $\nabla s \cdot (dx, dy)$ — yani **gradient**. "Eğri (level set) üzerinde kal" koşulu ($ds = 0$), adımın gradient'e **dik** olması demektir. Gradient descent'in özü budur: $-\nabla$ en dik iniş yönü, level set'e diktir.
- **Kapalı türev** → **implicit layers / Deep Equilibrium Models (DEQ)**: bir denge (fixed point) noktasından türev almak; **MAML** ve hyperparameter optimization'da **implicit gradient**.
- **İlgili oranlar (related rates)** → bağlı değişkenlerin değişim oranlarının birbirine kilitlenmesi; kısıt (constraint) propagasyonu.
- $d(\ln x) = 1/x$**, ters fonksiyondan** → ters fonksiyon türevi; **normalizing flows**'ta log-det-Jacobian hesabının temeli.

6.2 Çemberin Teğeti: Bir Fonksiyon Değil

Orijin merkezli, yarıçapı 5 olan bir çember al: $x^2 + y^2 = 5^2$. Bu çembere, örneğin $(3, 4)$ noktasında **teğet doğrunun eğimini** bulmak istiyoruz.

Geometriye hakimsen, teğetin o noktadaki yarıçapa dik olduğunu bilebilirsin. Ama bunu bilmediğimizi, ya da çemberin dışındaki eğrilere de genelleşen bir teknik istediğimizi varsayalım.

Bir sorun var: bu eğri bir **fonksiyonun grafiği değil**. x bir girdi, y bir çıktı değil; ikisi de bir denklemle birbirine bağlı, karşılıklı bağımlı değerler. Buna **kapalı eğri** denir.

Builder Notu — DEQ ve Implicit

Kapalı eğriler ML'de "çözümü açıkça yazılamayan ama bir koşulla tanımlı" nesnelerin habercisidir. Bir **Deep Equilibrium Model** (DEQ), çıktısını $z = f(z, x)$ sabit-nokta denklemiyle tanımlar — z'yi açıkça çözmek yerine kapalı koşulu kullanır. Bu modellerin gradyanı, tam da bu derste öğreneceğimiz kapalı türev mantığıyla (implicit function theorem) hesaplanır.

6.3 Kapalı Türev Prosedürü: Her İki Tarafı Türevle

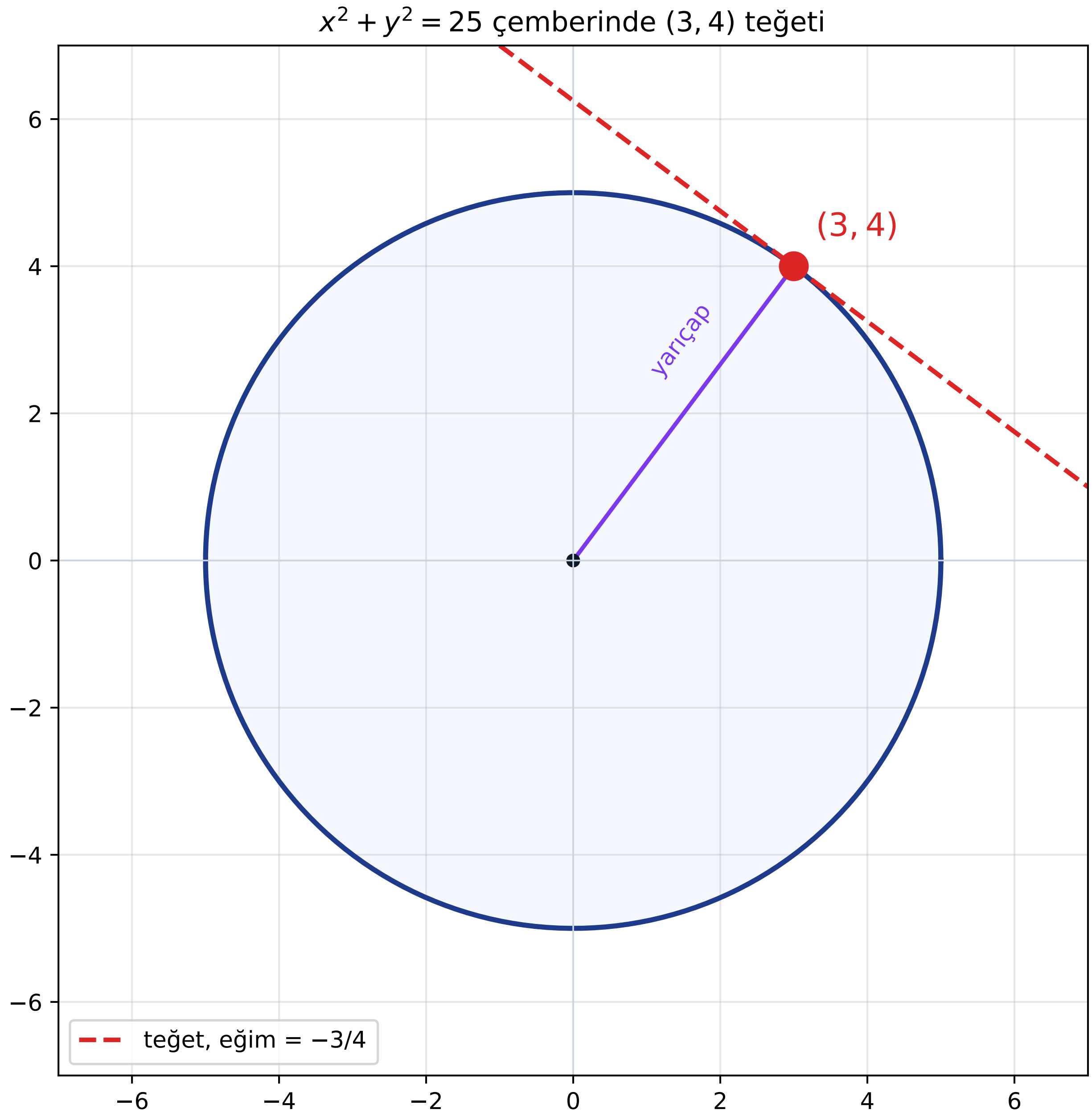
Mekanik şöyle. Denklemin **iki tarafını da** türevle. x^2 için $2x \cdot dx$ yaz, y^2 için $2y \cdot dy$ yaz, sağdaki sabit 5^2'nin türevi ise 0:

$$x^2 + y^2 = 5^2$$

$$2x\,dx + 2y\,dy = 0 \qquad \frac{dy}{dx} = -\frac{x}{y} = -\frac{3}{4}$$

$(3, 4)$ noktasında eğim $-3/4$ çıkar. Bu garip sürecin adı **kapalı türev**.

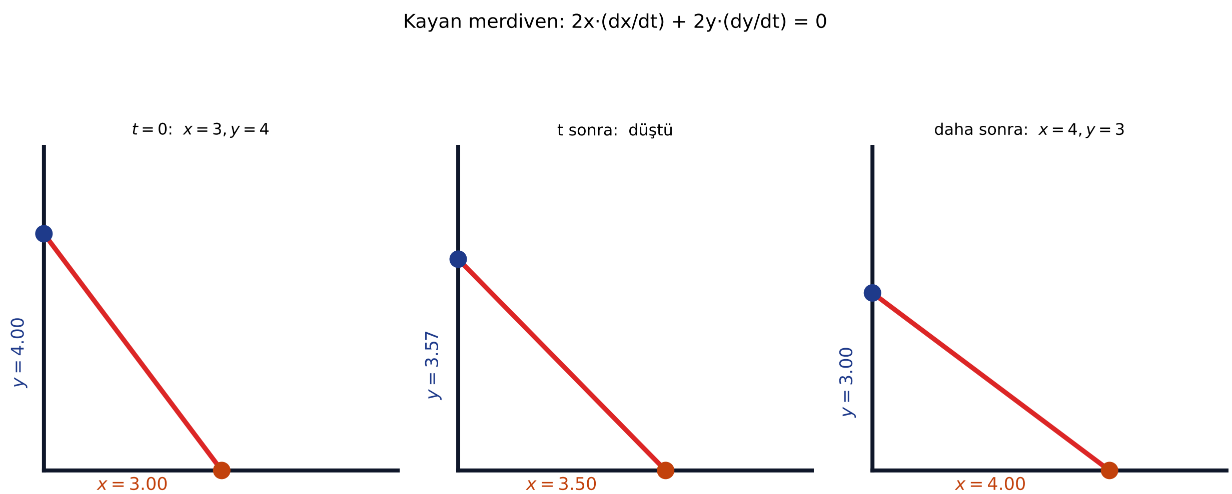
> *"this strange process is called implicit differentiation."* — Grant, 3:05



Şekil 6.2: Çember $x^2 + y^2 = 25$ ve $(3, 4)$ noktasındaki teğet. Eğim $-3/4$; yarıçapın eğimi $4/3$'ün negatif tersi.

6.4 İlgili Oranlar: Kayan Merdiven

Aynı denklem bambaşka bir problemde de karşımıza çıkar. 5 metrelik bir merdiven duvara dayalı; üst ucu yerden 4 m yukarıda başlıyor, dolayısıyla alt ucu duvardan 3 m uzakta. Merdiven kayıyor: üst uç saniyede 1 m düşüyor. Soru: o ilk anda alt uç duvardan ne hızla uzaklaşıyor?



Şekil 6.3: Kayan merdiven: $x(t)^2 + y(t)^2 = 25$ kısıtı zamanın her anında doğru. Üst 1 m/s düşerken alt $4/3$ m/s uzaklaşır.

Üst ucun yüksekliğine $y(t)$, alt ucun duvara uzaklığına $x(t)$ diyelim. Onları bağlayan denklem Pisagor, ve **her t anında doğru**:

$$x(t)^2 + y(t)^2 = 5^2$$

Zincir kuralıyla zamana göre türevini al:

$$2x\frac{dx}{dt} + 2y\frac{dy}{dt} = 0$$

$t = 0$'da $x = 3$, $y = 4$ ve $dy/dt = -1$. Yerine koy:

$$2(3)\frac{dx}{dt} + 2(4)(-1) = 0 \qquad \frac{dx}{dt} = \frac{4}{3}$$

Alt uç saniyede $4/3$ m hızla uzaklaşıyor.

> **Builder Notu — Manifold Optimizasyon**
>
> İlgili oranlar = **bağlı değişkenlerin değişim oranlarının birbirine kilitlenmesi**. ML'de bir kısıt yüzeyi üzerinde hareket ederken (örneğin normalize edilmiş ağırlıklar $\|w\| = 1$ koşulu) parametrelerin değişim oranları bağımsız değildir; biri değişince diğerleri kısıtı korumak için uyum sağlar. Bu, kısıtlı optimizasyon ve manifold üzerinde gradient (Riemannian optimization) düşüncesinin tohumu.

6.5 Kapalı Türevin Anlamı: İki Değişkenli Fonksiyon

Bu ifadeye bir isim verelim: $s = x^2 + y^2$. s aslında **iki değişkenli bir fonksiyondur** — düzlemdeki her (x, y) noktasını bir sayıyla eşler. Çember üzerindeki noktalar için bu sayı 25'tir; merkezden uzaklaşırsan daha büyük, merkeze yaklaşırsan daha küçük olur.

s'nin türevini almak demek: her iki değişkende küçük birer değişim (dx ve dy) düşünmek — illa çember üzerinde kalan bir adım değil, düzlemde **herhangi bir yönde** herhangi bir küçük adım — ve s'nin ne kadar değiştiğini sormak:

$$ds = 2x\,dx + 2y\,dy$$

İşte anahtar: adımlarını **çember boyunca** kısıtlarsan, s'nin değerini sabit (25) tutmak istiyorsun demektir, yani $ds = 0$:

$$2x\,dx + 2y\,dy = 0$$

Buradan $dy/dx = -x/y$ çıkar — ilk bölümdeki "garip" prosedür artık anlamlı: çemberde kalmak = s'yi değiştirmemek.

> *"when you restrict yourself to steps along the circle … you want to ensure that this value of s doesn't change … ds should be 0."* — Grant, 10:07

Builder Notu — Gradient'in Kalbi

Bu, çok değişkenli calculus'un ve gradient descent'in tam kalbidir. $ds = 2x \cdot dx + 2y \cdot dy$ aslında bir **iç çarpımdır**:

$$ds = \nabla s \cdot (dx, dy), \qquad \nabla s = (2x, 2y)$$

∇s **gradient** vektörüdür. "Level set (s = sabit) üzerinde kal" koşulu $ds = 0$, adımın gradient'e **dik** olması demektir. Gradient descent bunun tersini kullanır: s'yi en hızlı değiştirmek (azaltmak) için $-\nabla s$ yönünde adım atarsın.

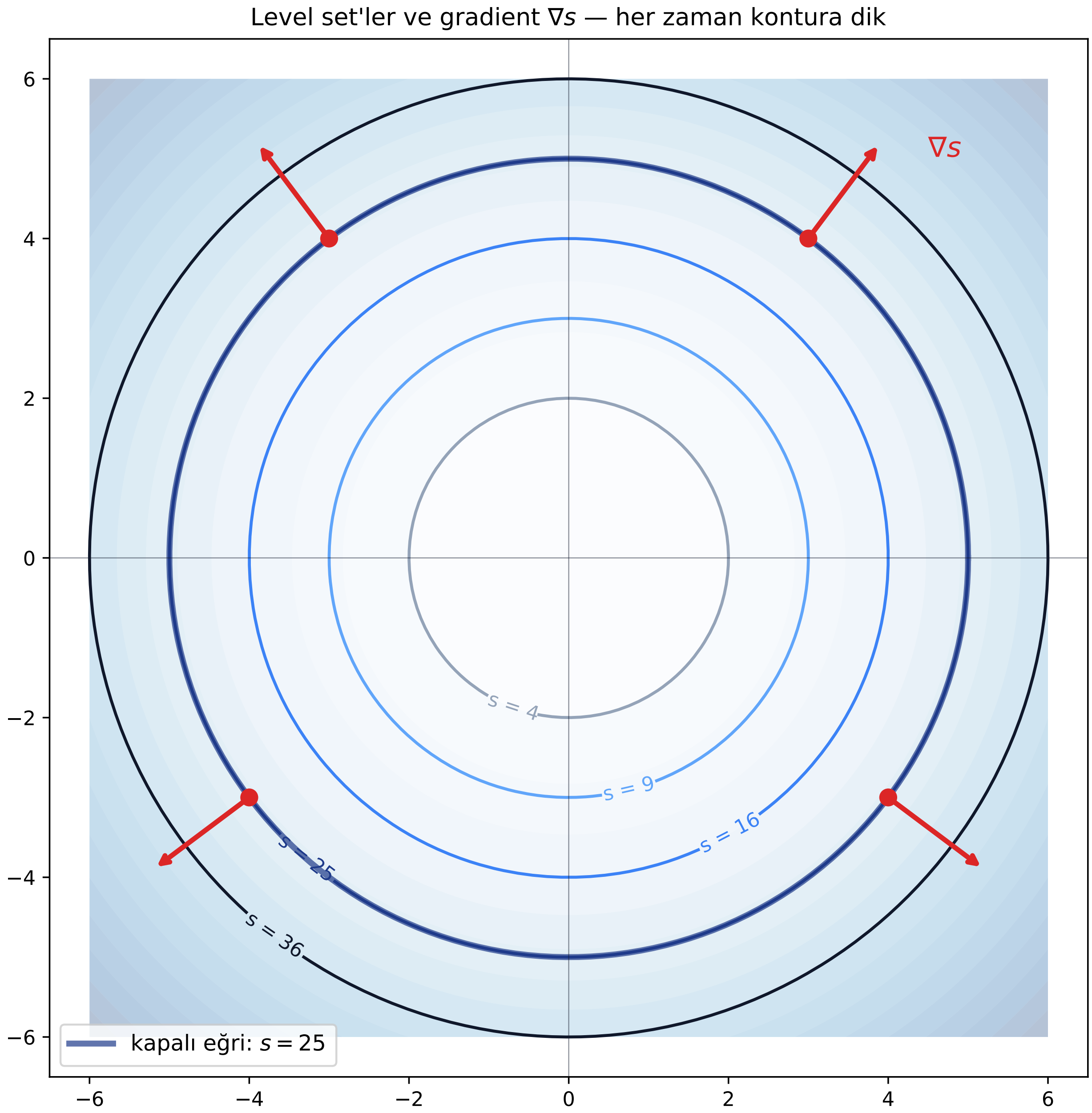
6.6 Daha Fazla Örnek: $\sin(x) \cdot y^2 = x$

$x^2 + y^2 = 5^2$'de özel bir şey yok; başka bir örnek deneyelim: $\sin(x) \cdot y^2 = x$. Bileşenleri dx, dy olan küçük bir adım al ve her iki tarafın türevini al.

Sol tarafta bir **çarpım** var, çarpım kuralı:

$$\sin(x) \cdot 2y\,dy + y^2 \cdot \cos(x)\,dx = dx$$

Sağ taraf yalnızca x olduğu için değişimi tam olarak dx'tir. İki tarafı eşitlemek şu demek: dx, dy adımı eğri üzerinde kalacaksa, sol ve sağ tarafların değişimi **aynı** olmalı.



Şekil 6.4: $s = x^2 + y^2$ fonksiyonunun kontur çizgileri (level set'leri). Gradient $\nabla s = (2x, 2y)$ her noktada konturlara dik; bir konturda kalmak için adım gradient'e dik olmalı.

Builder Notu — Kısıt Manifoldu

"İki tarafın değişimi eşit olmalı" ilkesi, bir **kısıt manifoldu** üzerinde kalmanın koşuludur. Generative modellerde ve fizik-bilgili ağlarda (PINN), bir denklemi (kısıtı) sağlayan çözümler ararsın; eğitim, tam bu "her iki tarafın değişimini eşitle" koşulunu bir kayıp terimine çevirir.

6.7 $\ln(x)$'in Türevi: Ters Fonksiyondan

Kapalı türevin güzel bir kullanımı: **yeni türev formülleri** keşfetmek. e^x'in türevinin kendisi olduğunu biliyoruz; peki onun ters fonksiyonu olan doğal logaritma $\ln(x)$'in türevi nedir?

$\ln(x)$'in grafiğini bir kapalı eğri olarak düşün: $y = \ln(x)$ olan tüm (x, y) noktaları. Bunu yeniden düzenle:

$$y = \ln(x) \qquad e^y = x$$

Artık e^y'nin türevini bildiğimiz için iki tarafın türevini alalım. Sol tarafın değişimi $e^y \cdot dy$, sağ tarafın değişimi dx. Eğri üzerinde kalmak için bu ikisi eşit olmalı:

$$e^y\, dy = dx \qquad \frac{dy}{dx} = \frac{1}{e^y} = \frac{1}{x}$$

Eğri üzerindeyken e^y zaten x'e eşit, dolayısıyla eğim $1/x$:

$$\frac{d}{dx}\ln(x) = \frac{1}{x}$$

"the derivative of ln of x is 1 divided by x." — Grant, 14:24

Builder Notu — Normalizing Flows

"Bir fonksiyonun türevini biliyorsan, tersinin türevini kapalı türevle çıkar" tekniği ML'de **normalizing flows**'un kalbidir. Bir flow, veriyi tersine çevrilebilir bir dönüşümle basit bir dağılıma eşler; olabilirliği hesaplamak için dönüşümün tersinin Jacobian'ının log-determinantı gerekir. $d(\ln x) = 1/x$ örneği, bu ters-fonksiyon-türevi mantığının en sade hâlidir.

6.8 Bu Dersin Özeti

1. **Kapalı eğri:** x ve y'nin bir denklemi sağladığı noktalar kümesi (bir fonksiyon grafiği olmak zorunda değil).
2. **Kapalı türev:** denklemin iki tarafını da türevle, sonra dy/dx'i çöz. Çemberde $(3, 4)$: $dy/dx = -x/y = -3/4$.
3. **İlgili oranlar (merdiven):** $x^2 + y^2 = 5^2$ zamanın fonksiyonu; türevle $\to 2x(dx/dt) + 2y(dy/dt) = 0 \to dx/dt = 4/3$ m/s.
4. **Anlam:** $s = x^2 + y^2$ iki değişkenli bir fonksiyondur; $ds = 2x \cdot dx + 2y \cdot dy$.

5. Eğri üzerinde kalmak = s'yi sabit tutmak = $ds = 0$. Ve $ds = \nabla s \cdot (dx, dy)$; gradient $\nabla s = (2x, 2y)$ her zaman level set'e diktir.
6. $\sin(x) \cdot y^2 = x$: iki tarafı türevle, eğri üstünde kalma = iki tarafın değişimi eşit.
7. $d(\ln x)/dx = 1/x$: $e^y = x$ kapalı eğrisinden.

! Tek bir cümle

Kapalı türev, bir eğriyi $y = f(x)$ olarak çözmeden denklemin iki tarafını türevleyip "eğri üstünde kal" koşulunu ($ds = 0$) yazmaktır; $ds = 2x \cdot dx + 2y \cdot dy$ aslında gradient ∇s ile adımın iç çarpımıdır — bu yüzden kapalı türev, çok değişkenli calculus ile gradient descent'in kapısıdır.

6.9 Kontrol Soruları

i Soru 1: $x^2 + y^2 = 25$ çemberinde (0, 5) noktasında dy/dx nedir? Sonuç mantıklı mı?

Cevap: Kapalı türev: $dy/dx = -x/y = -0/5 = 0$. Eğim sıfır, yani teğet **yatay**. Mantıklı: $(0, 5)$ çemberin en üst noktasıdır ve orada teğet doğru yataydır. (Benzer şekilde $(5, 0)$'da $-5/0$ tanımsız $\to$ dikey teğet.)

i Soru 2: Bir balon şişiyor: $V = (4/3)\pi r^3$ ve $dV/dt = 10$ cm^3/s. r = 5 cm iken yarıçap ne hızla büyüyor (dr/dt)?

Cevap: İki tarafı zamana göre türevle (zincir kuralı): $dV/dt = 4\pi r^2 \cdot (dr/dt)$. Yerine koy: $10 = 4\pi(25) \cdot (dr/dt) = 100\pi \cdot (dr/dt)$. Çöz: $dr/dt = 10/(100\pi) = 1/(10\pi) \approx 0{,}032$ cm/s.

i Soru 3: ds = 2x·dx + 2y·dy ifadesinde, 'eğri üzerinde kal' koşulu neden ds = 0?

Cevap: Eğri, $s = x^2 + y^2$ değerinin sabit (25) olduğu noktalar kümesidir. Eğri üzerinde kalmak, s'yi 25'te tutmak demek; bir niceliğin sabit kalması da değişiminin sıfır olması demektir, yani $ds = 0$. Geometrik olarak $ds = 0$ olan adımlar, gradient ∇s'ye dik olan adımlardır.

i Soru 4: (Builder) ds = ∇s·(dx, dy) iç çarpımı gradient descent ile nasıl bağlanır?

Cevap: İç çarpım, adım ∇s ile **aynı yöndeyse** en büyük (s en hızlı artar), ∇s'ye **dik** ise sıfırdır (s değişmez = level set boyunca). Gradient descent, bir kaybı en hızlı **azaltmak** için ters yöne, $-\nabla s$ yönüne adım atar (ds en negatif). Yani ∇s "en dik çıkış", $-\nabla s$ "en dik iniş", ∇s'ye dik yönler ise "kaybı değiştirmeyen" yönlerdir. Tüm optimizasyon bu resimden çıkar.

6.10 Egzersizler

Egzersiz 1. $x^3 + y^3 = 6xy$ (Descartes yaprağı) eğrisinde dy/dx'i kapalı türevle bul.

Egzersiz 2. *(İlgili oran)* Bir karenin alanı $A = s^2$. Alan $dA/dt = 4$ cm²/s hızla büyüyorsa, kenar $s = 10$ cm iken kenar ne hızla büyür (ds/dt)?

Egzersiz 3. $d(\sqrt{x})/dx$'i **kapalı türevle** bul: $y = \sqrt{x}$ ilişkisini $y^2 = x$ yazıp iki tarafı türevle. Sonucu $1/(2\sqrt{x})$ ile karşılaştır.

Egzersiz 4. *(Python — sembolik doğrulama)* SymPy'nin `idiff` fonksiyonuyla kapalı türevleri al.

```
x**2 + y**2 - 25   ->  dy/dx = -x/y
-x + y**2*sin(x)   ->  dy/dx = (-y*cos(x) + 1/y)/(2*sin(x))
(3,4) eğim: -3/4
```

Egzersiz 5. *(Sonraki dersin habercisi)* Bu seride türevi hep "çok küçük dx" sezgisiyle kullandık, ama "$dx \to 0$ limiti" tam olarak ne demek? Ders 7, **limitleri**, epsilon-delta tanımını ve L'Hôpital kuralını resmî olarak ele alacak.

6.11 Anahtar Kavramlar (Cheat Sheet)

Kavram	Tanım	Grant'ta
Kapalı eğri	x, y bir denklemi sağlayan noktalar (fonksiyon grafiği değil)	1m52
Kapalı türev	İki tarafı türevle: $x^2 \to 2x \cdot dx$, $y^2 \to 2y \cdot dy$	2m04
$dy/dx = -x/y$	Çemberin teğet eğimi; $(3, 4) \to -3/4$	2m51
İlgili oranlar	Merdiven: $2x(dx/dt) + 2y(dy/dt) = 0 \to 4/3$ m/s	3m26
$s = x^2 + y^2$	İki değişkenli fonksiyon; her $(x, y) \to$ bir sayı	8m03
$ds = 2x \cdot dx + 2y \cdot dy$	Bir (dx, dy) adımının s'yi ne kadar değiştirdiği	9m35
$ds = 0$ **koşulu**	Eğri (level set) üzerinde kalmanın koşulu	10m07
$\nabla s \perp$ **level set**	Gradient, kontura diktir; gradient descent'in temeli	8m30
$d(\ln x)/dx = 1/x$	$e^y = x$ kapalı eğrisinden (ters fonksiyon)	14m24

6.12 ML Bağlantıları Özeti

7 köprü

1. $ds = \nabla s \cdot (dx, dy) \rightarrow$ gradient; level set'e dik; gradient descent $-\nabla s$ yönünde en dik iniş.
2. **Kapalı türev** $\rightarrow$ implicit layers / Deep Equilibrium Models (DEQ): fixed-point'ten gradyan; MAML ve hyperparameter optimization'da implicit gradient.
3. **İlgili oranlar** $\rightarrow$ kısıtlı/manifold optimizasyon ($\|w\| = 1$), Riemannian gradient; bağlı oranların kilitlenmesi.
4. $ds = 0$ **koşulu** $\rightarrow$ eşitlik kısıtları, Lagrange çarpanları, kısıt manifoldu üzerinde kalma.
5. $d(\ln x) = 1/x$ **(ters fonksiyon)** $\rightarrow$ normalizing flows'ta log-det-Jacobian; değişken değiştirme formülü.
6. **İki değişkenli** s $\rightarrow$ çok değişkenli calculus; kısmi türevler $\partial/\partial x$, $\partial/\partial y$ ve tam gradient (bir sonraki büyük adım).
7. **Level set / kontur** $\rightarrow$ loss landscape görselleştirme; eğitim yörüngesini kontur çizgileri üzerinde okumak.

Tek bir şey alıp gideceksen

Kapalı türev sihir değil. $s = x^2 + y^2$ gibi iki değişkenli bir ifadenin küçük bir adımda ne kadar değiştiği $ds = 2x \cdot dx + 2y \cdot dy$'dir; bir eğri üzerinde kalmak, bu değişimi sıfırlamaktır ($ds = 0$). Bu ifade aslında gradient ∇s ile adımın iç çarpımıdır — ve gradient'in level set'e dik olması, tüm gradient-temelli optimizasyonun geometrik kalbidir.

7 Limitler, L'Hôpital ve Epsilon-Delta

Yaklaşmanın kesin tanımı + 0/0 belirsizliğini çözmek

ℹ Bölüm bilgisi

- **Grant'ın videosu:** YouTube — Chapter 7: Limits, L'Hôpital's rule, and epsilon delta definitions (≈18 dk)
- **Kaynak:** 3Blue1Brown — Essence of Calculus
- **Okuma süresi:** ≈24 dk

7.1 Bu Derste Ne Var?

Türev fikrinden integrallere geçmeden önce **limitlere** bir ara verelim. Aslında limit yeni bir şey değil — "yaklaşmak" kelimesinin ne demek olduğunu biliyorsan, limiti zaten biliyorsun. Ama bir ders ayırmanın üç nedeni var: (1) bu seride dx ve df'yi **somut, sonlu küçük dürtmeler** olarak düşünmenin, ders kitaplarındaki **resmî türev tanımıyla** birebir örtüştüğünü göstermek; (2) "yaklaşmak"ın **epsilon-delta** ile kesin anlamı; (3) limitleri hesaplayan akıllı bir hile: **L'Hôpital kuralı**.

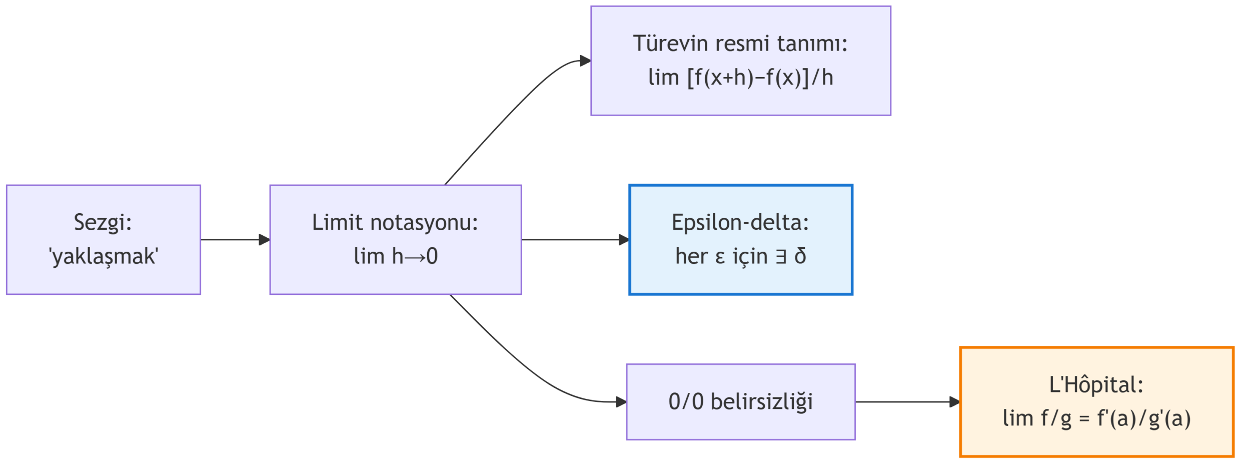
Üç ana fikir:

1. **Türevin resmî tanımı:** $f'(x) = \lim_{h \to 0}[f(x+h) - f(x)]/h$. Buradaki h, bizim dx'imizle aynı.
2. **Epsilon-delta:** "yaklaşmak", çıktı aralığını istediğin kadar küçültebilmektir (her ε için bir δ bulabilmek).
3. **L'Hôpital kuralı:** bir girdide $0/0$ görünüyorsa, pay ve paydanın **türevlerinin oranını** al.

> *"limitlerle ilgili büyük yaygara, sonsuz küçük değişiklikler hakkında konuşmaktan kaçınmamıza izin vermesi … değişkenimizde küçük bir değişikliğin boyutu 0'a yaklaştığında ne olacağını sormamıza izin vermesidir."* — Grant, 4:41 (Türkçe dublaj)

Builder Notu — ML Köprüleri

- **Limit / yakınsama** → eğitim yakınsaması ("loss minimuma yaklaşıyor"), optimizasyon ispatlarındaki ε-δ, sayısal tolerans (atol/rtol).
- **Türev = limit** → gradient kesin bir **limittir**, sonlu fark değil; autodiff'in sayısal türeve neden üstün olduğunun resmî temeli.
- $0/0$ **belirsizliği / L'Hôpital** → sayısal stabilite: log-sum-exp, softmax'ta $0/0$, küçük sayıya bölme; tekillik civarındaki oranları güvenle hesaplama.
- **"Tak-çalıştır yok, yaratıcılık gerek"** → yeni türev formülleri (ve genelde araştırma) sistematik



Şekil 7.1: Limitin üç yüzü: somut dürtmenin resmi karşılığı + epsilon-delta + L'Hôpital.

bir reçeteyle değil, sezgiyle keşfedilir.

7.2 Limit: "Yaklaşmak" Fikrine Resmî Bir İsim

Limit kavramı kavramsal olarak yeni bir şey getirmez: bir değerin başka bir değere **yaklaşması** sezgisine gösterişli bir gösterim atamaktan ibarettir.

Serinin başından beri türevi "dx kadar küçük bir dürtme, df kadar çıktı değişimi" diye anlattım ve dx'i **sonlu, sıfır-olmayan, somut** bir sayı olarak düşünmeni istedim — yeter ki o sayı 0'a yaklaştığında ne olduğunu sormayı unutma. İşte limit, bu "somut küçük dürtme" felsefesini tam katılığıyla destekleyen resmî araçtır: sonsuz küçüklükten bahsetmeden, "dürtme 0'a yaklaşırken oran neye yaklaşır?" diye sormamıza izin verir.

Builder Notu — Optimizasyon Yakınsaması

ML'de "yakınsama" sözcüğü tam da bu limit fikridir. "Eğitim yakınsadı" demek, kaybın bir değere keyfî yakınlıkta kalması (her ε için bir adım sayısından sonra $|L - L^*| < \varepsilon$) demektir. Optimizasyon teoremleri (SGD yakınsaması, öğrenme oranı koşulları) tamamen bu epsilon-delta dilinde yazılır.

7.3 Türevin Resmî Tanımı: $\lim_{h\to 0}$

Resmî gösterimle:

$$f'(x) = \lim_{h\to 0} \frac{f(x+h) - f(x)}{h}$$

Burada h, serideki dx ile **tamamen aynı şeydir** — f'nin girdisine yapılan sıfırdan farklı, son derece küçük (0,001 gibi) bir itme. Ders kitapları dx yerine genelde h (ya da Δx) yazar. Önemli nokta: sağ tarafta "sonsuz küçük değişim" gibi paradoksal hiçbir şey yok — limitlerin tüm amacı bundan kaçınmaktır.

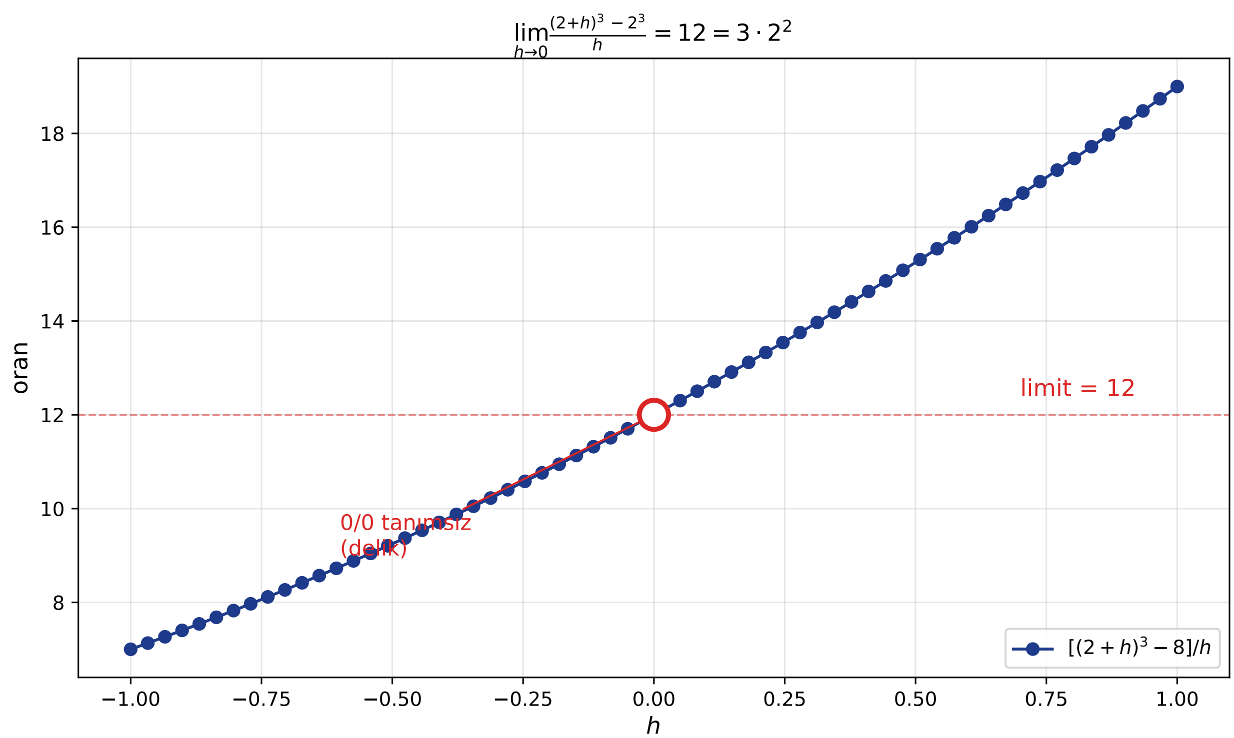
> *"bu somut, sonlu küçük dürtü felsefesiyle türevler hakkında söylediğim her şey, … bu resmî tanımın sadece bir çevirisidir."* — Grant, 4:30 (Türkçe dublaj)

Builder Notu — Autodiff Resmî Temel

Gradient, tam olarak bu limittir — sonlu fark değil. PyTorch/JAX'ın otomatik türevi, $[f(w+h)-f(w)]/h$ oranını küçük bir h ile **yaklaşık** hesaplamaz; cebirsel limiti (kuralları zincirleyerek) **kesin** verir. Sonlu fark yöntemi h seçimine ve yuvarlama hatasına takılır; limit tanımı bu sorunu baştan ortadan kaldırır.

7.4 Bir Limiti Hesaplamak: $0/0$ ve Delik

Şu fonksiyonu düşün: $[(2+h)^3-2^3]/h$. Bu, x^3'ün $x=2$'deki türevinin tanımını çözünce çıkan ifade. Grafiği güzel, sürekli görünen bir eğridir. Ne var ki $h=0$'da yerine koyarsan $0/0$ elde edersin, ki bu **tanımsızdır**. Yani grafiğin tam o noktasında bir **delik** vardır.



Şekil 7.2: $[(2+h)^3-8]/h$ ifadesi $h=0$'da $0/0$ olduğu için tanımsız (siyah halka), ama her iki taraftan 12'ye yaklaşır. Limit = 12.

Fonksiyon, 0'a istediğin kadar yakın girdiler için kusursuzca tanımlıdır. $h\to0$ iken çıktının yaklaştığı değere bak: hangi taraftan gelirsen gel, **12**'ye yaklaşır.

$$\lim_{h\to0}\frac{(2+h)^3-2^3}{h}=12$$

Limit budur: deliğin tam üzerindeki değeri hesaplayamasak da, çevresindeki değerlerin yaklaştığı sayıyı (12) söyleyebiliriz.

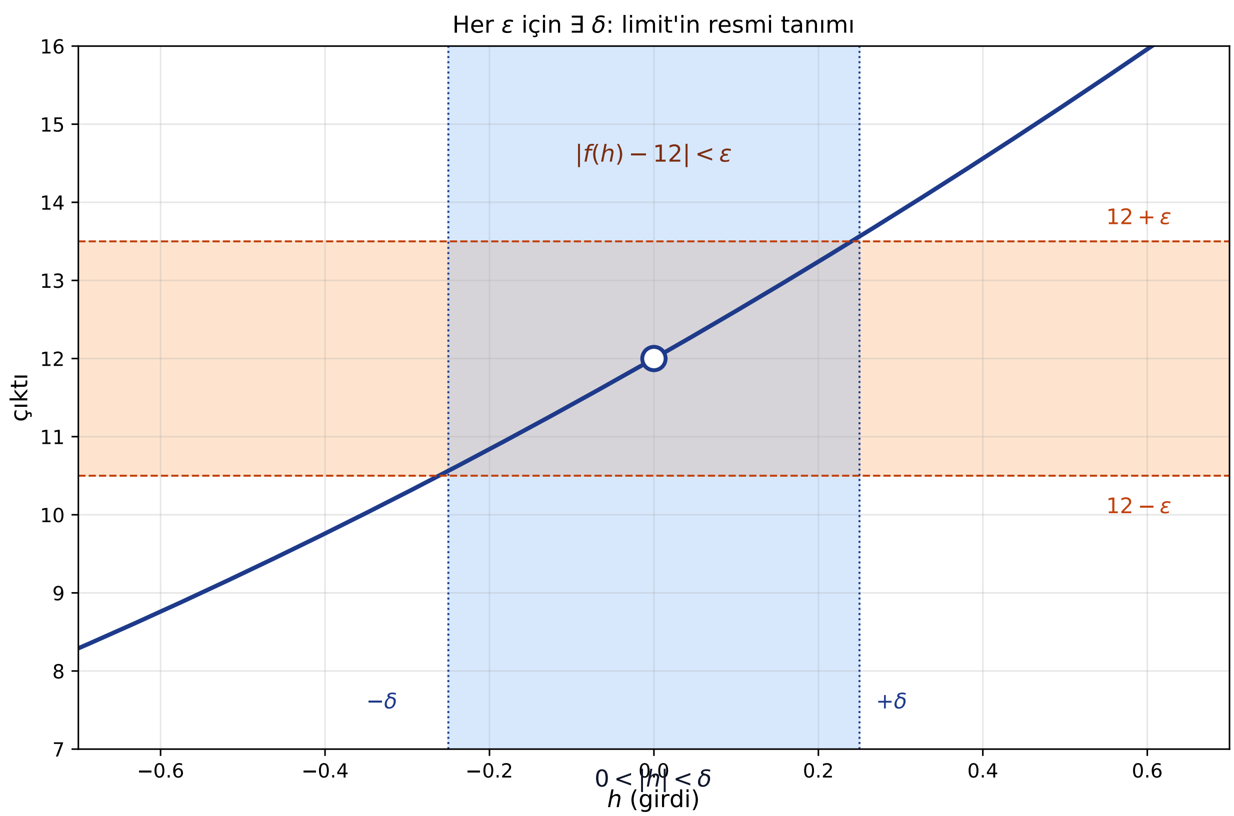
Builder Notu — Sayısal Kâbuslar

$0/0$ biçimindeki "delikler" ML'de sayısal kâbuslardır. Softmax'ta tüm logitler eşitken, attention'da maskelenmiş bir satırda, ya da bir kaybı sıfıra giden bir paydaya bölerken $0/0$ ortaya çıkabilir. Çözüm ya analitik limittir (ifadeyi sadeleştirip deliği "doldurmak", örn. log-sum-exp hilesi) ya da küçük bir ε eklemek (10^{-8}).

7.5 Epsilon-Delta: "Yaklaşmak"ın Kesin Tanımı

Calculus'u icat eden matematikçi olsaydın ve biri "yaklaşmak derken tam olarak neyi kastediyorsun?" diye sorsaydı, sinir bozucu ama haklı bir soru olurdu. İşte kesin yanıt.

Yasak nokta olan 0'ın kendisi hariç, 0'a yakın bir **girdi aralığına** ve buna karşılık gelen **çıktı aralığına** bak. Girdi aralığını 0'a giderek daha sıkı yaklaştırdıkça, çıktı aralığı 12'ye giderek daha sıkı kapanıyorsa — ve bu çıktı aralığı **istediğin kadar küçük** yapılabiliyorsa — limit 12'dir.



Şekil 7.3: Epsilon-delta: çıktı toleransı ε ne kadar küçük olursa olsun, ona karşılık bir δ bulunabilir. $|h| < \delta$ ise $|f(h) - 12| < \varepsilon$.

Bunu kesinleştiren epsilon-delta tanımıdır. 12'den bir ε **mesafesi** (çıktı toleransı, istediğin kadar küçük) düşün. Limit varsa: 0'ın etrafında öyle bir δ **mesafesi** (girdi aralığı) bulabilirsin ki,

$$0 < |h| < \delta \quad \text{ise} \quad |f(h) - 12| < \varepsilon$$

Kilit nokta: bu **her ε için** geçerli — ε ne kadar küçük olursa olsun, ona karşılık gelen bir δ her zaman bulunabilir.

Limit ne zaman yoktur? Karşı örnek: 0'da bir **sıçrama** yapan fonksiyon — sağdan yaklaşınca 2'ye, soldan yaklaşınca 1'e gider. Tek bir net değere yaklaşmadığından limit tanımsızdır.

> *"bir sınır mevcut olduğunda, çıkış aralığını istediğiniz kadar küçük yapabilirsiniz, ancak sınır olmadığında … bu çıkış aralığı belirli bir değerden daha küçük olamaz."* — Grant, 8:25 (Türkçe dublaj)

Builder Notu — Yakınsama Teoremi

Epsilon-delta, optimizasyon teorisinin dilidir. "SGD yakınsar" demek: her $\varepsilon > 0$ için öyle bir adım sayısı N vardır ki, $n > N$ olan her adımda parametreler optimuma ε mesafesindedir. "Yakınsamıyor" durumu ise tam o sıçrayan fonksiyon gibidir: kayıp belirli bir bandın altına inmez, salınır. Learning rate'in 0'a uygun hızda gitmesi koşulları (Robbins-Monro) tam bu ε-δ çerçevesinde kanıtlanır.

7.6 L'Hôpital Kuralı: $0/0$'ı Türevle Çözmek

Tüm bunlar teori ağırlıklıydı; şimdi limitleri gerçekten **hesaplayan** bir hile. Şu fonksiyonu incele:

$$\frac{\sin(\pi x)}{x^2 - 1}$$

$x = 1$'de $\sin(\pi) = 0$ ve $x^2 - 1 = 0$, yani $0/0$ — tanımsız.

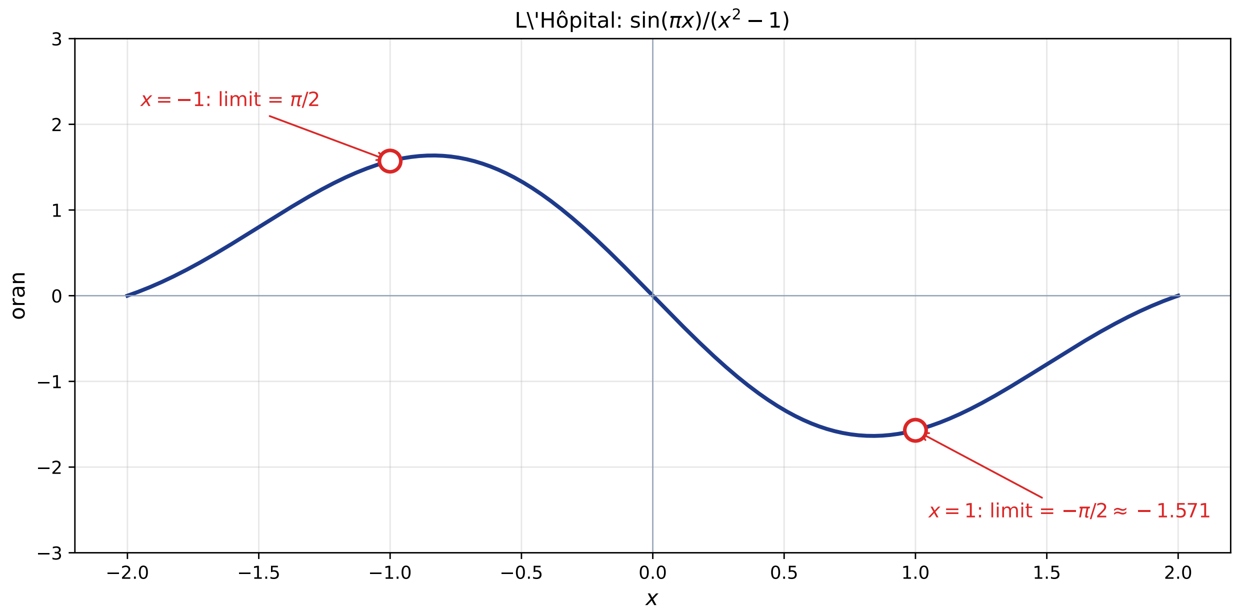
Hile: $x = 1$ civarına yakınlaş, küçük bir dx adımı at. Pay $\sin(\pi x)$'in değişimi (zincir kuralı) $\cos(\pi x) \cdot \pi \cdot dx$; $x = 1$'de $\cos(\pi) = -1$ olduğundan bu $-\pi \cdot dx$. Payda $x^2 - 1$'in değişimi $2x \cdot dx$; $x = 1$'de $2 \cdot dx$. Oran:

$$\frac{\sin(\pi x)}{x^2 - 1} \approx \frac{\cos(\pi) \cdot \pi \, dx}{2 \cdot 1 \cdot dx} = \frac{-\pi \, dx}{2 \, dx} = -\frac{\pi}{2}$$

dx'ler birbirini götürür; geriye $-\pi/2 \approx -1{,}5708$ kalır. Genel olarak, $x = a$'da ikisi de 0 olan türevlenebilir f ve g için:

$$\lim_{x \to a} \frac{f(x)}{g(x)} = \frac{f'(a)}{g'(a)} \qquad (f(a) = g(a) = 0)$$

Bu akıllı numara **L'Hôpital kuralıdır**.



Şekil 7.4: $\sin(\pi x)/(x^2 - 1)$ $x = \pm 1$'de $0/0$ delikleri taşır, ama her iki delikte de limit tanımlı: $-\pi/2$ ve $\pi/2$.

> **Builder Notu — Belirsiz Biçimler**
>
> L'Hôpital'in ruhu — "0/0'ı, pay ve paydanın yerel lineer davranışının oranıyla çöz" — ML'de **belirsiz biçimleri** ehlileştirmenin yoludur. Bir kayıp fonksiyonu $0/0$'a giden bir oran içerdiğinde (örneğin normalize edilmiş bir ağırlık, sıfıra giden bir sıcaklıkta softmax, ya da KL diverjansında $p \to 0$ terimleri), limiti analitik almak çökmeyi önler.

7.7 L'Hôpital'in Sınırı: Yaratıcılık Gerekir

Türevin tanımı zaten $0/0$ biçiminde bir kesrin limiti olduğuna göre, L'Hôpital'i **yeni türev formülleri** keşfetmek için kullanabilir miyiz? Hayır — bu hile yapmak olur, çünkü L'Hôpital payın türevini gerektirir, ki keşfetmeye çalıştığın şey tam da odur (döngüsel).

İş türev formülleri keşfetmeye gelince, sistematik bir "tak-çalıştır" yöntemi yoktur. Ama bu iyi bir şey: bir problemi çözmek yaratıcılık gerektiriyorsa, bu, gerçek bir şey yaptığının ve gelecekteki problemler için güçlü bir araç geliştirdiğinin işaretidir.

> *"bu seride oldukça fazla yaptığımız … türev formüllerini keşfetmeye gelince, sistematik bir tak-çalıştır yöntemi yoktur. Ama bu iyi bir şey!"* — Grant, 17:01 (Türkçe dublaj)

> **Builder Notu — Araştırma Disiplini**
>
> "Sistematik tarif yok, yaratıcılık şart" — bu, ML araştırmasının ta kendisidir. Yeni bir mimari, kayıp fonksiyonu veya optimizasyon hilesi, mekanik bir prosedürle değil; mevcut araçları (gradient, zincir

kuralı, üstel/log) yapısal anlayıp yeni bir biçimde birleştirerek bulunur.

7.8 Bu Dersin Özeti

1. Limit, "yaklaşmak" sezgisine resmî bir isimdir; kavramsal yenilik getirmez ama "somut küçük dürtme" felsefesini katı bir temele oturtur.
2. Türevin resmî tanımı: $f'(x) = \lim_{h\to 0}[f(x+h) - f(x)]/h$.
3. $[(2+h)^3 - 2^3]/h$ ifadesi $h = 0$'da $0/0$ (bir delik), ama $h \to 0$ iken limit tam 12'dir.
4. Epsilon-delta: her ε için bir δ bulabilmek; çıktı aralığı istediğin kadar küçük yapılabilir.
5. Limit yoksa: sıçrayan fonksiyon — sağdan ve soldan farklı değerlere gider.
6. L'Hôpital kuralı: $0/0$ biçimi için $\lim f/g = f'(a)/g'(a)$. Örnek: $\sin(\pi x)/(x^2 - 1) \to -\pi/2$.
7. L'Hôpital yeni türev formülleri keşfedemez (döngüsel); formül keşfi sistematik değil, yaratıcı bir iştir.

! Tek bir cümle

Limit, "$dx \to 0$ iken oran neye yaklaşır?" sorusuna sonsuz küçüklükten hiç bahsetmeden kesin yanıt verir (epsilon-delta); türev bu limitin resmî hâlidir ve $0/0$ biçimindeki limitler, pay ile paydanın türevlerinin oranıyla (L'Hôpital) çözülür.

7.9 Kontrol Soruları

i Soru 1: lim(x→3) (x^2 – 9)/(x – 3) nedir?

Cevap: $x = 3$ koyunca $0/0$ (belirsiz). İki yol: (a) Çarpanlara ayır: $(x-3)(x+3)/(x-3) = x+3 \to x \to 3$'te 6. (b) L'Hôpital: pay türevi $2x$, payda türevi $1 \to 2x/1$, $x = 3$'te 6.

i Soru 2: Bir fonksiyon x = a'da tanımsızken (0/0 deliği) limitin a'da var olması nasıl mümkün?

Cevap: Limit, fonksiyonun **a noktasındaki değeri** değil, a'nın **çevresindeki** davranışıdır. Delikteki noktayı hesaplayamasak da, çevresindeki girdilerin çıktıları tek bir değere yaklaşıyorsa limit odur. Süreklilik ile limit ayrı kavramlardır.

i Soru 3: lim(x→0) sin(x)/x nedir? (Ünlü limit)

Cevap: $x = 0$'da $0/0$. L'Hôpital: pay türevi $\cos(x)$, payda türevi $1 \to \cos(x)/1$, $x = 0$'da $\cos(0) = 1$. Yani $\lim_{x\to 0} \sin(x)/x = 1$. (Bu aynı zamanda sin'in $x = 0$'daki türevinin $\cos(0) = 1$ olmasıdır; küçük x için $\sin(x) \approx x$.)

Soru 4: (Builder) 'Eğitim yakınsadı' ifadesini epsilon-delta diliyle nasıl tanımlarsın?

Cevap: Her $\varepsilon > 0$ için öyle bir adım sayısı N vardır ki, $n > N$ olan her adımda kayıp, hedef değerine ε mesafesindedir: $|L_n - L^*| < \varepsilon$. Yani kayıp, herhangi bir küçük toleransın içine eninde sonunda girer ve orada kalır. Yakınsamama (sıçrayan fonksiyon gibi) ise kaybın belirli bir bandın altına hiç inmemesi, salınıp durmasıdır.

7.10 Egzersizler

Egzersiz 1. $\lim_{x\to 2}(x^2 - 4)/(x - 2)$'yi iki yolla bul: (a) çarpanlara ayırarak, (b) L'Hôpital ile.

Egzersiz 2. $\lim_{x\to 0}(1 - \cos x)/x^2$'yi L'Hôpital ile bul. (İpucu: bir kez uygulayınca yine $0/0$ çıkar — bir kez daha uygula. Sonuç $1/2$.)

Egzersiz 3. *(Epsilon-delta)* $f(x) = 2x + 1$ için $\lim_{x\to 3} f(x) = 7$. $\varepsilon = 0{,}1$ verildiğinde, "$0 < |x - 3| < \delta$ ise $|f(x) - 7| < 0{,}1$" koşulunu sağlayan bir δ bul.

Egzersiz 4. *(Python — sembolik + sayısal)* $\sin(\pi x)/(x^2 - 1)$'in $x \to 1$ limitini SymPy ile bul.

```
SymPy limit: -pi/2
sin(x)/x   : 1
h=1e-02  oran=-1.562724
h=1e-04  oran=-1.570718
h=1e-06  oran=-1.570796
-pi/2 = -1.5707963267948966
```

Egzersiz 5. *(Sonraki dersin habercisi)* Tüm seri boyunca "fonksiyon → türevi" yönünde gittik. Ters soru: türevi bilinen bir fonksiyonun **kendisini** nasıl buluruz (antitürev)? Ders 8, integrasyonu ve Calculus'un Temel Teoremi'ni anlatacak.

7.11 Anahtar Kavramlar (Cheat Sheet)

Kavram	Tanım	Grant'ta
Limit	"Yaklaşmak"a resmî isim; $\lim_{h\to 0}$	0m25
Türevin resmî tanımı	$f'(x) = \lim_{h\to 0}[f(x+h) - f(x)]/h$	2m08
$h = dx$	Sonsuz küçük değil, sıradan sıfır-olmayan küçük sayı	3m18
$0/0$ **deliği**	Tanımsız nokta; limit yine de var olabilir	5m32
Epsilon-delta	Her ε için bir δ; çıktı aralığı keyfî küçük	7m48

Kavram	Tanım	Grant'ta
Limit yok (sıçrama)	Sağ/sol farklı; çıktı tek değere büzülmez	7m02
L'Hôpital kuralı	$0/0 \to f'(a)/g'(a)$	16m13
$\sin(\pi x)/(x^2 - 1) \to -\pi/2$	L'Hôpital örneği	13m50
Yaratıcılık gerekir	Formül keşfinde "tak-çalıştır" yöntemi yok	17m01

7.12 ML Bağlantıları Özeti

7 köprü

1. **Türev = limit** → gradient kesin bir limittir; autodiff'in sonlu farktan üstünlüğünün resmî temeli.
2. **Epsilon-delta** → optimizasyon yakınsama teoremleri (SGD, Robbins-Monro learning rate koşulları).
3. $0/0$ **/ L'Hôpital** → sayısal stabilite: log-sum-exp, softmax $0/0$, paydaya ε ekleme.
4. **Süreklilik / delik** → ReLU köşesi, parçalı fonksiyonlar, türevlenemezlik noktaları.
5. $\sin(x)/x \to 1$ → küçük açı/küçük adım yaklaşımı; yerel lineerleştirme.
6. **Limit yok (sıçrama)** → süreksizlik; eğitimde yakınsamama (salınım, ıraksama).
7. **Yaratıcılık gerekir** → yeni mimari/kayıp/optimizasyon mekanik tarifle değil, yapısal sezgiyle bulunur.

! Tek bir şey alıp gideceksen

"$dx \to 0$" tüm seride bir sezgiydi; limit ona kesin bir anlam verir — sonsuz küçüklükten hiç bahsetmeden, "her ε için bir δ" diliyle. Türev bir limittir, $0/0$ delikleri limitle doldurulur (L'Hôpital), ve aynı kesinlik ML'de "model gerçekten yakınsıyor mu?" sorusunu yanıtlar.

8 İntegrasyon ve Temel Teorem

Hızdan mesafeye, antitürev ve FTC

Bölüm bilgisi

- **Grant'ın videosu:** YouTube — Chapter 8: Integration and the fundamental theorem of calculus (≈20 dk)
- **Kaynak:** 3Blue1Brown — Essence of Calculus
- **Okuma süresi:** ≈26 dk

8.1 Bu Derste Ne Var?

Ders 1'de integrali (eğri altı alan) sezmiş, Ders 2-7'de türevi derinlemesine işlemiştik. Bu ders çemberi kapatıyor: **integral, türevin tersidir**. Tek bir örneğe odaklanıyoruz — hızdan mesafeyi bulmak (Ders 2'deki hareketli arabanın **dualü**). Grant'ın hedefi, integralin türevin tersi olduğunun neredeyse **bariz** hissettirilmesi.

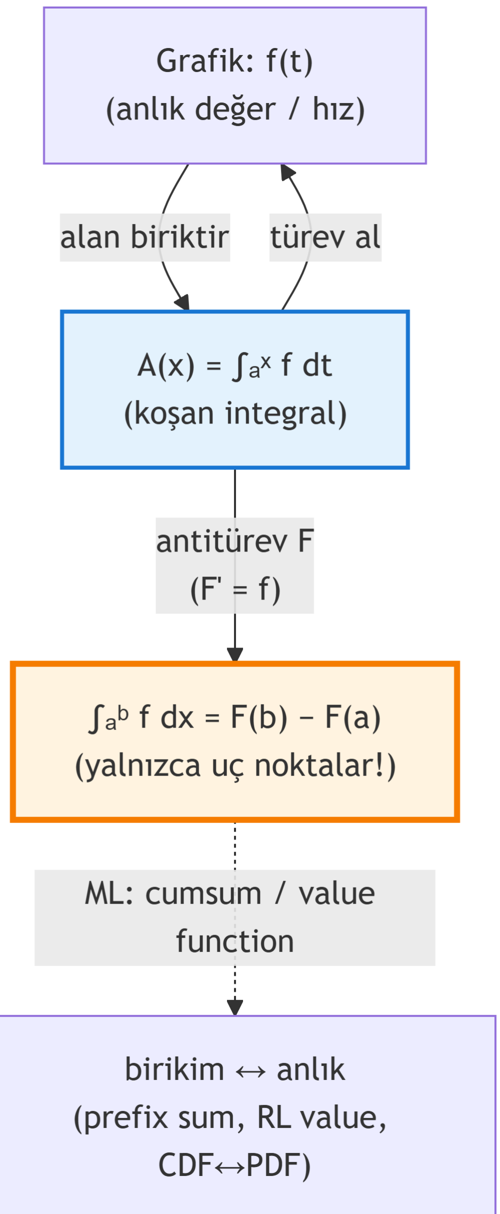
Üç ana fikir:

1. **İntegral = türevin tersi.** Hız $v(t)$'den mesafe $s(t)$'yi bulmak = **antitürev** (türevi v olan fonksiyonu bulmak).
2. **İntegral = eğri altı alan = küçük dilimlerin** $(v \cdot dt)$ **toplamının limiti.** Notasyon: $\int_0^8 v(t)\,dt$.
3. **Calculus'un Temel Teoremi (FTC):** alan fonksiyonunun türevi, grafiğin kendisidir ($ds/dt = v$); ve $\int_a^b f\,dx = F(b) - F(a)$.

> *"Too often in math, we dive into showing that a certain fact is true … before stepping back and making sure it feels reasonable, and preferably obvious."* — Grant (Grothendieck'ten esinle), 0:18

Builder Notu — ML Köprüleri

- **İntegral = küçük niceliklerin toplamı** → beklenen değer $E[f(X)] = \int f(x)p(x)dx$, **Monte Carlo** entegrasyon, ELBO.
- **Antitürev / FTC** → kümülatif ödül (RL'de **value function**) ↔ anlık ödül; **cumsum** ↔ **diff**; **CDF** ↔ **PDF** ilişkisi.
- **"Sadece iki uç nokta" (FTC)** → telescoping toplamlar, verimli kümülatif hesap; bir aralığın toplamını prefix-sum farkıyla almak.
- **İşaretli alan** → işaretli ölçüler, net akış, KL diverjansındaki pozitif/negatif terimler.



Şekil 8.1: FTC üçgeni: alan fonksiyonu (sürekli birikim) ↔ grafiğin kendisi (anlık değer) ↔ antitürev (uç noktalarla hesap).

8.2 Tersine Problem: Hızdan Mesafe

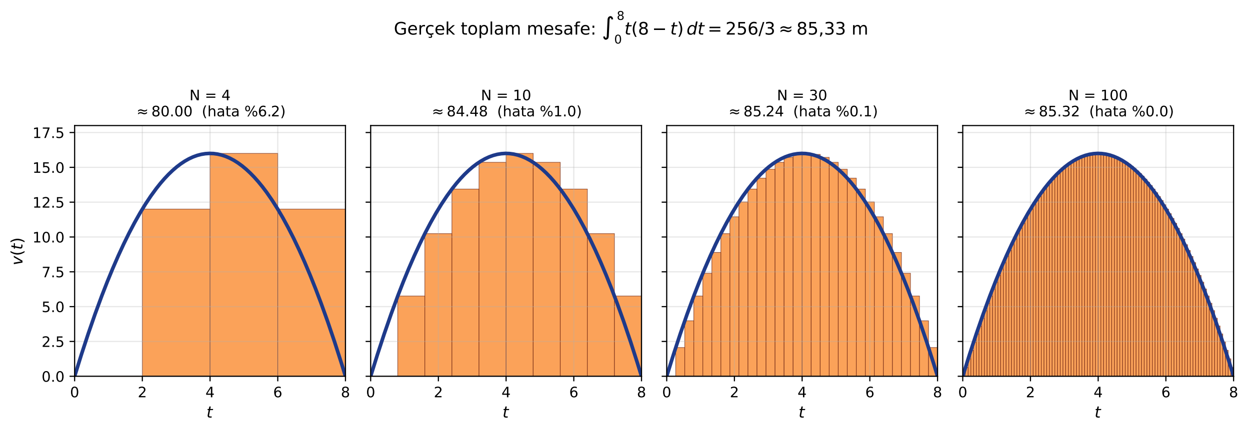
Bir arabada oturduğunu, pencereden dışarıyı göremediğini, yalnızca **hız göstergesini** gördüğünü hayal et. Araba bir noktada hareket eder, hızlanır, sonra yavaşlayıp durur — toplam 8 saniye. Soru: yalnızca hız göstergesine bakarak, bu sürede ne kadar yol gittiğini bulabilir misin?

Diyelim hızı $v(t) = t(8 - t)$ ile modelledin (m/s). Ders 2'de **tersini** yapıyorduk: $s(t)$'yi biliyorduk, türevini alarak $v(t)$'yi buluyorduk. Şimdi elimizde yalnızca $v(t)$ var; dolayısıyla mesafeyi bulmak, "türevi $t(8 - t)$ olan fonksiyon hangisidir?" diye sormaya iner. Buna bir fonksiyonun **antitürevini** bulmak denir.

> Builder Notu — Kümülatif Nicelikler
>
> "Türevi bilinen fonksiyonu geri bul" = antitürev, ML'de **kümülatif** niceliklerin temelidir. Bir ödül akışından toplam (return) hesaplamak, anlık değerlerden birikimli değere geçmektir — ayrık dünyada `cumsum`, sürekli dünyada integral. RL'deki value function (gelecekteki ödüllerin birikimi), anlık ödülün antitürevi gibidir.

8.3 Değişen Hızı Dilimlemek: $\int$ Notasyonu

Zaman eksenini 0 ile 8 saniye arasında, her biri küçük bir dt genişliğinde birçok aralığa böl. Her aralıkta hızı **sabitmiş gibi** yaklaştır; mesafe $\approx v(t) \cdot dt$.



Şekil 8.2: Riemann toplamı: $v(t) = t(8 - t)$ altındaki alan, N arttıkça gerçek değer $256/3 \approx 85{,}33$'e yakınsar.

$$\int_0^8 v(t)\,dt = \lim_{dt \to 0} \sum_t v(t)\,dt$$

Bu notasyon: bilindik $\sum$ yerine uzatılmış bir "S" kullanırız, çünkü bu ifade belirli bir dt için belirli bir toplam **değil**; $dt \to 0$ iken o toplamın **yaklaştığı** değerdir. Ve yaklaştığı şey, eğri ile yatay eksen arasındaki **alandır**.

> *"this expression is called an integral of v of t, since it brings all of its values together, it integrates them."* — Grant, 8:22

Builder Notu — Monte Carlo

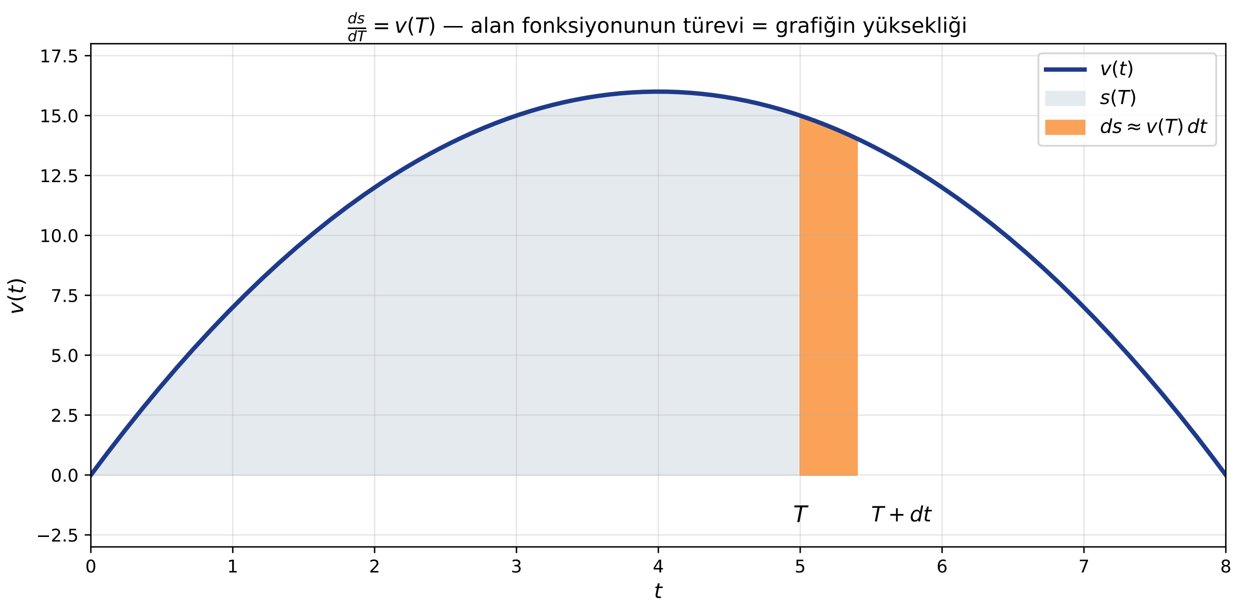
$\int$ = "sürekli $\sum$" denklemi, ML'deki beklenen değerin tam tanımıdır: $E[f(X)] = \int f(x)p(x)dx$. Bunu hiçbir zaman analitik hesaplamayız; **Monte Carlo** ile tahmin ederiz: N örnek al, f değerlerinin ortalamasını al. Bu ortalama, tam da "$v(t) \cdot dt$'leri topla" mantığıdır; $N \to \infty$ iken gerçek integrale yakınsar.

8.4 Calculus'un Temel Teoremi: Alan Fonksiyonunun Türevi

Şimdi alanı bulmaya çalışmak yerine, sağ uç noktayı bir **değişken** T olarak düşün. 0 ile T arasındaki alan — yani $\int_0^T v(t)\,dt$ — T saniye sonra kat edilen mesafedir; bu zaten bizim $s(T)$ mesafe fonksiyonumuz.

$$s(T) = \int_0^T v(t)\,dt$$

Bu alan fonksiyonunun **türevi** nedir? Girdiyi dt kadar dürtersen alan, ince bir dilim (ds) kadar artar. Bu dilimin yüksekliği o noktadaki grafik değeri $v(t)$, genişliği dt:



Şekil 8.3: FTC birinci yarı: alan fonksiyonu $s(T)$'yi dt dürt, $ds \approx v(T) \cdot dt$ ince dilim. $ds/dt = v(T)$ — alan fonksiyonunun türevi = grafiğin yüksekliği.

$$ds \approx v(t)\,dt \qquad \frac{ds}{dt} = v(t)$$

Bu çok genel bir argümandır: **bir grafiğin altındaki alanı veren herhangi bir fonksiyonun türevi, grafiğin kendisidir.**

> *"the derivative of any function giving the area under a graph like this is equal to the function for the graph itself."* — Grant, 11:09

Builder Notu — CDF ↔ PDF

$s(T) = \int_0^T v$, üst sınırın değişken olduğu bir **koşan integraldir** — olasılıkta bunun adı CDF'tir: $F(x) = \int_{-\infty}^{x} p$. Ve $ds/dt = v(t)$, CDF'nin türevinin PDF olması ($F' = p$) ile birebir aynı ilişkidir. RL'de value function (kümülatif ödülün koşan integrali) ile anlık ödül arasındaki bağ da budur.

8.5 Antitürev ve Sınırları Değerlendirmek

$v(t) = t(8-t) = 8t - t^2$ ise, türevi buna eşit olan s nedir? Parça parça: $8t$ için, $d(t^2) = 2t$ olduğundan $d(4t^2) = 8t$. $-t^2$ için, $d(t^3) = 3t^2$ olduğundan $d(-\frac{1}{3}t^3) = -t^2$. Birleştir:

$$v(t) = 8t - t^2 \qquad s(t) = 4t^2 - \frac{1}{3}t^3 + C$$

Bir incelik: herhangi bir **sabit** C eklemek türevi değiştirmez. Yani aslında sonsuz tane antitürev var, hepsi $4t^2 - \frac{1}{3}t^3 + C$ biçiminde.

8 saniyedeki toplam mesafe, ifadeyi $t = 8$'de değerlendirmektir:

$$\int_0^8 (8t - t^2)\,dt = \left[4t^2 - \frac{1}{3}t^3\right]_0^8 = \frac{256}{3} \approx 85{,}33$$

Genel kural, **Calculus'un Temel Teoremi**:

$$\int_a^b f(x)\,dx = F(b) - F(a), \qquad F'(x) = f(x)$$

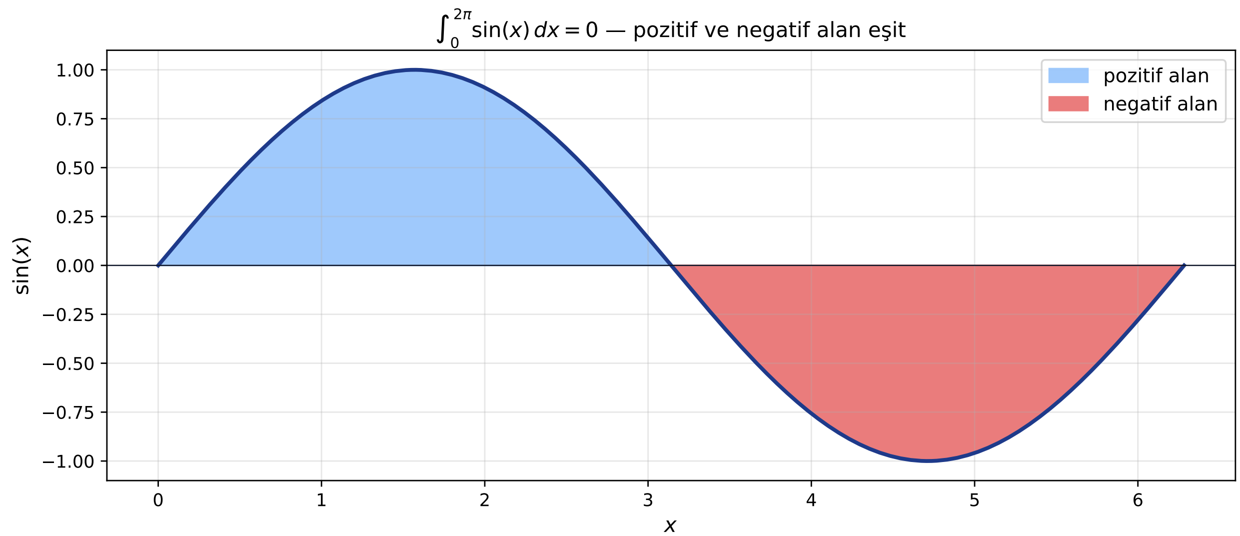
> *"to actually compute it using an antiderivative, you only look at two inputs … It almost feels like cheating."* — Grant, 15:56

Builder Notu — Prefix Sum

FTC'nin "çılgın" yanı şu: integral, alt ve üst sınır arasındaki **tüm continuum'u** hesaba katar; ama antitürevle hesaplarken yalnızca **iki** noktaya (sınırlara) bakarsın. Bu, ayrık dünyadaki **prefix-sum / telescoping** ile aynıdır: bir aralığın toplamını, birikimli dizinin iki ucundaki farkıyla $O(1)$'de alırsın.

8.6 İşaretli Alan

Ya hız bir noktada negatifse, yani araba geri giderse? Küçük bir zaman aralığındaki mesafe ds yine $v(t) \cdot dt$'dir; sadece v negatif olduğundan ds de negatiftir.



Şekil 8.4: İşaretli alan: sin(x) $[0, 2\pi]$'de — pozitif yarı (mavi) ile negatif yarı (kırmızı) birbirini götürür, net integral 0.

Bir grafik yatay eksenin altına indiğinde, o bölge ile eksen arasındaki alan **negatif** sayılır. İntegraller alanı değil, grafik ile yatay eksen arasındaki **işaretli alanı** ölçer.

> *"integrals don't measure area per se, they measure the signed area between the graph and the horizontal axis."* — Grant, 18:50

Builder Notu — Advantage ve KL

İşaretli alan, ML'de **net katkı** olan her yerde devrededir. KL diverjansı, beklenti farkları, politika gradyanındaki avantaj (advantage, pozitif/negatif) — hepsi pozitif ve negatif bölgelerin toplamıdır.

8.7 Bu Dersin Özeti

1. İntegral, türevin tersidir; hızdan mesafe bulmak = antitürev bulmak.
2. Sabit hızda mesafe = hız × süre = bir dikdörtgenin alanı.
3. Değişen hızı aralıklara böl, her birinde mesafe $\approx v(t) \cdot dt$, hepsini topla. $\int_0^8 v(t)dt$, $dt \to 0$ limiti.
4. FTC (birinci yarı): alan fonksiyonu $s(T) = \int_0^T v$'nin türevi, grafiğin kendisidir: $ds/dt = v(t)$.
5. Antitürev: türevi v olan fonksiyon. Sonsuz tane vardır $(+C)$.
6. FTC (ikinci yarı): $\int_a^b f(x)dx = F(b) - F(a)$. Tüm continuum, yalnızca iki uç noktayla hesaplanır.
7. İşaretli alan: grafik eksenin altındayken alan negatif sayılır.

! Tek bir cümle

İntegral, bir eğri altındaki (işaretli) alandır — sayısız $v(t) \cdot dt$ diliminin $dt \to 0$ limiti — ve Calculus'un Temel Teoremi onu türevin tersine bağlar: alan fonksiyonunun türevi grafiğin kendisidir, dolayısıyla $\int_a^b f(x)dx = F(b) - F(a)$.

8.8 Kontrol Soruları

i Soru 1: $\int_0^3 2x, dx$ integralini antitürevle hesapla.

Cevap: $2x$'in antitürevi x^2 (çünkü $d(x^2)/dx = 2x$). FTC: $F(3) - F(0) = 3^2 - 0^2 = 9$. Geometrik kontrol: $2x$ doğrusu altında, taban 3, yükseklik 6 olan üçgen → $\frac{1}{2} \cdot 3 \cdot 6 = 9$. ✓

i Soru 2: d/dT [$\int_0^T \sin(t), dt$] nedir?

Cevap: FTC'nin birinci yarısı: bir koşan integralin (üst sınıra göre) türevi, integrandın kendisidir. Yani $d/dT \int_0^T \sin(t)dt = \sin(T)$.

i Soru 3: $\int_0^{2\pi} \sin(x), dx$ nedir? Sonuç neden sezgisel?

Cevap: sin'in antitürevi $-\cos$. FTC: $[-\cos x]_0^{2\pi} = -\cos(2\pi) + \cos(0) = -1 + 1 = 0$. Sezgisel: $\sin(x)$ ilk yarıda pozitif, ikinci yarıda negatiftir; pozitif alan ile negatif alan eşit büyüklükte ve **işaretli alan** olarak birbirini tam götürür.

i Soru 4: (Builder) Ayrık ödül dizisi r[1], r[2], … için kümülatif toplam G ile FTC'nin ilişkisi nedir?

Cevap: Prefix-sum $G[n] = \sum_{k \leq n} r[k]$, integralin ("birikim") ayrık karşılığıdır; ardışık fark $r[n] = G[n] - G[n-1]$ ise türevin ("anlık değer") karşılığıdır. FTC'nin ayrık akrabası: a'dan b'ye toplam = $G[b] - G[a]$. RL'de return bu G'dir; anlık ödül onun "türevidir". cumsum ↔ diff, integral ↔ türev ile birebir eşleşir.

8.9 Egzersizler

Egzersiz 1. $\int_1^4 (2t+1)dt$ integralini antitürevle hesapla.

Egzersiz 2. $d/dx[\int_0^x e^{-t^2} dt]$ nedir? (FTC birinci yarısını uygula. Not: e^{-t^2}'nin kapalı-form antitürevi yoktur — bu yüzden integral "erf" diye adlandırılır.)

Egzersiz 3. $\int_{-1}^1 x^3 dx$'i hesapla ve sonucun neden 0 olduğunu işaretli alanla açıkla.

Egzersiz 4. *(Python — sayısal + sembolik)* $v(t) = t(8-t)$ için sol-uç Riemann toplamını hesapla; N büyüdükçe $256/3 \approx 85{,}33$'e yakınsadığını göster.

```
N=    8  yaklasik=84.0000
N=   32  yaklasik=85.2500
N=  128  yaklasik=85.3281
N= 1024  yaklasik=85.3333
N= 8192  yaklasik=85.3333
kesin (SymPy): 256/3 = 256/3 ≈ 85.3333333333333
```

Egzersiz 5. *(Sonraki dersin habercisi)* Bir fonksiyonun bir aralıktaki **ortalama değeri** nedir ve bu, integralle nasıl ilişkilidir? Ders 9, alan ile eğim arasındaki bağı farklı bir açıdan derinleştirecek.

8.10 Anahtar Kavramlar (Cheat Sheet)

Kavram	Tanım	Grant'ta
İntegral = türevin tersi	Hızdan mesafe = antitürev bulmak	0m39
Mesafe = alan	Sabit hızda $v \cdot t$ = dikdörtgen alanı	3m05
Riemann toplamı	$v(t) \cdot dt$ ince dikdörtgenlerini topla	4m09
$\int$ **notasyonu**	$dt \to 0$ iken $\sum v(t)dt$ = eğri altı alan	6m36
FTC (birinci yarı)	$ds/dt = v$; alan fonksiyonunun türevi = grafik	11m09
Antitürev $(+C)$	Türevi f olan fonksiyon; sonsuz tane	12m32
FTC (ikinci yarı)	$\int_a^b f\,dx = F(b) - F(a)$	15m32
İşaretli alan	Eksenin altı negatif sayılır	18m50
$\int_0^8 t(8-t)dt = 256/3$	Toplam mesafe $\approx 85{,}33$ m	14m13

8.11 ML Bağlantıları Özeti

7 köprü

1. **İntegral = küçük niceliklerin toplamı** → beklenen değer $E[f(X)] = \int f \cdot p\,dx$, Monte Carlo entegrasyon, ELBO.
2. **FTC / antitürev** → CDF ↔ PDF ($F' = p$), RL value function ↔ anlık ödül; birikim ile anlık değer arasındaki bağ.
3. **"İki uç nokta yeter"** → prefix-sum / telescoping; bir aralığın toplamını $O(1)$'de almak.

4. **Riemann toplamı** → Monte Carlo entegrasyon; $N \to \infty$ iken kestirim gerçeğe yakınsar.
5. **İşaretli alan** → KL diverjansı terimleri, advantage (pozitif/negatif), net akış.
6. **Mesafe = alan (birim çarpımı)** → beklenti = (değer × ağırlık) alanı olarak okunur.
7. **cumsum** ↔ **diff** → ayrık integral/türev; dizi modellerinde birikimli özellikler.

! Tek bir şey alıp gideceksen

İntegral, bir eğri altındaki işaretli alandır — sayısız küçük dilimi $(v(t) \cdot dt)$ toplamanın $dt \to 0$ limiti. Calculus'un Temel Teoremi onu türevin tersine bağlar: alan fonksiyonunun türevi grafiğin kendisidir, dolayısıyla bir integrali, antitürevinin yalnızca iki uç noktasındaki değerinden hesaplayabilirsin.

9 Alanın Eğimle İlişkisi Nedir?

Sürekli ortalama, integral/genişlik ve FTC'nin ikinci yüzü

ℹ Bölüm bilgisi

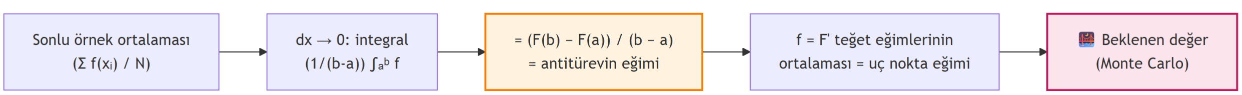
- **Grant'ın videosu:** YouTube — Chapter 9: What does area have to do with slope? (≈12 dk)
- **Kaynak:** 3Blue1Brown — Essence of Calculus
- **Okuma süresi:** ≈20 dk

9.1 Bu Derste Ne Var?

İntegralin sıkça karşımıza çıktığı bir problem türü: **sürekli bir değişkenin ortalaması**. Hem kendi başına faydalı, hem de integral ile türevin neden birbirinin tersi olduğuna — alanın neden eğimle ilişkili olduğuna — **yepyeni bir bakış** verir. Örneğimiz: sin(x)'in $[0, \pi]$ aralığındaki (yarım periyot) **ortalama yüksekliği**.

Üç ana fikir:

1. **Sürekli ortalama:** sonsuz değeri toplayıp ∞'a bölemezsin; çözüm integraldir. Ortalama = alan / genişlik = $(1/(b-a)) \int_a^b f$.
2. **Çözüm:** sin'in antitürevi $-\cos$; $\int_0^\pi \sin = 2$; ortalama = $2/\pi \approx 0{,}64$.
3. **Yeni bakış:** ortalama değer = antitürevin uç noktaları arasındaki **eğim** (rise/run); "ortalama eğim = toplam eğim". FTC'ye ikinci bir perspektif.



Şekil 9.1: FTC'nin ikinci yüzü: f'in ortalaması = antitürev F'in uç noktaları arası eğim.

> *"the average height of this graph is this area divided by its width."* — Grant, 5:14

Builder Notu — ML Köprüleri

- **Sürekli ortalama = integral / genişlik → beklenen değer**: bir aralıkta düzgün (uniform) dağılımda ortalama = $(1/(b-a)) \int f$; genel dağılımda $E[X] = \int x \cdot p(x) dx$.
- **"Sonlu ortalamayı sonsuz continuum'a genelle = integral"** sezgisi → olasılıkta beklenti, **Monte Carlo**; sürekli dağılımların tüm teorisi bu adımla kurulur.
- **Ortalama eğim = uç nokta eğimi** → ortalama değer teoremi; gradyanların bir yörünge boyunca

ortalaması; telescoping.
- **İşaretli alan / genişlik** → bir sinyalin ortalaması (DC bileşeni), bir kaybın bir epoch boyunca ortalaması.

9.2 Sürekli Bir Değişkenin Ortalaması

sin(x)'in 0 ile π arasındaki grafiğine bak (periyodunun yarısı). Bu aralıkta grafiğin **ortalama yüksekliği** nedir? Boş bir soru değil: dünyadaki bir sürü döngüsel olgu sinüs dalgalarıyla modellenir — örneğin güneşin gökte kaldığı saat sayısı, yılın gününe göre bir sinüs deseni izler.

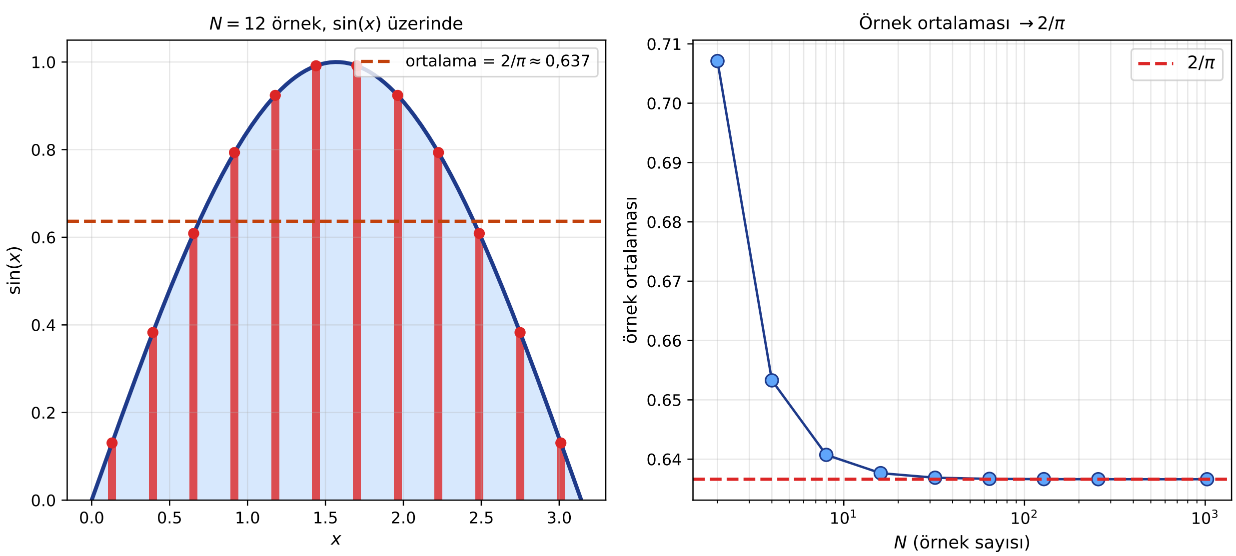
Tuhaf bir soru: ortalama denince genelde **sonlu** sayıda değer düşünürüz — hepsini topla, adetlerine böl. Ama $[0, \pi]$ arasında sin(x)'in **sonsuz** tane değeri var.

Builder Notu — Beklenen Değer

Bu "bir niceliğin ortalaması" sorusu, ML'de her yerdedir ama genelde **beklenen değer** adıyla. Bir kaybın veri dağılımı üzerindeki ortalaması $E[L]$, bir politikanın ortalama ödülü, bir aktivasyonun beklenen değeri — hepsi "sonsuz/çok sayıda değeri ortalama" sorusudur.

9.3 Sonsuz Değeri Ortalamak: Önce Sonlu Örnek

Aralık boyunca eşit aralıklı **sonlu sayıda nokta** örnekle. Sonlu örnek olduğu için ortalamayı alışıldık yolla bulabilirsin: her noktadaki yükseklikleri toplayıp örnek sayısına böl.



Şekil 9.2: Örnek ortalamasının integral ortalamasına yakınsaması: N büyüdükçe sin(x)'in $[0, \pi]$ örnek ortalaması $2/\pi \approx 0{,}637$'ye yaklaşır.

Builder Notu — Büyük Sayılar Yasası

"Önce sonlu örnekle yaklaştır, sonra örnek sayısını artır" — bu, **Monte Carlo** yönteminin tam tanımıdır. Bir beklentiyi hesaplayamadığında, dağılımdan N örnek al, ortalamalarını al; $N \to \infty$ iken bu, gerçek beklentiye yaklaşır (Büyük Sayılar Yasası).

9.4 Ortalama = Alan / Genişlik

Ortalama ifadesini (yüksekliklerin toplamı bölü örnek sayısı) dx cinsinden yeniden yazalım. Örnekler arası boşluk dx ise ve aralık 0'dan π'ye uzanıyorsa, örnek sayısı $\approx \pi/dx$:

$$\text{ortalama} \approx \frac{\sum \sin(x)}{\pi/dx} = \frac{1}{\pi}\sum \sin(x)\,dx \;\to\; \frac{1}{\pi}\int_0^{\pi} \sin(x)\,dx$$

dx'i paya dağıtınca topladığın terimler $\sin(x) \cdot dx$ oldu — yani pay tam bir **integral ifadesidir**. Daha çok nokta için bu ortalama, gerçek integralin aralık uzunluğuna (π) bölümüne yaklaşır. Başka deyişle: **ortalama yükseklik = alan / genişlik**.

Builder Notu — Beklenti Tanımı

Bu birebir **beklenen değerin** tanımıdır. Sonlu durumda $E[X] = \sum(\text{değer} \times \text{olasılık})$; $[a,b]$'de düzgün dağılımda her noktanın ağırlığı $dx/(b-a)$'dır, dolayısıyla $E[X] = (1/(b-a)) \int_a^b x\,dx$.

9.5 Çözüm: $-\cos$ Antitürevi ve $2/\pi$

İntegrali hesaplamak için $\sin(x)$'in **antitürevini** bulmalıyız: türevi sin olan fonksiyon. $-\cos x$'in türevi $+\sin x$'tir.

$$\int_0^{\pi} \sin(x)\,dx = [-\cos x]_0^{\pi} = -\cos\pi + \cos 0 = 1 + 1 = 2$$

sin grafiğinin $[0,\pi]$ altındaki alan tam olarak **2** — şık bir sonuç. O hâlde ortalama yükseklik:

$$\frac{1}{\pi}\int_0^{\pi} \sin(x)\,dx = \frac{2}{\pi} \approx 0{,}64$$

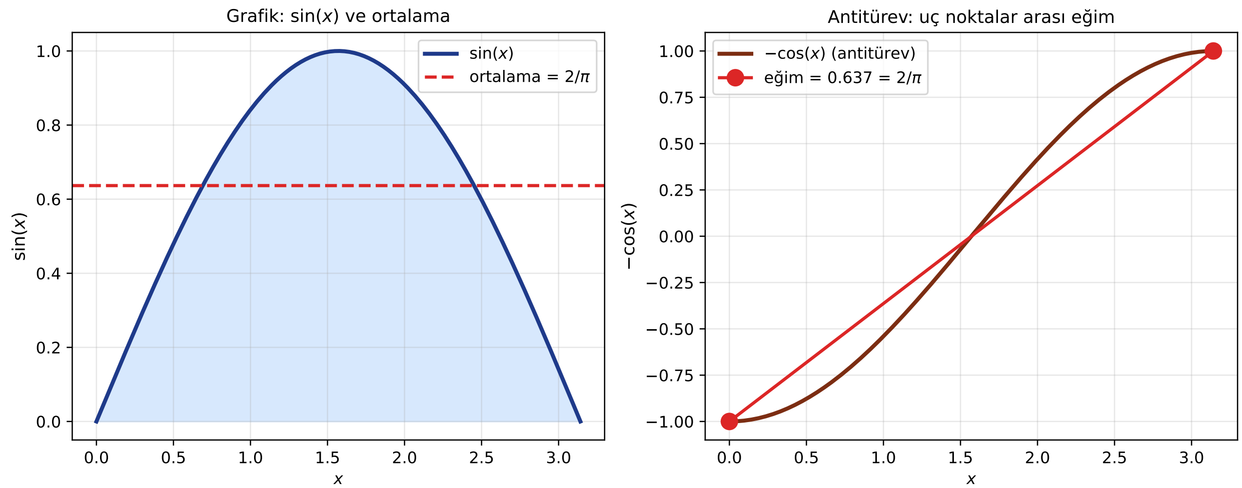
Builder Notu — Epoch Ortalama Kaybı

Bu $2/\pi \approx 0{,}64$ sonucu, sinyal işlemede yarım-dalga doğrultulmuş bir sinüsün ortalama değeri olarak karşına çıkar. ML'de bir aktivasyon veya kayıp sinyalinin bir aralıktaki ortalaması, tam bu "integral / genişlik" hesabıdır — örneğin bir epoch boyunca ortalama kayıp.

9.6 Yeni Bakış: Ortalama Eğim = Uç Noktalar Arası Eğim

Ortalama değer $2/\pi$'ye, antitürev $(-\cos x)$'in girdi aralığındaki **değişiminin**, aralık uzunluğuna bölümüyle ulaştık:

$$\frac{2}{\pi} = \frac{(-\cos\pi) - (-\cos 0)}{\pi - 0}$$



Şekil 9.3: sin'in $[0, \pi]$ ortalaması $(2/\pi)$ = antitürev $-\cos$'un iki uç nokta arası eğimi. f'nin teğet eğimlerinin ortalaması = F'in toplam eğimi.

Bu kesri başka türlü oku: $-\cos$ grafiğinin π üzerindeki noktası ile 0 üzerindeki noktası arasındaki **rise/run eğimi**. Peki bu eğim neden $\sin(x)$'in ortalama değerini temsil etsin? Çünkü tanım gereği $\sin(x)$, $-\cos$ grafiğinin **türevidir** — yani her noktadaki teğet eğimidir.

> *"the average slope of a graph over all of its points in a certain range should equal the total slope between the start and end points."* — Grant, 8:13

Genel olarak, $f(x)$'in $[a, b]$'deki ortalama değeri:

$$\frac{1}{b-a}\int_a^b f(x)\,dx = \frac{F(b) - F(a)}{b-a}$$

Builder Notu — Telescoping

"f'nin ortalama değeri = F'nin uç noktaları arası eğimi" ifadesi, **ortalama değer teoreminin** integral hâlidir ve telescoping'in sürekli karşılığıdır. ML'de bir gradyanın bir yörünge boyunca ortalaması, yalnızca başlangıç ve bitiş parametrelerinin farkına bağlıdır (ara yol önemsiz) — tıpkı $F(b) - F(a)$'nın aradaki tüm noktaları görmezden gelmesi gibi.

9.7 İkinci Sezgi: Sonlu → Sonsuz Genelleme = İntegral

Ders 8'de integralleri akla getiren bir his tarif etmiştim: "problem küçük şeyleri toplayarak yaklaştırılabiliyorsa, integral düşün." Burada **ikinci bir his** ekliyoruz: bir fikri **sonlu** bir bağlamda anlıyorsan ve o fikir birden çok değeri toplamayı içeriyorsa (bir sayı kümesinin ortalamasını almak gibi), ve bunu **sonsuz, sürekli** bir değer aralığına genellemek istiyorsan — onu bir integral cinsinden ifade etmeyi dene.

> *"if you want to generalize that idea to apply to an infinite continuous range of values, try seeing if you can phrase things in terms of an integral."* — Grant, 11:58

Builder Notu — Olasılık Teorisi

Bu "sonlu fikri sürekliye taşı = integral" sezgisi, **olasılık teorisinin** kurucu adımıdır. Kesikli bir dağılımın beklentisi $\sum x \cdot p(x)$; sürekliye geçince $\int x \cdot p(x)dx$ olur. Varyans, entropi, KL diverjansı — hepsi önce sonlu toplamla tanımlanır, sonra integralle sürekliye genellenir.

9.8 Bu Dersin Özeti

1. Sürekli bir değişkenin ortalaması: sonsuz değer var, hepsini toplayıp ∞'a bölemezsin — çözüm bir **integraldir**.
2. Önce sonlu örnekle yaklaş: yükseklikleri topla, örnek sayısına böl. Örnek sayısı $\approx \pi/dx$.
3. Yeniden yaz: ortalama $\approx (\sum \sin(x) \cdot dx)/\pi \to (1/\pi)\int_0^\pi \sin(x)dx$ = **alan / genişlik**.
4. Çözüm: sin'in antitürevi $-\cos$; $\int_0^\pi \sin dx = 2$; ortalama = $2/\pi \approx 0{,}64$.
5. Yeni bakış: bu ortalama, antitürev $-\cos$'un uç noktaları arasındaki **rise/run eğimidir**.
6. $f = F'$ olduğundan, f'nin ortalama değeri = F'nin teğet eğimlerinin ortalaması = uç noktalar arası eğim.
7. İkinci sezgi: sonlu bir "toplama/ortalama" fikrini sürekli, sonsuz bir aralığa genellemek istersen, onu bir integral cinsinden ifade et.

! Tek bir cümle

Sürekli bir fonksiyonun bir aralıktaki ortalaması, integralinin (işaretli alan) genişliğe bölümüdür — ve bu, antitürevin uç noktaları arasındaki eğime eşittir; yani "ortalama değer = ortalama eğim = uç noktalar arası eğim", alanın eğimle ilişkisine ikinci bir kanıttır.

9.9 Kontrol Soruları

i Soru 1: f(x) = x^2'nin [0, 3] aralığındaki ortalama değeri nedir?

Cevap: Ortalama = $(1/(3-0))\int_0^3 x^2dx$. Antitürev $x^3/3$, dolayısıyla $\int_0^3 x^2dx = [x^3/3]_0^3 = 27/3 = 9$. Ortalama = $9/3 = 3$. (Sezgi: x^2 eğrisi 0'dan 9'a çıkar; ortalama 3, yani maksimumun üçte biri.)

> ℹ Soru 2: 'Ortalama = alan / genişlik' birimler açısından neden mantıklı?
>
> **Cevap:** $\int f\,dx$'in birimi, (f'nin birimi) × (x'in birimi) = bir "alan" birimidir. Bunu genişliğe (x'in birimi) bölünce geriye yalnızca f'nin birimi kalır — yani bir **yükseklik/değer** birimi.

> ℹ Soru 3: f'nin ortalama değeri neden antitürev F'nin uç noktaları arası eğimine eşittir?
>
> **Cevap:** Ortalama = $(1/(b-a))\int_a^b f$. FTC ile $\int_a^b f = F(b) - F(a)$, dolayısıyla ortalama = $(F(b) - F(a))/(b-a)$ — bu zaten F'nin iki uç nokta arasındaki rise/run eğimi. $f = F'$ olduğundan f, F'nin her noktadaki teğet eğimidir; bu eğimlerin ortalaması da uç noktalar arası toplam eğime eşittir.

> ℹ Soru 4: (Builder) Uniform(a, b) dağılımının beklenen değeri, 'ortalama = alan/genişlik' ile nasıl çıkar?
>
> **Cevap:** Düzgün dağılımda her nokta eşit ağırlıklı, yani $E[X] = x$'in $[a,b]$'deki ortalama değeri = $(1/(b-a))\int_a^b x\,dx$. $\int_a^b x\,dx = (b^2-a^2)/2$, dolayısıyla $E[X] = (b^2-a^2)/(2(b-a)) = (a+b)/2$ — aralığın orta noktası.

9.10 Egzersizler

Egzersiz 1. cos(x)'in $[0, \pi/2]$ aralığındaki ortalama değerini bul.

Egzersiz 2. $f(x) = x$'in $[0, L]$ aralığındaki ortalama değerini integralle bul. Sonucun $L/2$ olduğunu göster.

Egzersiz 3. sin(x)'in $[0, 2\pi]$ aralığındaki ortalama değerini hesapla ve sonucun neden 0 olduğunu **işaretli alanla** açıkla.

Egzersiz 4. *(Python — sayısal + sembolik)* sin'in $[0, \pi]$ ortalamasını sonlu örnekle yaklaştır.

```
N=    4  ornek_ortalama=0.60355
N=   16  ornek_ortalama=0.63457
N=   64  ornek_ortalama=0.63649
N=  256  ornek_ortalama=0.63661
N= 1024  ornek_ortalama=0.63662
2/pi = 0.6366197723675814
kesin (integral/genislik): 2/pi
```

Egzersiz 5. *(Sonraki dersin habercisi)* Bu seride hep **birinci** türevi (eğim, hız) kullandık. Türevin türevi — **ikinci türev** — ne anlatır? Ders 10, yüksek mertebeden türevleri ele alacak.

9.11 Anahtar Kavramlar (Cheat Sheet)

Kavram	Tanım	Grant'ta
Sürekli ortalama	Sonsuz değer; toplayıp ∞'a bölemezsin $\to$ integral	1m43
Sonlu örnekle yaklaş	Yükseklikleri topla/say; örnek sayısı $\approx$ aralık/dx	2m17
Ortalama = alan / genişlik	$(1/(b-a)) \int_a^b f\,dx$	5m14
$\int_0^\pi$ sin $dx = 2$	sin'in $[0,\pi]$ altındaki alan	6m37
Ortalama sin $= 2/\pi$	$\approx$ 0,64; alan/genişlik	6m50
Ortalama değer = uç nokta eğimi	$(F(b)-F(a))/(b-a)$	7m19
Ortalama eğim = toplam eğim	$f = F'$ teğetlerinin ortalaması	8m13
Sonlu $\to$ sonsuz = integral	İkinci "integral düşün" sezgisi	11m58

9.12 ML Bağlantıları Özeti

7 köprü

1. **Sürekli ortalama = integral / genişlik** $\to$ beklenen değer $E[X] = \int x \cdot p(x)dx$; bir aralıkta ortalama kayıp.
2. **Sonlu örnek $\to$ integral** $\to$ Monte Carlo entegrasyon, Büyük Sayılar Yasası.
3. **Ortalama = uç nokta eğimi** $\to$ ortalama değer teoremi, telescoping; bir yörüngede net ilerleme yalnızca uçlara bağlı.
4. **"Sonlu fikri sürekliye genelle = integral"** $\to$ olasılık teorisi: beklenti, varyans, entropi, KL diverjansı.
5. **İşaretli alan ortalaması** $\to$ net/DC bileşen; pozitif ve negatif katkıların dengesi.
6. **Uniform beklenti** $(a+b)/2$ $\to$ düzgün dağılım, ağırlık başlatma aralıkları.
7. **Epoch ortalama kaybı** $\to$ "integral / genişlik" hesabının doğrudan pratik karşılığı.

! Tek bir şey alıp gideceksen

Sürekli bir şeyin ortalaması = integral (işaretli alan) bölü genişlik. Ve bu, antitürevin uç noktaları arasındaki eğime eşittir — alanın eğimle ilişkisinin ikinci kanıtı. Sonlu bir "ortalama/toplam" fikrini sonsuz bir continuum'a taşımak istediğinde refleksin "integral" olsun; bu, olasılık ve beklenti teorisinin kapısıdır.

10 Yüksek Mertebeden Türevler

Eğrilik, ivme ve jerk — Taylor'a kapı

ℹ Bölüm bilgisi

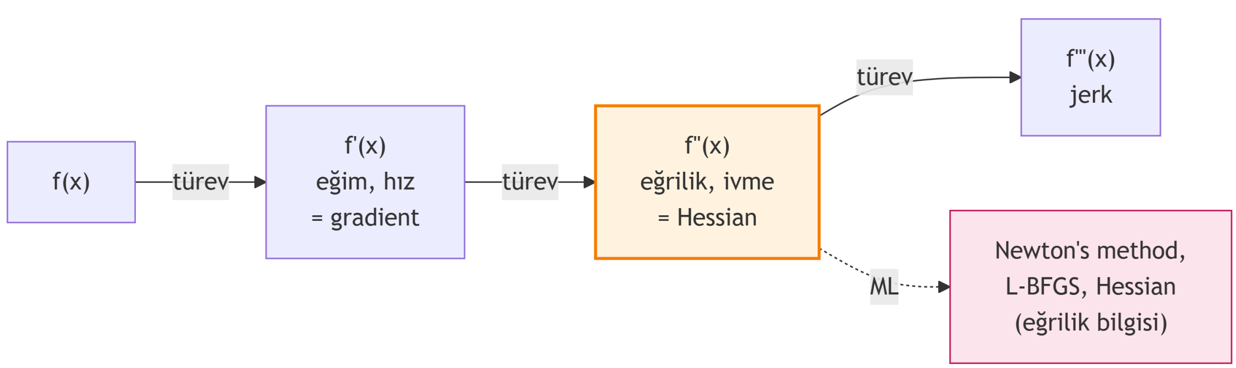
- **Grant'ın videosu:** YouTube — Chapter 10: Higher order derivatives (≈5 dk)
- **Kaynak:** 3Blue1Brown — Essence of Calculus
- **Okuma süresi:** ≈12 dk

10.1 Bu Derste Ne Var?

Bu kısa bölüm, Ders 11'deki **Taylor serilerine** geçmeden önce bir dipnot: **yüksek mertebeden türevler**, özellikle **ikinci türev**. İkinci türev, türevin türevidir — yani eğimin nasıl değiştiğini söyler. Geometrik karşılığı **eğrilik**, fiziksel karşılığı **ivme**dir.

Üç ana fikir:

1. **İkinci türev = türevin türevi;** eğimin değişim oranı. Notasyon: $f''(x) = d^2f/dx^2$.
2. **Geometrik:** eğrilik. Yukarı bükey (◡) → $f'' > 0$; aşağı bükey (◠) → $f'' < 0$.
3. **Fiziksel:** ivme (mesafe → hız → ivme). Üçüncü türev = **jerk**.



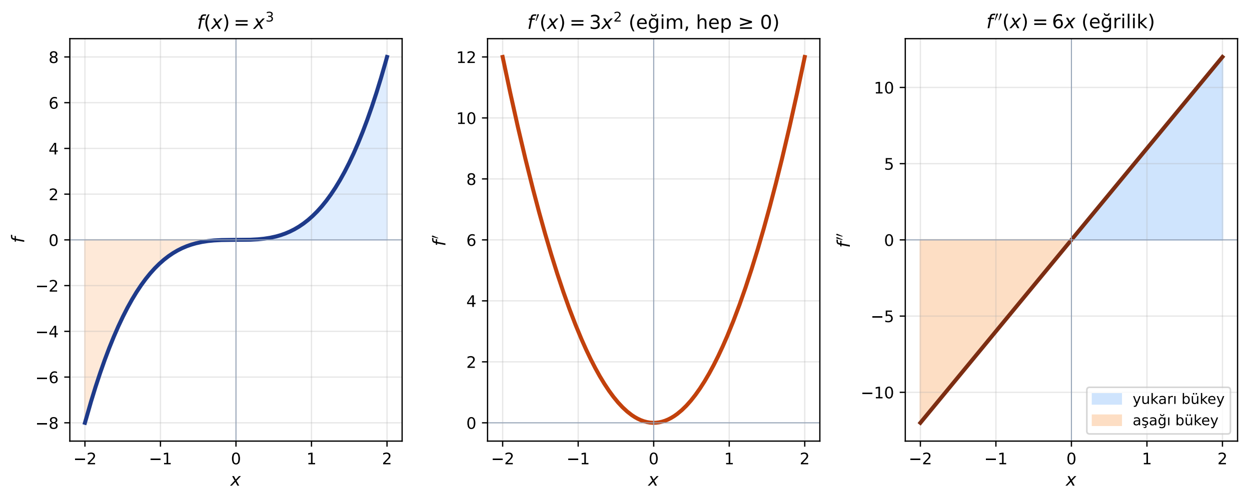
Şekil 10.1: Türev mertebeleri ve ML eşdeğerleri: gradient (1.) + Hessian (2.) + jerk (3.).

> *"the most visceral understanding of the second derivative is that it represents acceleration."* — Grant, 4:01

> **Builder Notu — ML Köprüleri**
>
> - **İkinci türev → Hessian.** Çok değişkenli halde ikinci türevler bir matris (Hessian) oluşturur; loss landscape'in **eğriliğini** kodlar. Newton's method ve ikinci-derece optimizer'lar (L-BFGS) bunu kullanır.
> - **Eğrilik işareti → konvekslik.** $f'' > 0$ her yerde ise fonksiyon konveks (tek minimum); **ikinci türev testi** kritik noktanın min mi ($f'' > 0$) max mı ($f'' < 0$) olduğunu söyler.
> - **Düz vs keskin minimum** → düşük eğrilikli (flat) minimumlar daha iyi genelleme yapar; sharpness-aware minimization (SAM) tam bunu hedefler.
> - **İvme analojisi** → SGD **momentum**: gradyan "kuvvet", güncelleme "hız"; momentum, ikinci-derece dinamik bir sezgi katar.

10.2 İkinci Türev: Türevin Türevi

Bir $f(x)$ fonksiyonunda türev, grafiğin bir nokta üzerindeki **eğimidir**: dik eğim büyük türev, aşağı eğim negatif türev. **İkinci türev** ise türevin türevidir — yani o eğimin **nasıl değiştiğini** söyler.



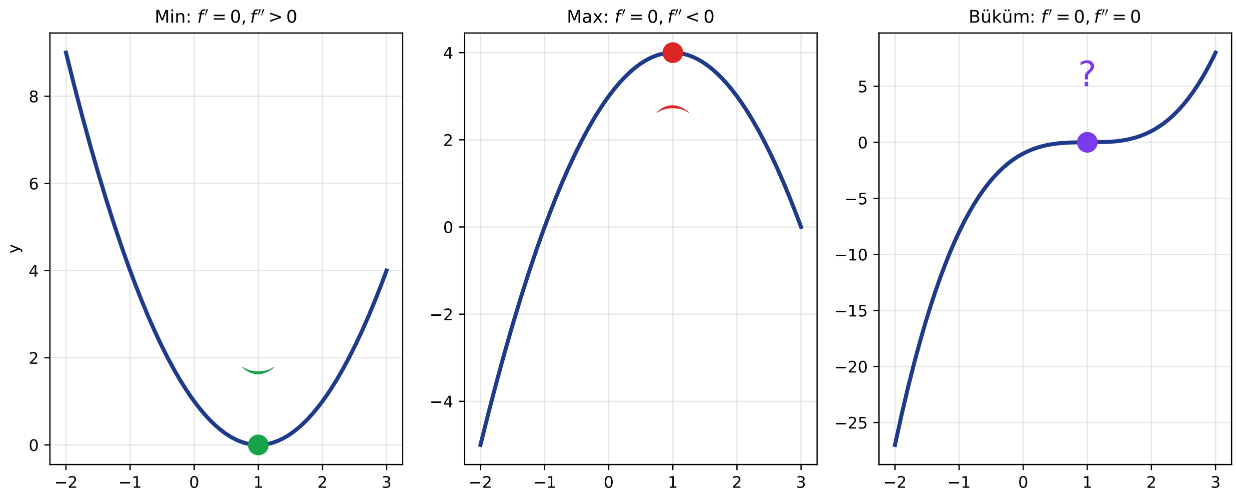
Şekil 10.2: $f(x) = x^3$, $f'(x) = 3x^2$, $f''(x) = 6x$. $x = 0$'da büküm noktası ($f'' = 0$, işaret değiştirir); $x > 0$ yukarı bükey, $x < 0$ aşağı bükey.

Bunu bir bakışta görmenin yolu, $f(x)$ grafiğinin nasıl **büküldüğüne** bakmaktır. Grafiğin yukarı büküldüğü noktalarda eğim artıyordur, yani ikinci türev pozitiftir. Aşağı büküldüğü noktalarda eğim azalıyordur, ikinci türev negatiftir.

> **Builder Notu — Newton's Method**
>
> Gradient descent yalnızca birinci türevi (eğim) kullanır; nereye gideceğini bilir ama "zemin ne kadar bükülüyor" bilmez. İkinci türev (Hessian) bu eksik bilgiyi verir: Newton's method, adımı eğriliğe göre ölçekler ve çok daha hızlı yakınsayabilir.

10.3 Eğrilik: Yukarı/Aşağı Bükey

Eğriliğin şiddeti de önemli. Bir noktada eğim **hızla** artıyorsa, ikinci türev **çok** pozitiftir. Hiç eğrilik olmayan (doğrusal) noktalarda ise ikinci türev tam **0**'dır.



Şekil 10.3: İkinci türev testi: kritik noktada $f' = 0$. $f'' > 0$ ise yerel min; $f'' < 0$ ise yerel max; $f'' = 0$ ise belirsiz (büküm).

İkinci türevin işareti, grafiğin yerel şeklini anlatır: pozitif ise vadi ($\smile$), negatif ise tepe ($\frown$). Bu, bir kritik noktanın (eğimin 0 olduğu yer) minimum mu maksimum mu olduğunu ayırt etmenin anahtarıdır.

> **Builder Notu — Eyer Noktaları**
>
> Eğrilik, optimizasyonun kalbidir. Bir kaybın minimumunda gradyan 0'dır; ama orada ikinci türeve bakarak minimum ($f'' > 0$), maksimum ($f'' < 0$) ya da eyer noktası (karışık) ayırt edilir. Yüksek boyutta bu, Hessian'ın özdeğerlerinin işaretlerine bakmaktır — derin ağların kayıp yüzeylerinin neden eyer noktalarıyla dolu olduğunu da bu açıklar.

10.4 Notasyon: d^2f/dx^2

İkinci türevi şöyle yazarız:

$$f''(x) = \frac{d}{dx}\left(\frac{df}{dx}\right) = \frac{d^2f}{dx^2}$$

Bir girdiden başla ve sağa, her biri dx boyutunda **iki küçük adım** at. İlk adım fonksiyonda bir df_1 değişimine, ikinci adım benzer ama biraz farklı bir df_2 değişimine yol açar. Bu iki değişim arasındaki fark ddf'tir. Bunu çok küçük düşün; tipik olarak $(dx)^2$ ile orantılıdır:

$$f''(x) \approx \frac{ddf}{(dx)^2} = \frac{df_2 - df_1}{(dx)^2}$$

Builder Notu — Taylor'ın İkinci Terimi

$(dx)^2$ ile orantılı olan ddf, Ders 2-3'te **attığımız** terimdi. İkinci türev tam olarak o atılan terimi yakalar — bu yüzden bir fonksiyonu birinci derece (teğet doğru) yerine **ikinci derece** (parabol) ile yaklaştırmak istediğinde ikinci türev devreye girer.

10.5 İvme ve Jerk

İkinci türevin en somut anlamı **ivmedir**. Bir doğru üzerindeki harekette:

$$s(t) \;\to\; v(t) = \frac{ds}{dt} \;\to\; a(t) = \frac{d^2 s}{dt^2}$$

İkinci türev pozitifse hızlanma vardır; negatifse yavaşlama. Üçüncü türev — şaka değil — **jerk** (sarsıntı) diye adlandırılır.

> *"The third derivative, and this is not a joke, is called jerk."* — Grant, 4:54

Builder Notu — SGD Momentum

Mesafe → hız → ivme zinciri, SGD **momentum**unun sezgisidir: gradyan bir "kuvvet" gibi davranır, parametre güncellemesi bir "hız" biriktirir, böylece optimizasyon düz bölgelerde ivmelenir ve gürültüyü yumuşatır. Adam gibi optimizer'lar gradyanın birinci momentini (hız) ve ikinci momentini (ölçek) ayrı ayrı izleyerek bu fiziksel analojiyi daha da ileri taşır.

10.6 Bu Dersin Özeti

1. İkinci türev = türevin türevi; eğimin değişim oranıdır. Notasyon: $f''(x) = d^2 f/dx^2$.
2. Geometrik anlam eğriliktir: yukarı bükey ($\smile$) → $f'' > 0$; aşağı bükey ($\frown$) → $f'' < 0$; eğrilik yoksa $f'' = 0$.
3. Eğriliğin şiddeti = ikinci türevin büyüklüğü (hızlı bükülme → büyük f'').
4. Notasyon mantığı: iki dx adımı al, $ddf = df_2 - df_1 \propto (dx)^2$; $f'' = ddf/(dx)^2$.
5. Fiziksel anlam ivmedir: $a(t) = d^2 s/dt^2$. Üçüncü türev = jerk.
6. Yüksek mertebeden türevler, fonksiyonları yaklaştırmanın (Taylor serisi, Ders 11) anahtarıdır.

! Tek bir cümle

İkinci türev, türevin türevidir — eğimin nasıl değiştiğini, yani grafiğin eğriliğini (ve hareketin ivmesini) ölçer; $d^2 f/dx^2$ notasyonundaki "kareler", birinci türevde atılan $(dx)^2$ terimini yakalamasından gelir ve bu, fonksiyonları ikinci dereceden yaklaştırmanın temelidir.

10.7 Kontrol Soruları

Soru 1: f(x) = x^3 için ikinci türev nedir? Eğriliği nasıl yorumlarsın?

Cevap: $f'(x) = 3x^2$, $f''(x) = 6x$. $x > 0$'da $f'' > 0$ (yukarı bükey), $x < 0$'da $f'' < 0$ (aşağı bükey), $x = 0$'da $f'' = 0$ (büküm noktası — eğrilik yön değiştirir). Bu yüzden x^3 grafiği S şeklindedir.

Soru 2: Bir kritik noktada (f' = 0) ikinci türevin işareti ne söyler?

Cevap: $f'' > 0$ ise yerel **minimum** (vadi); $f'' < 0$ ise yerel **maksimum** (tepe); $f'' = 0$ ise test belirsizdir. Buna **ikinci türev testi** denir — gradyan sıfırken minimum/maksimum ayrımının anahtarı.

Soru 3: Bir araba s(t) = t^2 ile hareket ediyor. Hızı ve ivmesi nedir?

Cevap: Hız $v = ds/dt = 2t$ (zamanla doğrusal artar). İvme $a = d^2s/dt^2 = 2$ (sabit, pozitif). Yani araba sabit ivmeyle sürekli hızlanır — serbest düşüş gibi.

Soru 4: (Builder) Bir loss yüzeyinde Hessian'ın tüm özdeğerleri pozitifse ne anlama gelir? Eyer noktası nedir?

Cevap: Tüm özdeğerler pozitif (gradyan da 0) ise nokta yerel **minimumdur** — her yönde yukarı bükey. **Eyer noktası**, bazı özdeğerlerin pozitif bazılarının negatif olduğu yerdir: bir yönde minimum, başka bir yönde maksimum gibi. Gradyan 0'dır ama minimum değildir. Yüksek boyutlu derin ağlarda kritik noktaların çoğu minimumdan ziyade eyer noktasıdır.

10.8 Egzersizler

Egzersiz 1. $f(x) = x^4$'ün birinci, ikinci ve üçüncü türevlerini bul ($4x^3$, $12x^2$, $24x$).

Egzersiz 2. $f(x) = \sin(x)$'in ikinci türevini bul. (İpucu: $d(\sin)/dx = \cos$, $d(\cos)/dx = -\sin \to f'' = -\sin x$. sin, iki türevde kendi negatifine döner.)

Egzersiz 3. Bir grafiğin **büküm noktası**, eğriliğin yön değiştirdiği yerdir: $f'' = 0$ ve işaret değiştirir. $f(x) = x^3$ için büküm noktasını bul.

Egzersiz 4. *(Python — sayısal doğrulama)* $f(x) = x^3$ için ikinci türevi merkezi-fark formülüyle $[f(x+h) - 2f(x) + f(x-h)]/h^2$ hesapla.

```
sayisal f'' = 12.0   teorik 6x = 12.0
```

Egzersiz 5. *(Sonraki dersin habercisi)* Bir fonksiyonu bir nokta civarında polinomla yaklaştır: birinci türev sana en iyi **teğet doğruyu**, ikinci türev en iyi **parabolü** verir. Ders 11, **Taylor serilerini** anlatacak.

10.9 Anahtar Kavramlar (Cheat Sheet)

Kavram	Tanım	Grant'ta
İkinci türev	Türevin türevi; eğimin değişim oranı	0m57
d^2f/dx^2 **notasyonu**	İki dx adımı; $ddf/(dx)^2$	2m19
Yukarı bükey → $f'' > 0$	Eğim artıyor (vadi $\smile$)	1m08
Aşağı bükey → $f'' < 0$	Eğim azalıyor (tepe $\frown$)	1m17
$f'' = 0$	Eğrilik yok; doğrusal nokta veya büküm	1m46
İvme = d^2s/dt^2	Hızın değişim oranı	4m01
İkinci türev testi	Kritik noktada $f'' > 0$ min, $f'' < 0$ max	1m26
Jerk	Üçüncü türev; ivmenin değişimi	4m54

10.10 ML Bağlantıları Özeti

7 köprü

1. **İkinci türev → Hessian** → loss landscape'in eğriliği; ikinci-derece bilgi.
2. **Eğrilik işareti → konvekslik** → ikinci türev testi (min / max / eyer ayrımı).
3. **Newton's method / L-BFGS** → ikinci-derece optimizer; adımı eğriliğe göre ölçekler.
4. **Düz vs keskin minimum** → düşük eğrilikli (flat) minimumlar daha iyi genelleme; SAM.
5. **İvme → SGD momentum** → gradyan "kuvvet", güncelleme "hız"; Adam birinci+ikinci moment.
6. $(dx)^2$ **terimi** → ikinci dereceden (parabol) yaklaşım; Taylor'ın ikinci terimi.
7. **Eyer noktaları** → yüksek boyutlu loss yüzeylerinin baskın kritik noktaları (Hessian özdeğer işaretleri).

! Tek bir şey alıp gideceksen

İkinci türev, eğimin nasıl değiştiğidir — geometrik olarak eğrilik, fiziksel olarak ivme. İşareti bir kritik noktanın minimum mu maksimum mu olduğunu söyler (optimizasyonun kalbi), büyüklüğü ise bir fonksiyonu ne kadar iyi bir parabolle yaklaştırabileceğini. Bu da bizi doğrudan Taylor serilerine götürür.

11 Taylor Serileri

Tek nokta türev bilgisini civar davranışa çevirme

Bölüm bilgisi

- **Grant'ın videosu:** YouTube — Chapter 11: Taylor series (≈22 dk)
- **Kaynak:** 3Blue1Brown — Essence of Calculus
- **Okuma süresi:** ≈28 dk

11.1 Bu Derste Ne Var?

Taylor serileri, matematiğin fonksiyon yaklaşımı için sunduğu en güçlü araçlardan biridir. Temel fikir: polinom-olmayan bir fonksiyonu, bir nokta civarında **polinomlarla** yaklaştırmak — çünkü polinomları hesaplamak, türevlemek ve integrallemek çok daha kolaydır.

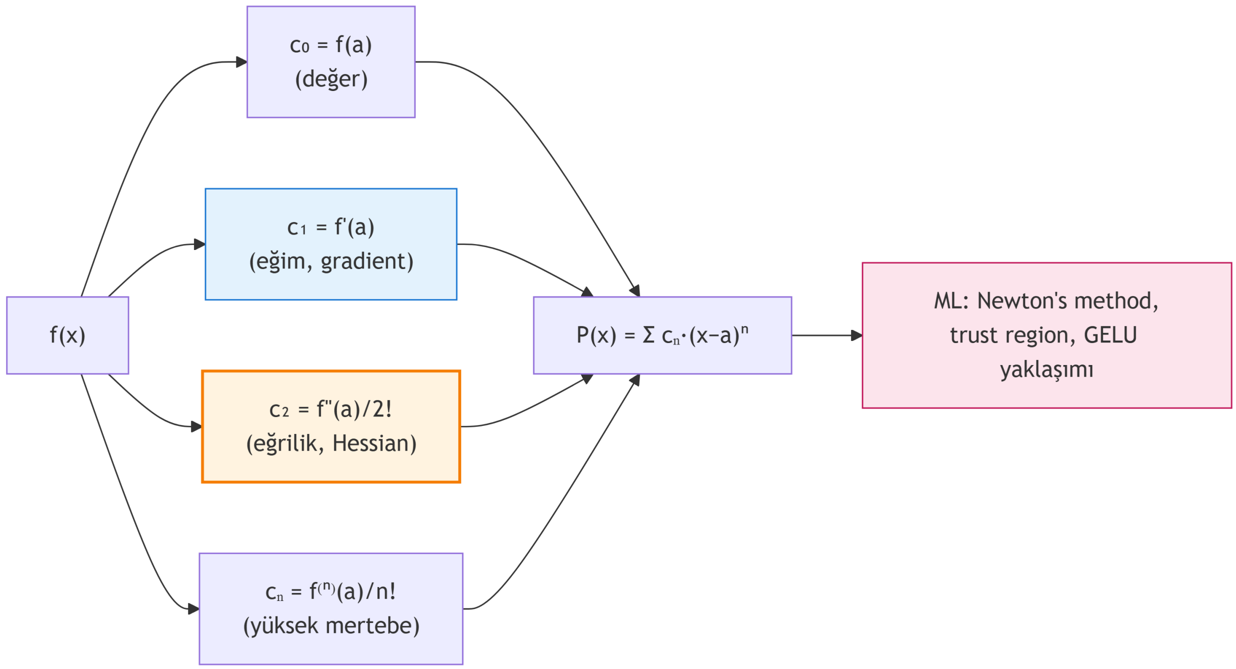
Üç ana fikir:

1. **Taylor polinomu:** bir fonksiyonu bir noktada polinomla yaklaştır; polinomun katsayıları, türevleri o noktada **eşleştirir** (değer, eğim, eğrilik…).
2. **n. terimin katsayısı =** $f^{(n)}(0)/n!$ — faktöriyel, kuvvet kuralının kademeli etkisini götürür.
3. **Taylor serisi** (sonsuz terim): bazen **her yerde** yakınsar (e^x, sin, cos), bazen yalnızca bir **yakınsama yarıçapı** içinde.

> *"they translate derivative information at a single point to approximation information around that point."* — Grant, 21:33

Builder Notu — ML Köprüleri

- **Birinci-derece Taylor** → gradient / **lineerleştirme**: $f(x) \approx f(a) + f'(a)(x-a)$; gradient descent adımı tam bu yerel lineer modeldir.
- **İkinci-derece Taylor → Hessian, Newton's method**: $f(x) \approx \ldots + \frac{1}{2}f''(a)(x-a)^2$; trust region, doğal gradyan ve ikinci-derece optimizer'lar bu kuadratik modeli kullanır.
- e^x **serisi** → softmax/exp hesabı; **GELU**'nun tanh/erf ile yaklaşımı, sigmoid açılımları.
- **"Tek noktadaki türev bilgisi → civardaki davranış"** → yerel / surrogate modeller, RL ve kontrolde sistemi bir nokta civarında lineerleştirme.
- **Yakınsama yarıçapı** → bir yaklaşımın geçerli kaldığı bölge; sayısal serilerin ve aktivasyon yaklaşımlarının sınırı.



Şekil 11.1: Taylor zinciri: değer + eğim + eğrilik + … = giderek daha iyi polinom yaklaşımı.

11.2 Neden Taylor? Polinomla Yaklaşmak

Grant'ın bu fikri ilk kavradığı an bir fizik dersinde olmuş: bir sarkacın potansiyel enerjisi $1 - \cos(\theta)$ ile orantılıydı ve bu cosine ifadesi problemi hantal yapıyordu. Ama $\cos(\theta)$'yı $1 - \theta^2/2$ ile yaklaştırınca her şey yerine oturdu.

Motivasyon, polinomların diğer fonksiyonlardan çok daha **uysal** olması: kolay hesaplanır, kolay türevlenir, kolay integrallenir.

Builder Notu — Yerel Polinom

ML'de bir fonksiyonu yerel olarak basit bir modelle değiştirmek her yerdedir. Gradient descent, kaybı bir noktada **birinci-derece** (lineer) Taylor ile değiştirir ve o yönde adım atar. Newton's method ve trust-region yöntemleri **ikinci-derece** (kuadratik) Taylor kullanır.

11.3 $\cos(x)$'i Parabolle Yaklaştırmak

$\cos(x)$'i $x = 0$ civarında bir **kuadratikle** yaklaştıralım:

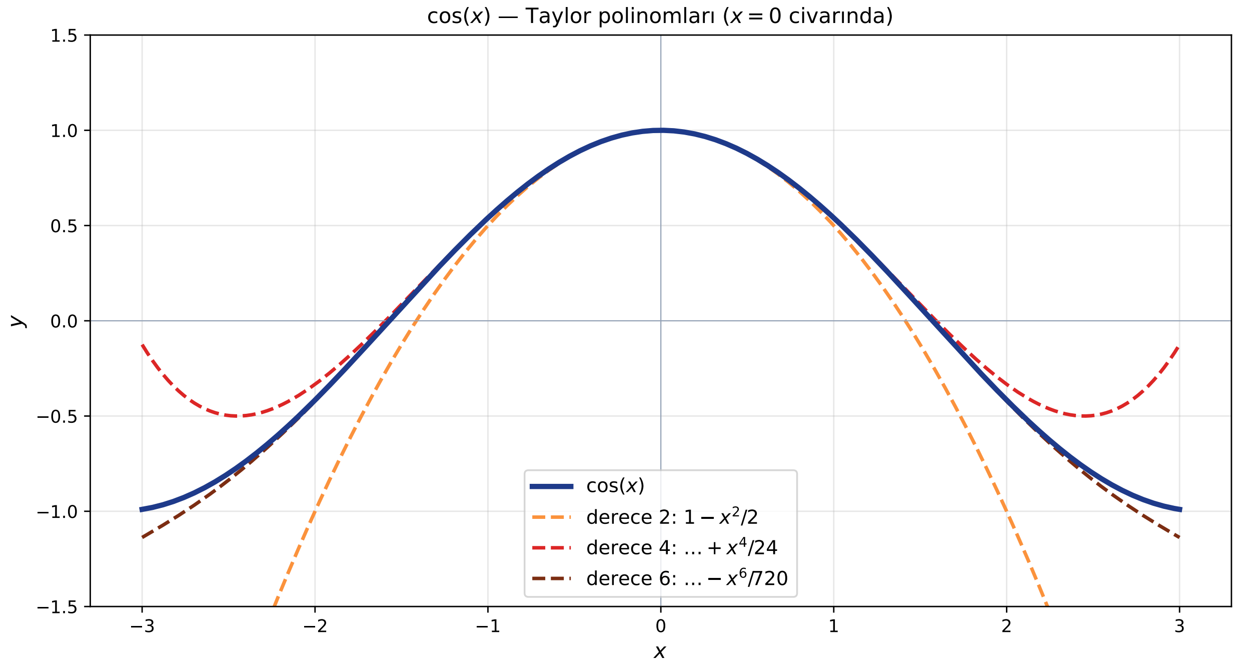
$$P(x) = c_0 + c_1 x + c_2 x^2$$

Üç koşulu sırayla dayatıyoruz:

- **Değer eşleşsin:** $\cos(0) = 1$. $P(0) = c_0$, dolayısıyla $c_0 = 1$.

- **Eğim eşleşsin:** $\cos' = -\sin$, $x = 0$'da 0. $P'(x) = c_1 + 2c_2x$, $x = 0$'da c_1. Dolayısıyla $c_1 = 0$.
- **Eğrilik eşleşsin:** $\cos'' = -\cos$, $x = 0$'da -1. $P''(x) = 2c_2$. Dolayısıyla $c_2 = -1/2$.

$$\cos(x) \approx 1 - \frac{1}{2}x^2$$



Şekil 11.2: $\cos(x)$'in Taylor polinomları: dereceden derece daha iyi yaklaşım. $x = 0$ civarında 4. derece çok küçük hatayla uyar.

$\cos(0{,}1)$'i bu polinomla tahmin edersen $0{,}995$ çıkar — ve $\cos(0{,}1)$'in gerçek değeri de $0{,}995$. Üç serbestlik derecesi sırasıyla **değeri**, **eğimi** ve **eğriliği** cos ile eşleştirdi.

> Builder Notu — Newton's Method
>
> Bu "değer + eğim + eğrilik eşleştirme" tam olarak ikinci-derece optimizasyonun yaptığıdır. Newton's method, kaybı bir noktada bu üç bilgiyle (f, f', f'') kuadratik bir parabolle değiştirir ve doğrudan o parabolün minimumuna atlar.

11.4 Daha Çok Terim: Faktöriyeller

Daha çok serbestlik için terim ekle. c_3x^3 eklersen: bir kübiğin üçüncü türevi $1 \cdot 2 \cdot 3 \cdot c_3 = 6c_3$'tür. cos'un üçüncü türevi $\sin x$, $x = 0$'da 0. Eşleşmesi için $c_3 = 0$.

c_4x^4 eklersen iyileşme olur. cos'un dördüncü türevi yine cos'tur, $x = 0$'da 1. Polinomun dördüncü türevi $1 \cdot 2 \cdot 3 \cdot 4 \cdot c_4 = 24c_4$:

$$c_4 = \frac{\cos^{(4)}(0)}{4!} = \frac{1}{24}$$

Faktöriyeller doğal olarak çıkar: x^n'in n ardışık türevini alınca kuvvet kuralı kademeli iner ve geriye $1 \cdot 2 \cdot 3 \cdots n = n!$ kalır.

> Builder Notu — exp Serisi
>
> x^n'in türevlerinden çıkan $n!$ faktörü, neden Taylor katsayılarında ve dolayısıyla exp/softmax serilerinde her yerde faktöriyel gördüğünü açıklar. Bir framework e^x'i hesaplarken bu $1/n!$ katsayıları kullanır; faktöriyel hızlı büyüdüğü için seri hızlı yakınsar.

11.5 Genel Taylor Formülü ve e^x

Genel olarak, herhangi bir f için x^n teriminin katsayısı:

$$f(x) \approx \sum_{n=0}^{N} \frac{f^{(n)}(0)}{n!} x^n$$

0 yerine başka bir a noktası civarında:

$$f(x) \approx \sum_{n=0}^{N} \frac{f^{(n)}(a)}{n!} (x-a)^n$$

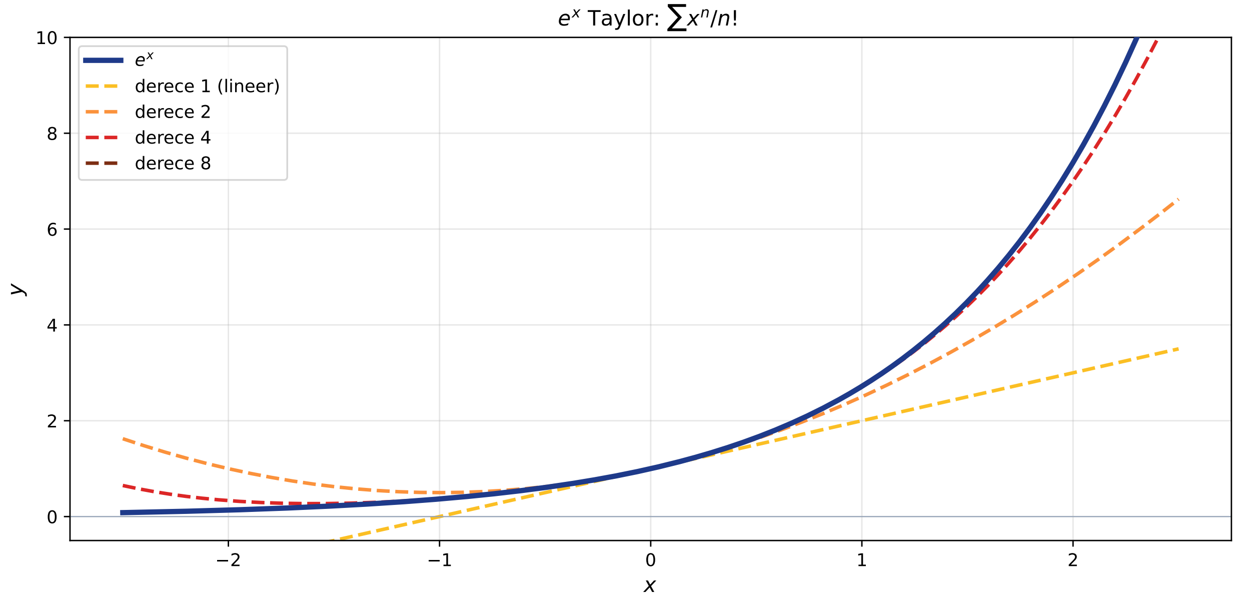
En şık örnek e^x'tir: e^x'in türevi kendisi olduğundan tüm türevler e^x, $x = 0$'da hepsi 1. Dolayısıyla tüm katsayılar $1/n!$:

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots$$

> Builder Notu — Gradient ve Hessian
>
> ML'de en sık ilk iki/üç terimini kullanırsın: $f(a)$ (değer), $f'(a)(x-a)$ (gradient/lineer terim), $\frac{1}{2}f''(a)(x-a)^2$ (Hessian/kuadratik terim). Optimizasyon, kayıp manifoldunu bu kesilmiş Taylor ile modeller.

11.6 İkinci Terimin Geometrik Anlamı (FTC)

Taylor'ın ikinci-derece terimini Calculus'un Temel Teoremi'yle de görebiliriz. Bir grafiğin altındaki **alanı** veren fonksiyonu düşün. Bu kez grafiği değil, **alan fonksiyonunu** yaklaştırıyoruz.



Şekil 11.3: e^x Taylor serisi: birkaç terimle bile $[-2, 2]$ aralığında çok iyi; her yerde yakınsar.

FTC der ki: grafiğin kendisi, alan fonksiyonunun türevidir. Ama değişim $x - a$ küçük değilse, bir de şu **üçgeni** hesaba katmalısın. Tabanı $x - a$, yüksekliği grafiğin eğimi çarpı $x - a$. Grafik, alan fonksiyonunun türevi olduğundan, onun eğimi alan fonksiyonunun **ikinci türevidir**:

$$\frac{1}{2}(x-a) \cdot f''(a)(x-a) = \frac{1}{2}\, f''(a)\,(x-a)^2$$

$$A(x) \approx f(a) + f'(a)(x-a) + \frac{1}{2} f''(a)(x-a)^2$$

Bu tam olarak Taylor polinomudur — ama her terimin diyagramda işaret edebileceğin net bir anlamı var: değer, dikdörtgen, üçgen.

Builder Notu — Trust Region

Optimizasyonda $f(a)$ mevcut kayıp, $f'(a)$ gradyan (lineer iyileşme), $\frac{1}{2} f''(a)$ eğrilik düzeltmesi (Hessian terimi) — trust-region yöntemleri tam bu kuadratik modeli kurar ve ne kadar güvenebileceğini (adım yarıçapını) eğrilik terimine göre ayarlar.

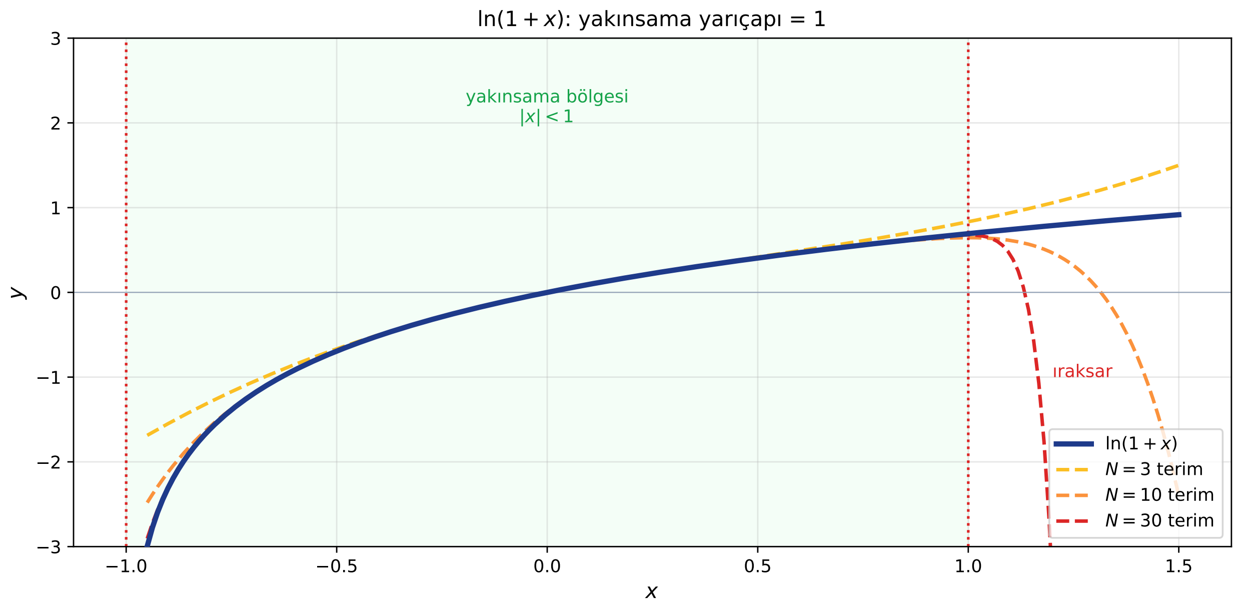
11.7 Yakınsama: Taylor Polinomu vs Serisi

Hiç durmayıp **sonsuz** terim eklesek? Matematikte sonsuz toplama **seri** denir.

e^x'in Taylor serisine herhangi bir x için seri e^x'e yakınsar — $x = 0$'daki türev bilgisinden kurulmuş olmasına rağmen, her girdide geçerli. e^x kendi Taylor serisine **her yerde** eşittir (sin ve cos da öyle):

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} \qquad (\text{her } x)$$

Ama her zaman böyle olmaz. $\ln(x)$'in $x = 1$ civarındaki Taylor serisi, yalnızca $x \in (0, 2)$ aralığında yakınsar.



Şekil 11.4: Yakınsama yarıçapı: $\ln(1 + x)$ Taylor serisi $|x| < 1$ içinde yakınsar; $x = 1$ sınırda, $x > 1$'de ıraksar.

Yaklaştırdığın nokta ile serinin yakınsadığı en uzak nokta arasındaki mesafeye **yakınsama yarıçapı** denir.

> *"we say that e to the x equals its own Taylor series at all inputs x, which is kind of a magical thing to have happen."* — Grant, 19:18

Builder Notu — Güven Bölgesi

Yakınsama yarıçapı, bir seri-temelli yaklaşımın **nerede güvenli** olduğunu söyler. Bir aktivasyonu (örneğin GELU'yu) Taylor/seri açılımıyla yaklaştırırken, yalnızca yakınsama bölgesinde geçerlidir. Aynı şekilde, bir modeli bir çalışma noktası civarında lineerleştirdiğinde (kontrol, RL, perturbation analizi), yaklaşım yalnızca o nokta yakınında güvenilirdir.

11.8 Bu Dersin Özeti

1. Taylor serileri: polinom-olmayan bir fonksiyonu bir nokta civarında polinomla yaklaştırma sanatıdır.
2. $\cos(x) \approx 1 - \frac{1}{2}x^2$: değeri ($c_0 = 1$), eğimi ($c_1 = 0$) ve eğriliği ($c_2 = -1/2$) cos ile eşleştirerek.
3. Daha çok terim = daha yüksek türevleri eşleştirme. $\cos(x) \approx 1 - \frac{1}{2}x^2 + (1/24)x^4$.
4. Faktöriyeller doğal çıkar: katsayı $= f^{(n)}(0)/n!$.
5. Genel formül: $f(x) \approx \sum f^{(n)}(a)/n! \cdot (x - a)^n$. $e^x = 1 + x + x^2/2! + x^3/3! + \ldots$

6. İkinci terimin geometrik anlamı (FTC): alan = değer + dikdörtgen $f'(x-a)$ + üçgen $\frac{1}{2}f''(x-a)^2$.
7. Taylor serisi: bazen her yerde yakınsar (e^x, sin, cos), bazen yalnızca bir yakınsama yarıçapı içinde.

! Tek bir cümle

Taylor serisi, bir fonksiyonun tek bir noktadaki tüm türev bilgisini alıp o nokta civarında fonksiyonu yaklaştıran bir polinoma çevirir; n. terimin katsayısı $f^{(n)}(a)/n!$'dir ve yakınsama yarıçapı içinde, yeterince terimle polinom fonksiyonun kendisine eşit olur.

11.9 Kontrol Soruları

i Soru 1: sin(x)'in x = 0 civarında 3. dereceye kadar Taylor polinomu nedir?

Cevap: sin'in türevleri: sin, cos, $-$sin, $-$cos; $x = 0$'daki değerleri $0, 1, 0, -1$. Katsayılar $f^{(n)}(0)/n!$: $c_0 = 0$, $c_1 = 1$, $c_2 = 0$, $c_3 = -1/6$. Yani $\sin(x) \approx x - x^3/6$.

i Soru 2: Katsayıyı neden $f^{(n)}(0)$'a değil de $f^{(n)}(0)/n!$'e eşitliyoruz?

Cevap: x^n teriminin n. türevini alınca kuvvet kuralı kademeli iner ve $1 \cdot 2 \cdot 3 \cdots n = n!$ çarpanı çıkar. $n!$'e bölmek bu fazlalığı tam götürür, böylece polinomun n. türevi tam olarak $f^{(n)}(0)$'a eşit olur.

i Soru 3: $e^x \approx 1 + x + x^2/2$ yaklaşımıyla $e^{0,1}$'i tahmin et.

Cevap: $1 + 0{,}1 + (0{,}1)^2/2 = 1{,}105$. Gerçek değer $e^{0,1} \approx 1{,}10517$. Yalnızca üç terimle dört hane doğru.

i Soru 4: (Builder) Newton's method, kaybın 2. dereceye kadar Taylor'ını kullanır. Bu parabolün minimumu nerededir?

Cevap: Yaklaşım $f(x) \approx f(a) + f'(a)(x-a) + \frac{1}{2}f''(a)(x-a)^2$. Minimum için türevini sıfırla: $f'(a) + f''(a)(x-a) = 0 \to x = a - f'(a)/f''(a)$. Bu, **Newton adımıdır**.

11.10 Egzersizler

Egzersiz 1. $\cos(x)$'in $x = 0$ civarında 4. dereceye kadar Taylor polinomunu yaz ve $\cos(0{,}5)$'i tahmin et.

Egzersiz 2. $f(x) = 1/(1-x)$'in $x = 0$ civarındaki Taylor serisini bul. (İpucu: $1 + x + x^2 + x^3 + \ldots$)

Egzersiz 3. $\ln(x)$'in $x = 1$ civarında 2. dereceye kadar Taylor polinomunu bul.

Egzersiz 4. *(Python — sembolik)* SymPy'nin `series` fonksiyonuyla Taylor açılımları al.

```
cos(x) -> 1 - x**2/2 + x**4/24 + O(x**6)
exp(x) -> 1 + x + x**2/2 + x**3/6 + x**4/24 + x**5/120 + O(x**6)
```

```
sin(x) -> x - x**3/6 + x**5/120 + O(x**6)
log(x + 1) -> x - x**2/2 + x**3/3 - x**4/4 + x**5/5 + O(x**6)
```

Egzersiz 5. *(Sonraki dersin habercisi)* Türevi düşünmenin, bu seride gördüğümüzden farklı, daha derin bir görsel yolu var mı? Ders 12, türevin alternatif bir geometrik yorumunu sunacak.

11.11 Anahtar Kavramlar (Cheat Sheet)

Kavram	Tanım	Grant'ta
Taylor polinomu	Fonksiyonu bir nokta civarında polinomla yaklaştırma	1m43
$\cos(x) \approx 1 - \frac{1}{2}x^2$	Değer + eğim + eğrilik eşleştirme	8m02
Katsayı $= f^{(n)}(0)/n!$	Faktöriyel, kuvvet kuralı kademesini götürür	8m32
$e^x = \sum x^n/n!$	Tüm türevler 1; her yerde yakınsar	13m35
Genel: $f^{(n)}(a)/n! \cdot (x-a)^n$	a noktası civarında Taylor	13m02
İkinci terim = üçgen	$\frac{1}{2}f''(a)(x-a)^2$ (FTC, alan yorumu)	16m18
Taylor serisi	Sonsuz terim; yakınsarsa fonksiyona eşit	17m30
Yakınsama yarıçapı	Serinin yakınsadığı maksimum mesafe	20m44

11.12 ML Bağlantıları Özeti

7 köprü

1. **Birinci-derece Taylor** → gradient / lineerleştirme; gradient descent adımı $f(a) + f'(a)(x-a)$.
2. **İkinci-derece Taylor** → Hessian, Newton's method, trust region, doğal gradyan; eğrilik düzeltmesi.
3. e^x **serisi** → softmax/exp hesabı; GELU'nun tanh/erf yaklaşımı, üstel-aile dağılımları.
4. $f^{(n)}/n!$ **katsayıları** → faktöriyel paydalar, serinin hızlı yakınsaması.
5. **"Tek nokta türevi → civar davranış"** → yerel / surrogate modeller, perturbation analizi.
6. **Yakınsama yarıçapı** → bir yaklaşımın geçerli kaldığı güven bölgesi.
7. **Kesilmiş Taylor** → düşük dereceli modeller; ikinci-derece optimizasyonun matematiksel iskeleti.

❗ Tek bir şey alıp gideceksen

Taylor serisi, bir fonksiyonun **tek** bir noktadaki türev bilgisini, o nokta **civarındaki** davranışına çevirir. Katsayılar $f^{(n)}(a)/n!$; ilk birkaç terim ML'de her gün kullandığın yaklaşımlardır — gradient (birinci derece) ve Hessian (ikinci derece). "Karmaşığı yerel bir polinomla değiştir", calculus'un en güçlü pratik hamlesidir.

12 Calculus’ta Sana Öğretmedikleri

Dönüşümsel görüş — sayı doğrusundan sayı doğrusuna germe/sıkıştırma

i Bölüm bilgisi

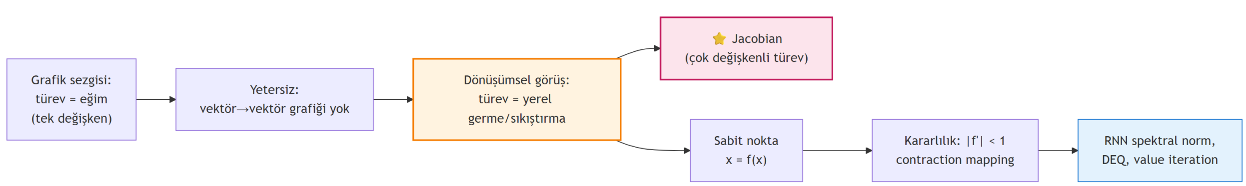
- **Grant’ın videosu:** YouTube — Chapter 12: What they won’t teach you in calculus (≈14 dk)
- **Kaynak:** 3Blue1Brown — Essence of Calculus
- **Okuma süresi:** ≈22 dk

12.1 Bu Derste Ne Var?

Serinin son bölümü. Çoğu calculus dersinin atladığı ama öğrenmeyi hızlandıran bir bakış: **dönüşümsel görüş**. Bu seride tüm sezgilerimiz **grafiklere** dayandı (türev = eğim, integral = alan). Ama girdi-çıktısı yalnızca sayı olmayan fonksiyonlara geçince grafik çizemezsin; grafik-temelli sezgi, çok değişkenli calculus ve kompleks analiz gibi ileri konulara gereksiz bir engel olur.

Üç ana fikir:

1. **Dönüşümsel görüş:** türev = girdiyi bir nokta civarında ne kadar gerip/sıkıştırdığı. (x^2’nin 1’deki türevi 2 = yerel olarak $\times 2$ germe.)
2. **Sabit nokta:** sonsuz kesir, $f(x) = 1 + 1/x$’in **sabit noktasıdır**; iki çözüm var ($\varphi \approx 1{,}618$ ve $-1/\varphi \approx -0{,}618$).
3. **Kararlılık:** $|f'| < 1$ ise sabit nokta **çekici** (stable), $|f'| > 1$ ise **itici** (unstable).



Dönüşümsel türev: eğim olmaktan duyarlılık olmaya, sayı doğrusundan Jacobian’a.

> *“the stability of a fixed point is determined by whether or not the magnitude of its derivative is bigger or smaller than 1.”* — Grant, 12:23

Builder Notu — ML Köprüleri

- **Germe/sıkışma faktörü → Jacobian determinantı.** Çok değişkenlide türev, yerel hacmi ne kadar gerip sıkıştırdığını ($\det J$) söyler; **normalizing flows**'ta log-det-Jacobian tam budur.
- **Negatif türev → yön çevirme** (Jacobian determinantının işareti); **sıfır türev → çöküş** (tekil/singular Jacobian, bilgi kaybı).
- **Kararlılık** $|f'| < 1$ **→ contraction mapping** (Banach sabit nokta teoremi): DEQ yakınsaması, power iteration, RL'de value iteration.
- $|$**türev**$|$ **ve kararlılık → RNN'de gradyan patlama/sönme:** tekrarlı çarpımın Jacobian spektral normu > 1 ise patlar, < 1 ise söner; **spectral normalization** ve Lipschitz kısıtları tam bunu kontrol eder.

12.2 Grafik Sezgisinin Sınırı: Neden Yeni Bir Bakış?

Bu ilk yılın görsel sezgilerinin neredeyse tamamı **grafiklere** dayanır: türev bir grafiğin eğimi, integral o grafiğin altındaki alan. Ama calculus'u, girdi ve çıktısı yalnızca sayı olan fonksiyonların ötesine genelledikçe, analiz ettiğin fonksiyonu her zaman grafikleyemezsin.

Grant'ın paylaştığı alternatif: türevi, daha sorunsuz genelleşen bir biçimde düşünmek. Önemli nokta: "türev = eğim"i türevin **tanımı** sanma. Türev, daha temelde, **fonksiyonun girdideki küçük dürtmelere ne kadar duyarlı olduğudur**.

Builder Notu — Jacobian

"Türev = eğim"e fazla bağlanmanın bedeli, ML matematiğinde net görülür: orada türev neredeyse hiç "eğim" değildir. Bir sinir ağı vektörleri vektörlere eşler; türevi bir **Jacobian matrisidir** (yerel lineer dönüşüm). "Duyarlılık" ve "yerel dönüşüm" bakışı, tek-değişkenli eğimden çok daha sorunsuz genelleşir.

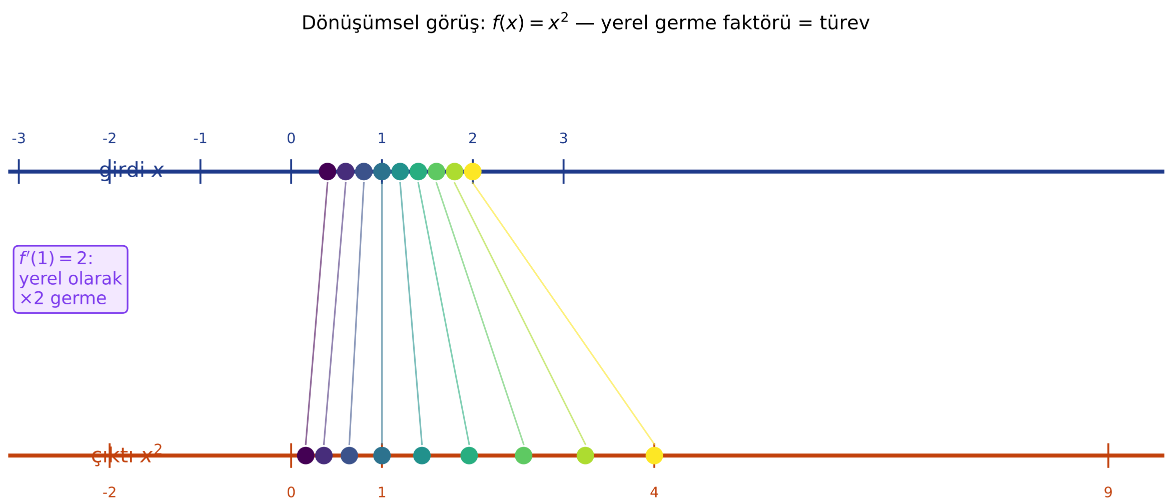
12.3 Dönüşümsel Görüş: Türev = Yerel Germe/Sıkışma

Alternatif görselin temel fikri: fonksiyonu, girdi doğrusundaki tüm noktaları **başka bir sayı doğrusundaki** karşılık gelen çıktılara eşleyen bir harita olarak düşün. Türev sana, girdi uzayının çeşitli bölgelerde ne kadar **gerildiğini ya da sıkıştığını** söyler.

Örnek: $f(x) = x^2$. Bu fonksiyon 1'i 1'e, 2'yi 4'e, 3'ü 9'a eşler. Girdi 1 etrafındaki küçük bir nokta kümesine yakınlaşıp nereye düştüklerine bakarsan, yaklaşık **2 katı** gerildiklerini görürsün.

$$f(x) = x^2 : \qquad f'(1) = 2, \quad f'(3) = 6, \quad f'\left(\tfrac{1}{4}\right) = \tfrac{1}{2}$$

Girdi 3 etrafında noktalar 6 katı gerilir; girdi $1/4$ etrafında ise $1/2$ katı **büzülür**.



Şekil 12.1: Dönüşümsel görüş: $f(x) = x^2$. Girdi doğrusundaki eşit aralıklı noktalar, çıktı doğrusunda nasıl gerilir veya sıkışır. Türev = yerel ölçek faktörü.

> **Builder Notu — Normalizing Flows**
>
> "Yerel germe/sıkışma faktörü" tek-değişkenlide bir sayı, çok değişkenlide bir **matristir** — Jacobian. Ve bu faktörün büyüklüğü (Jacobian'ın determinantı), bir bölgenin **hacminin** ne kadar gerildiğini söyler. **Normalizing flows** tam bunu kullanır: veriyi tersine çevrilebilir bir dönüşümle başka bir uzaya taşırken, olasılık yoğunluğunun nasıl değiştiğini log-det-Jacobian ile hesaplar.

12.4 Özel Durumlar: Sıfır, Negatif, Çöküş

Girdi 0'da: x^2 için 0 etrafına 10x, 100x, 1000x yakınlaştıkça, küçük bir komşuluğun giderek tek bir noktaya (0'a) **çöktüğünü** görürsün. Türevin 0 olması tam budur — yerel davranış, tüm sayı doğrusunu 0 ile çarpmaya benzer.

$$f'(0) = 2 \cdot 0 = 0$$

Negatif girdilerde: girdi -2 etrafındaki noktalar yalnızca gerilmez, aynı zamanda **ters çevrilir**.

$$f'(-2) = 2 \cdot (-2) = -4$$

> **Builder Notu — Tekil Jacobian**
>
> Negatif türev = yön çevirme, çok değişkenlide Jacobian determinantının **işaretine** karşılık gelir. Sıfır türev = çöküş ise **tekil (singular) Jacobian**: dönüşüm bir boyutu ezer, bilgi geri döndürülemez biçimde kaybolur. Bu, otomatik kodlayıcılarda dar boğazın ve boyut indirgemenin neden bilgi kaybettiğinin

geometrik nedenidir.

12.5 Sonsuz Kesir Bulmacası: İki Sabit Nokta

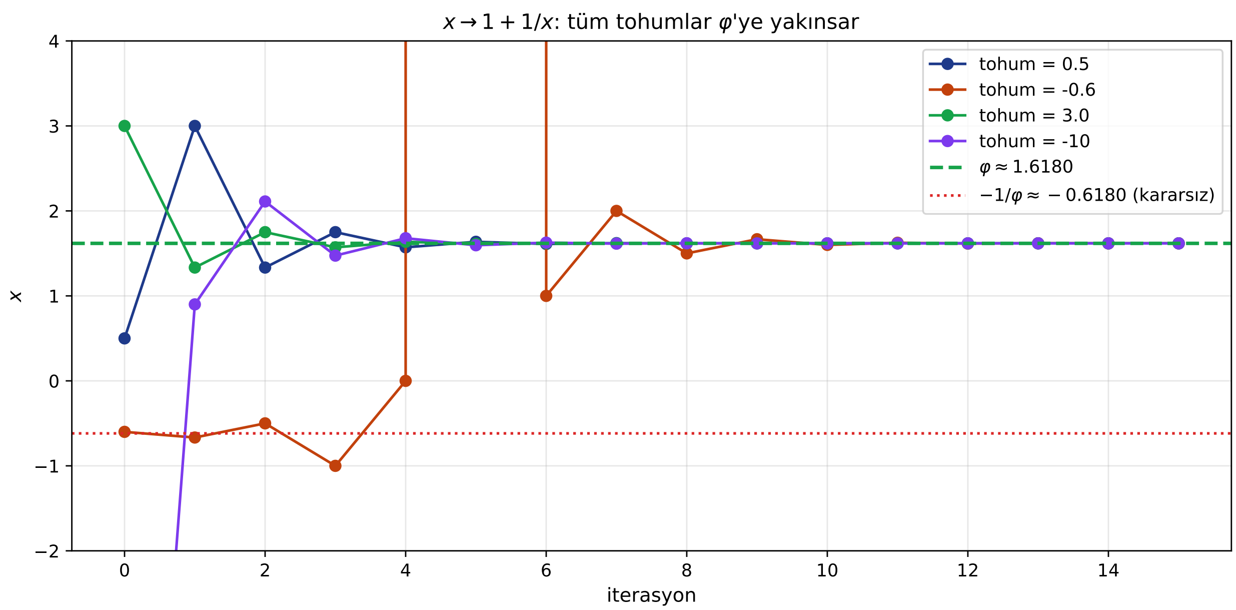
Sonsuz kesir $1 + \cfrac{1}{1 + \cfrac{1}{1 + \cfrac{1}{1 + \dots}}}$ aslında $f(x) = 1 + 1/x$ fonksiyonunun bir **sabit noktasını** arıyor:

$$x = 1 + \frac{1}{x} \qquad x^2 - x - 1 = 0$$

İki çözüm var: altın oran φ ve onun “küçük kardeşi” $-1/\varphi$:

$$\varphi = \frac{1+\sqrt{5}}{2} \approx 1{,}618, \qquad -\frac{1}{\varphi} \approx -0{,}618$$

Bir hesap makinesi al, **herhangi** bir sayıyla başla ve $f(x) = 1 + 1/x$’i defalarca uygula: hangi sayıyla başlarsan başla, sonunda hep $1{,}618$’e varırsın — küçük kardeşe çok yakın başlasan bile ondan kaçıp φ’ye sıçrar.



Şekil 12.2: Sabit nokta iterasyonu: $f(x) = 1 + 1/x$. Farklı tohumlar hep φ’ye yakınsar; küçük kardeş kararsız.

Builder Notu — DEQ / Value Iteration

"$x = 1 + 1/x$'in sabit noktasını ara, ama iterasyonla bul" — bu, **sabit-nokta iterasyonunun** tam örneğidir. ML'de Deep Equilibrium Models (DEQ) çıktısını $z = f(z, x)$ sabit noktası olarak tanımlar; RL'de value iteration, Bellman operatörünün sabit noktasını arar.

12.6 Sabit Noktaların Kararlılığı: $|f^{'}| < 1$

Cevap, dönüşümsel türevde. φ etrafına yakınlaş: eşleme sırasında o bölgedeki noktalar φ'ye doğru **büzülür**, yani $f(x) = 1 + 1/x$'in oradaki türevinin büyüklüğü 1'den küçüktür.

$$f(x) = 1 + \frac{1}{x}, \qquad f'(x) = -\frac{1}{x^2}$$

$$|f'(\varphi)| \approx 0{,}38 < 1 \ (\text{stable}), \qquad \left|f'\left(-\tfrac{1}{\varphi}\right)\right| \approx 2{,}62 > 1 \ (\text{unstable})$$

İşte çok yararlı bir gerçek: bir sabit noktanın **kararlılığı**, türevinin büyüklüğünün 1'den küçük mü büyük mü olduğuyla belirlenir. $|f'| < 1$ ise kararlı (çekici); $|f'| > 1$ ise kararsız (itici).

> *"the stability of a fixed point is determined by whether or not the magnitude of its derivative is bigger or smaller than 1."* — Grant, 12:23

Builder Notu — RNN ve Spectral Normalization

$|f'| < 1$ kararlılık koşulu, ML'in her yerindedir. Bir **contraction mapping** (Banach sabit nokta teoremi) tam olarak $|f'| < 1$ olan haritadır; DEQ ve value iteration'ın yakınsamasını bu garanti eder. Tekrarlı dinamiklerde (RNN'ler) ise tekrarlı çarpımın Jacobian'ının spektral normu > 1 ise gradyan **patlar**, < 1 ise **söner** — bu yüzden gradient clipping, ortogonal başlatma ve **spectral normalization** ile bu büyüklük 1 civarında tutulur.

12.7 Neden Öğrenmeli? Sonrası İçin

Grant'ın dürüst itirafı: türevi bu "yoğunluk değişimi" olarak görmek, bütün bir fonksiyonu resmetmek söz konusu olduğunda grafiklerden daha hantal olabilir. Asıl neden, tek-değişkenli calculus anlayışına kattığı şey değil — **sonrasında geleni** kolaylaştırmasıdır.

Çok değişkenli calculus, kompleks analiz, diferansiyel geometri… hepsinde fonksiyonlar artık birer dönüşümdür ve türev bir yerel germe/sıkışma (Jacobian) olarak çok daha doğal oturur.

> *"the real reason I'd recommend you carry this perspective with you as you learn new topics … it's for what comes after."* — Grant, 13:58

Builder Notu — Asıl Çalışma Dili

ML matematiğinin tamamı, bu son dersin bakışıyla yazılır: fonksiyonlar dönüşümlerdir, türevler Jacobian'lardır, eğitim bu dönüşümlerin yerel davranışını (gradient, eğrilik, spektral norm) kontrol etmektir. Grafik sezgisi tek-değişkenlide harikadır; ama derin öğrenmeye geçtiğinde, Grant'ın bu "atlanmış" dönüşümsel görüşü senin asıl çalışma dilin olur.

12.8 Bu Dersin Özeti

1. Bu serinin grafik sezgisi (türev = eğim) güçlüdür ama sayı→sayı fonksiyonlarıyla sınırlıdır.
2. Dönüşümsel görüş: fonksiyon, bir sayı doğrusunu başka bir sayı doğrusuna eşleyen dönüşümdür; türev = yerel **germe/sıkışma faktörü**.
3. x^2: girdi 1'de $\times 2$, 3'te $\times 6$, $1/4$'te $\times 1/2$ (büzülme), 0'da çöküş (türev 0), -2'de $\times -4$ (gerilme + ters çevirme).
4. Sonsuz kesir = $f(x) = 1 + 1/x$'in sabit noktası; iki çözüm: $\varphi \approx 1{,}618$ ve $-1/\varphi \approx -0{,}618$.
5. Kararlılık: $|f'| < 1$ ise çekici (stable), $|f'| > 1$ ise itici (unstable).
6. Türev, daha temelde "girdideki küçük dürtmelere duyarlılık"tır; eğim bunun yalnızca bir görünümüdür.
7. Bu bakış, türevi grafiklerin ötesine (Jacobian, çok değişkenli calculus) taşıyan esnek köprüdür.

! Tek bir cümle

Türev, "eğim" olmaktan daha temelde, bir fonksiyonun girdiyi yerel olarak ne kadar gerip sıkıştırdığıdır (negatifse ters çevirir, sıfırsa çökertir); bu dönüşümsel bakış, sabit noktaların kararlılığını $|f'| < 1$ ile açıklar ve türevi grafiklerin ötesine — Jacobian'lara — taşıyan köprüdür.

12.9 Kontrol Soruları

Soru 1: f(x) = x^3'ün x = 2'deki türevini dönüşümsel olarak yorumla.

Cevap: $f'(x) = 3x^2$, dolayısıyla $f'(2) = 3 \cdot 4 = 12$. Dönüşümsel anlamı: girdi 2 etrafındaki küçük bir nokta kümesi, eşlemeden sonra yaklaşık **12 katı gerilir**.

Soru 2: f(x) = x/2'nin sabit noktası 0'dır. Kararlı mı? İterasyon ne yapar?

Cevap: $f'(x) = 1/2$. $|f'| = 1/2 < 1$ olduğundan 0 **kararlı** (çekici). Herhangi bir tohumdan başla: $x, x/2, x/4, x/8, \ldots \to 0$. Her adım komşuluğu yarıya büzer, bir contraction mapping.

Soru 3: Dönüşümsel görüşte negatif türev ne anlama gelir? Ya sıfır türev?

Cevap: Negatif türev: yerel komşuluk hem gerilir/büzülür hem de **ters çevrilir**. **Sıfır türev:** komşuluk giderek tek bir noktaya **çöker** — bilgi yerel olarak kaybolur.

ℹ Soru 4: (Builder) Bir RNN'in tekrarlı haritasının Jacobian'ının spektral normu ≈ 1,5 ise, uzun bir dizide gradyana ne olur?

Cevap: Geri yayılım, her zaman adımında bu Jacobian'la çarpar; spektral norm $\approx 1{,}5$ ise gradyanın büyüklüğü her adımda $\sim 1{,}5$ kat artar, T adımda $\sim 1{,}5^T$ olur — yani **patlar**. < 1 olsaydı $\sim 0{,}x^T$ ile **sönerdi**. Gradient clipping, ortogonal başlatma ve spectral normalization bu büyüklüğü 1 civarında tutmak içindir.

12.10 Egzersizler

Egzersiz 1. $f(x) = 2x + 1$'in her noktadaki germe faktörü nedir? (İpucu: f' sabit.)

Egzersiz 2. $f(x) = \sqrt{x}$'in $x = 4$'teki germe/sıkışma faktörünü bul. Bu, x^2'nin $x = 2$'deki germesinin tersi olmalı; neden?

Egzersiz 3. $f(x) = \cos(x)$'in sabit noktasını (Dottie sayısı $\approx 0{,}739$) bir hesap makinesinde tekrarlı cos alarak bul. Kararlı mı?

Egzersiz 4. *(Python — sabit nokta ve kararlılık)* $f(x) = 1 + 1/x$'i farklı tohumlardan iterasyonla uygula.

```
tohum    0.5: -> 1.618034
tohum   -0.6: -> 1.618034
tohum  100.0: -> 1.618034
tohum  -10.0: -> 1.618034
phi = 1.618034
|f'(phi)|     = 0.382  (<1, kararli)
|f'(-1/phi)|  = 2.618  (>1, kararsiz)
```

Egzersiz 5. *(Seri sonu)* Bir an dur ve 12 bölümü topla: küçük dürtmeler (dx), yaklaşıktan kesine geçiş (limit), türev (oran/germe), integral (toplam/alan), terslik (FTC), üsteller, Taylor, dönüşümler. "Calculus'u kendim icat edebilirdim" hissini bir cümleyle yaz.

12.11 Seri Sonu: Calculus'tan Sonra

Essence of Calculus'un 12 bölümü burada tamamlanıyor. Tek bir daireyi dilimleyerek başladık (Ders 1), türevi küçük dürtmelerle kurduk (Ders 2-4), üstelleri ve e'yi anladık (Ders 5-6), limitlerle her şeyi sağlamlaştırdık (Ders 7), integral-türev tersliğini kapattık (Ders 8-9), eğriliği ve Taylor'ı ekledik (Ders 10-11) ve nihayet türevin grafiklerin ötesine geçen dönüşümsel yüzünü gördük (Ders 12).

Sırada ne var? Bu dönüşümsel bakış, doğrudan **çok değişkenli calculus**a açılır: gradient (her yöndeki duyarlılık), Jacobian (vektör→vektör dönüşümün yerel hâli), Hessian (eğrilik matrisi) — hepsi burada gördüğün tek-değişkenli fikirlerin doğal genellemesi. ML için bu üçü kritiktir.

12.12 Anahtar Kavramlar (Cheat Sheet)

Kavram	Tanım	Grant'ta
Dönüşümsel görüş	Fonksiyon = bir sayı doğrusu $\rightarrow$ başka sayı doğrusu	2m22
Türev = germe/sıkışma faktörü	Girdinin yerel ölçek değişimi	2m33
x^2 **türevleri (dönüşümsel)**	1'de $\times 2$, 3'te $\times 6$, $1/4$'te $\times 1/2$	2m59
Sıfır türev = çöküş	Komşuluk tek noktaya ezilir	4m13
Negatif türev = ters çevirme	Gerilme + yön değişimi	4m55
Sabit nokta	$x = f(x)$; sonsuz kesir $\rightarrow \varphi, -1/\varphi$	6m05
Kararlılık $\|f'\| < 1$	Çekici (stable) vs itici (unstable)	12m23
Sonrası için	Jacobian, çok değişkenli; esnek bakış	13m58

12.13 ML Bağlantıları Özeti

7 köprü

1. **Germe/sıkışma → Jacobian determinantı** → normalizing flows'ta log-det-Jacobian, değişken değiştirme.
2. **Negatif/sıfır türev** → yönelim (det işareti) / tekil (singular) Jacobian: bilgi kaybı, bottleneck.
3. **Sabit nokta iterasyonu** → Deep Equilibrium Models (DEQ), RL'de value iteration, power iteration.
4. **Kararlılık** $|f'| < 1$ → contraction mapping (Banach sabit nokta teoremi); iteratif yöntemlerin yakınsama garantisi.
5. **Spektral norm** → RNN'de gradyan patlama/sönme; spectral normalization, Lipschitz kısıtları.
6. **Türev = duyarlılık (eğim değil)** → Jacobian; çok değişkenli calculus ve derin öğrenmenin asıl dili.
7. **Tersine çevrilebilirlik** → normalizing flows, invertible networks (Jacobian determinantı $\neq 0$ şartı).

! Bu dersten — ve tüm seriden — tek bir şey alıp gideceksen

Türev, bir eğrinin eğimi olmaktan daha temelde, bir fonksiyonun girdiye **duyarlılığıdır** — yerel bir germe/sıkışma. Bu bakış eğimden Jacobian'a, tek değişkenden derin ağlara sorunsuz geçer. 12 bölüm boyunca gördüğün her şey — küçük dürtmeler, limit, türev, integral, terslik, Taylor — tek bir cümlede toplanır: **küçük değişimlere yeterince dikkatle bakarsan, calculus'un tamamını kendin keşfedebilirsin.**