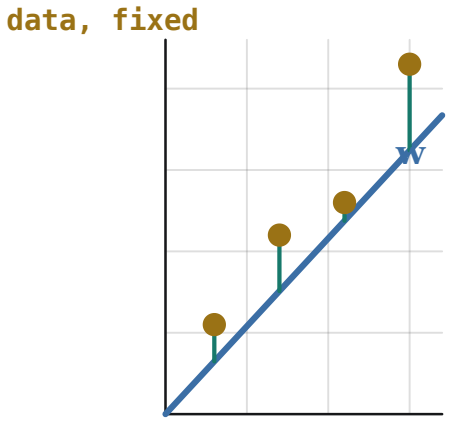
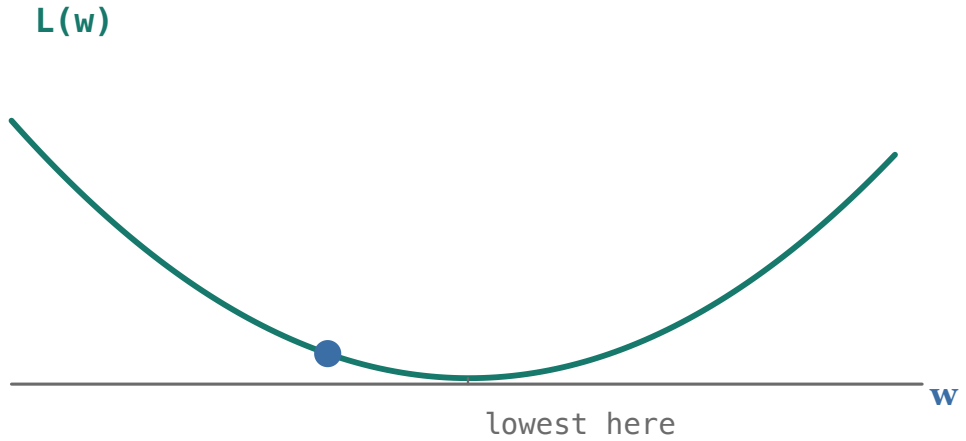
From Derivatives to Learning

A model that improves from data





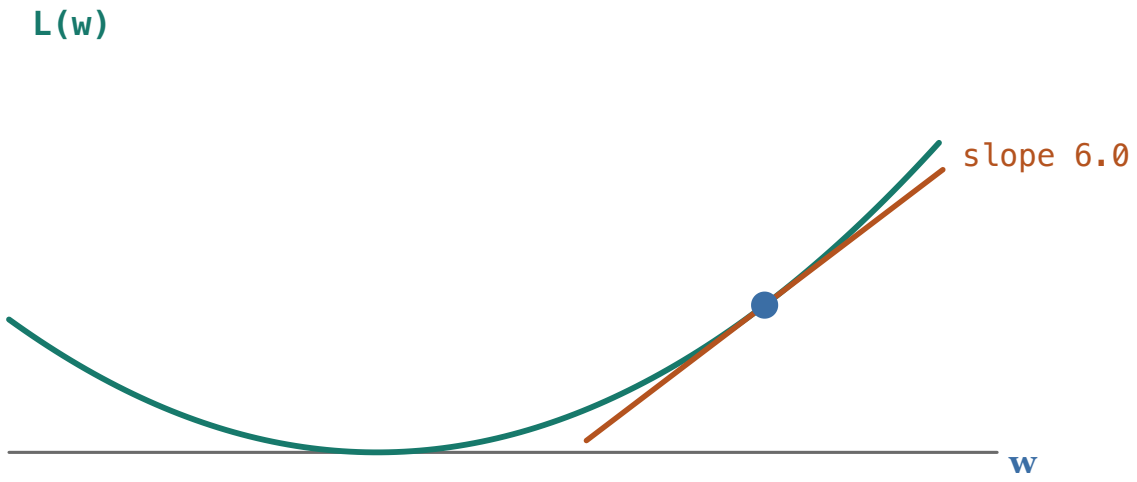
$$\hat{y} = f_\theta(x), \qquad L_1(\theta) = \tfrac{1}{2}\big(f_\theta(x) - y^\star\big)^2$$

- x, $y^\star$ given; the update never moves them
- θ the only thing we may change
- L_1 one example; the curve sums all four

> Which way should θ move, and how far?

From Derivatives to Learning

What the gradient gives us



parameter w

2.00

a value and a rate, never the same object

gradient dL/dw

6.0

$$g_j = \frac{\partial L}{\partial \theta_j}, \qquad g = \nabla_\theta L \in \mathbb{R}^p$$

g_j one number per parameter

g a column, shaped like θ

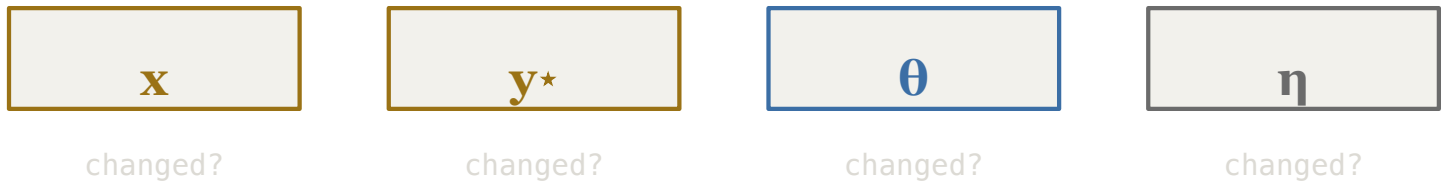
θ_j the parameter it belongs to

p how many parameters there are

VOTE

From Derivatives to Learning

Which quantity changes?



Fitting a **fixed** dataset: which quantity does the update directly adjust?

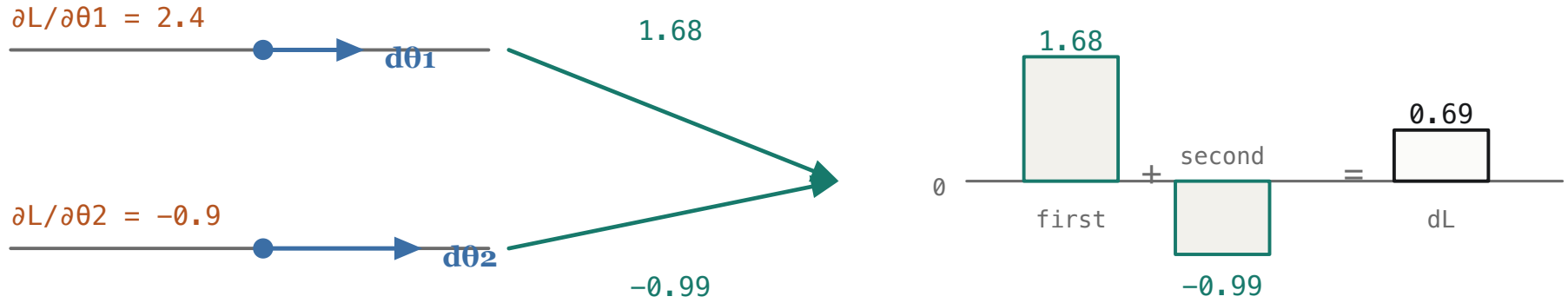
A the target $y^\star$

B the model weight θ

C the observed input x

From Derivatives to Learning

A change has several contributions

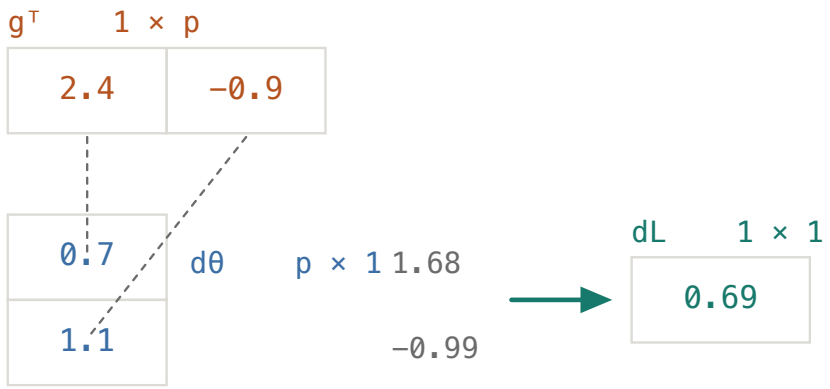


$$dL = \frac{\partial L}{\partial \theta_1}\, d\theta_1 + \frac{\partial L}{\partial \theta_2}\, d\theta_2$$

$\partial L/\partial \theta_j$	how sensitive L is to θ_j, at the current θ
$d\theta_j$	the change we make to that one parameter
`the sum`	the local linear model, not the exact change

From Derivatives to Learning

The sum is an inner product



the same pairing, for a finite step



in general not zero

$$dL = g^{\mathsf{T}} d\theta, \qquad L(\theta + \Delta\theta) - L(\theta) = g^{\mathsf{T}} \Delta\theta + o(\|\Delta\theta\|)$$

g^{T} $1 \times p$, a row of rates

$d\theta$ $p \times 1$, a column of increments

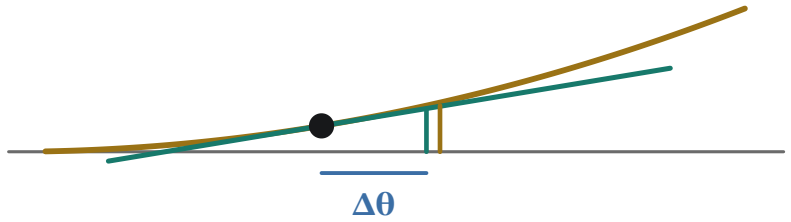
$g^{\mathsf{T}} d\theta$ 1×1: the differential, exact

$o(\|\Delta\theta\|)$ what a finite step leaves over

VOTE

From Derivatives to Learning

Derivative or finite change?



tangent
the loss itself

Which expression is an **exact** identity at a differentiable point?

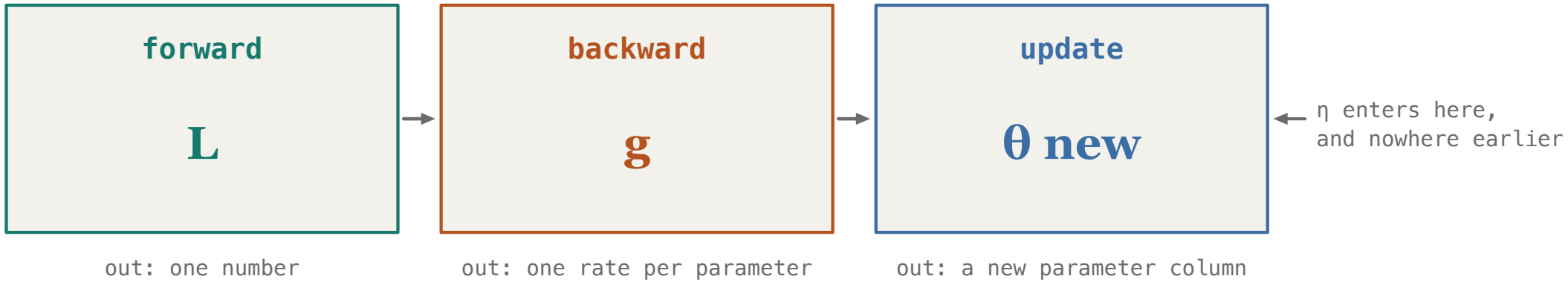
A $dL = g^\top d\theta$

B $\Delta L = g^\top \Delta\theta$ for every step $\Delta\theta$

C $dL = \|g\|$

From Derivatives to Learning

Three jobs in one learning step



(i) evaluate L (ii) compute $g = \nabla_\theta L$ (iii) choose $\Delta\theta$, $\theta_{\text{new}} = \theta + \Delta\theta$

(i) one number out

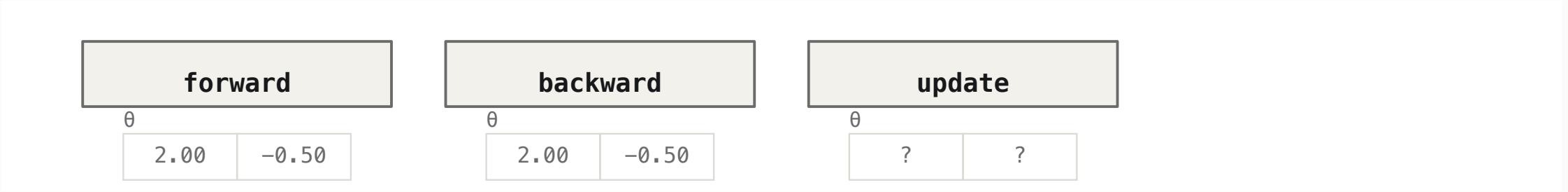
(ii) one rate per parameter out, and nothing is moved yet

(iii) a new parameter column; this is where η enters

VOTE

From Derivatives to Learning

Where does updating happen?



In ordinary gradient descent, which operation changes a weight?

A the forward pass

B the backward pass

C $\theta \leftarrow \theta - \eta g$

WHERE WE ARE

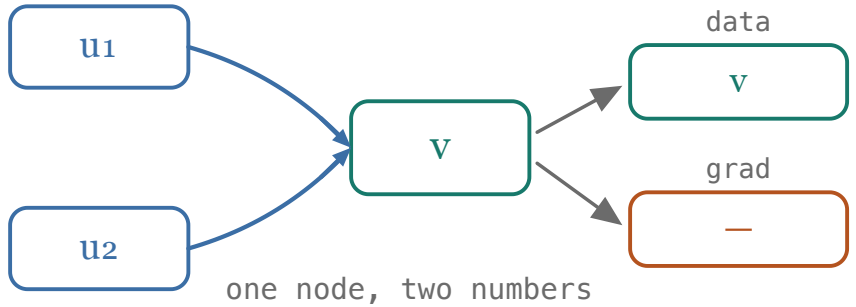
Automatic Differentiation

Remembering a computation

one node of the graph, as a record

data	v = f(u₁, u₂)
grad	– nothing yet
parents	u₁, u₂
backward	∂v/∂u₁ , ∂v/∂u₂

the same record, in place



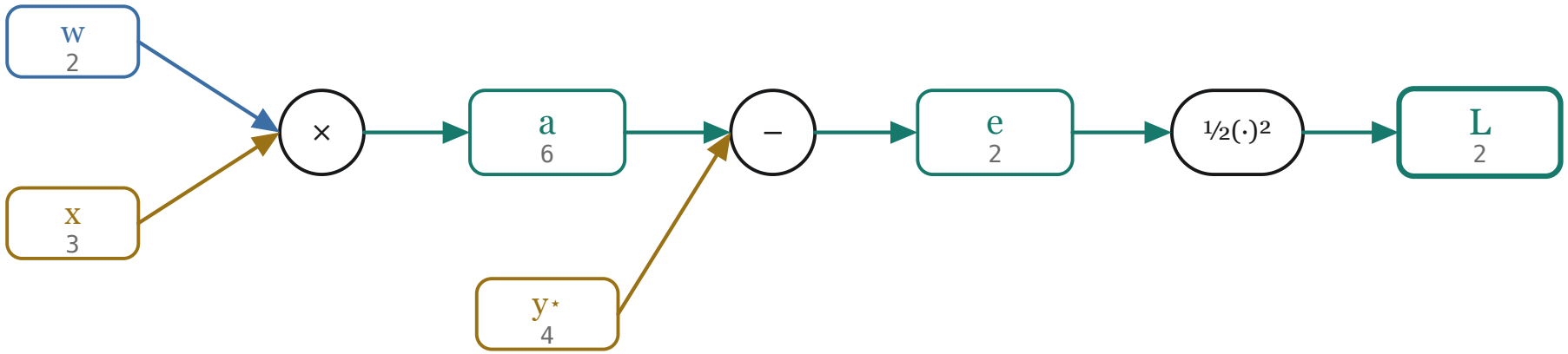
$$v = f(u_1, \ldots, u_m), \qquad \bar{v} = \frac{\partial L}{\partial v}$$

v the forward number it holds — $u_1, \ldots, u_m$ its parents, saved

$\bar{v}$ a second slot, filled backward — $\partial v / \partial u_i$ one local rule per parent

Automatic Differentiation

One graph, forward values



forward values only; nothing differentiated yet

$$a = wx, \qquad e = a - y^\star, \qquad L = \tfrac{1}{2}e^2$$

w the parameter, here 2 — x, $y^\star$ given: 3 and 4

a, e stored forward: 6 and 2 — L one scalar out: 2

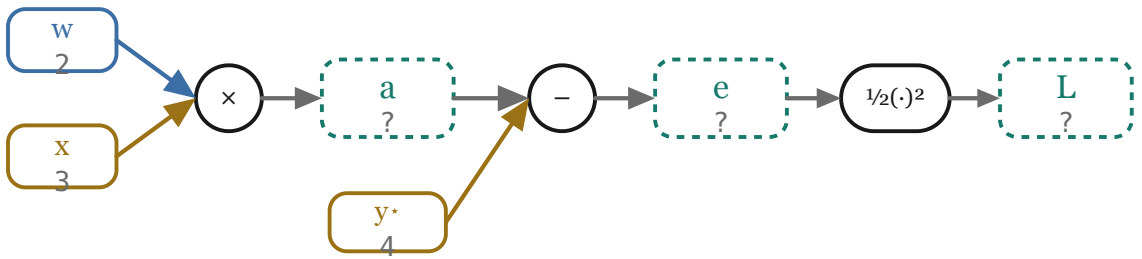
THE FIXTURE, FORWARD

$w = 2, x = 3, y^\star = 4$: $a = 6, e = 2, L = 2.$

VOTE

Automatic Differentiation

Value and sensitivity



node a

data	?	two numbers
grad	?	at one node

Same graph, at node a. What is the pair $(a, \bar{a})$?

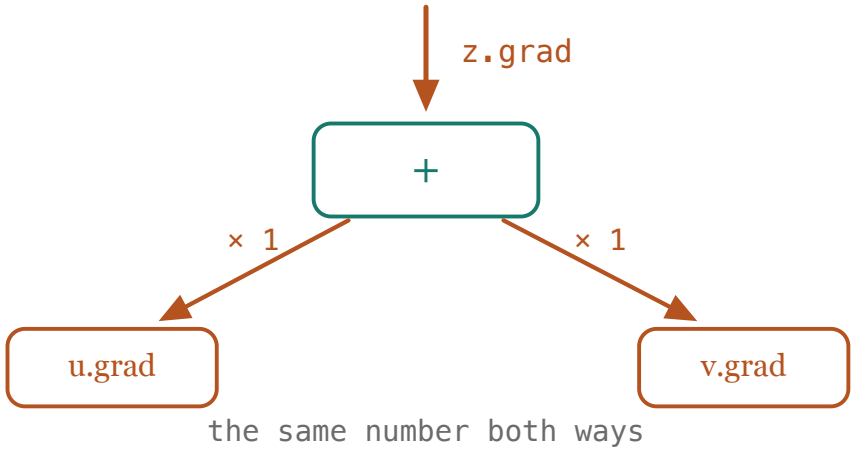
A $(6, 2)$

B $(2, 6)$

C $(6, 6)$

Automatic Differentiation

Local rules become backward callbacks



the local rule, as code

```
def _backward():
    u.grad += 1.0 * z.grad
    v.grad += 1.0 * z.grad
```

the local rule

the upstream factor

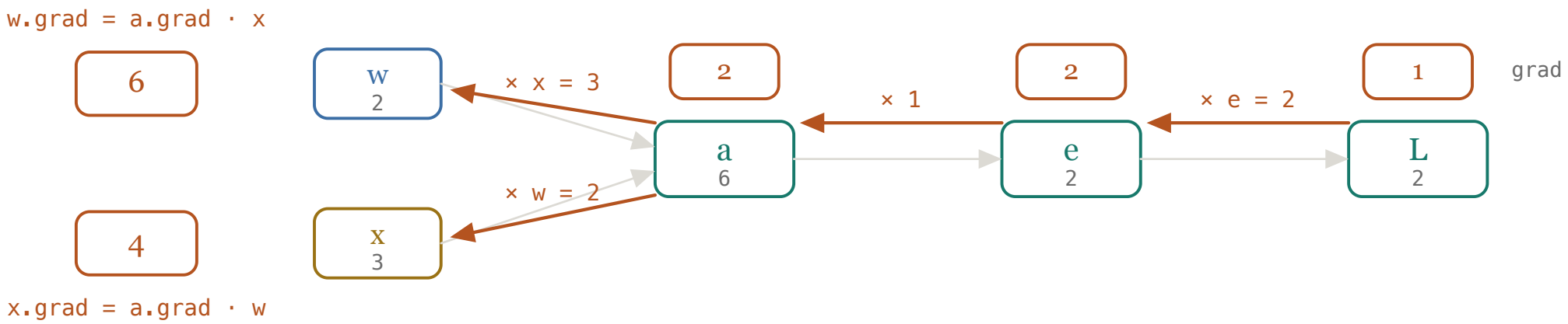
$$z = u + v: \qquad \bar{u} \mathrel{+}= \bar{z}, \qquad \bar{v} \mathrel{+}= \bar{z}$$

- $\bar{z}$ the upstream factor
- $\bar{u}, \bar{v}$ the two parent slots
- $\partial z/\partial u$ addition's local rule: 1
- $+=$ add in; a parent has other uses

`callback` a function *called during* the backward pass

Automatic Differentiation

Multiplication uses saved values



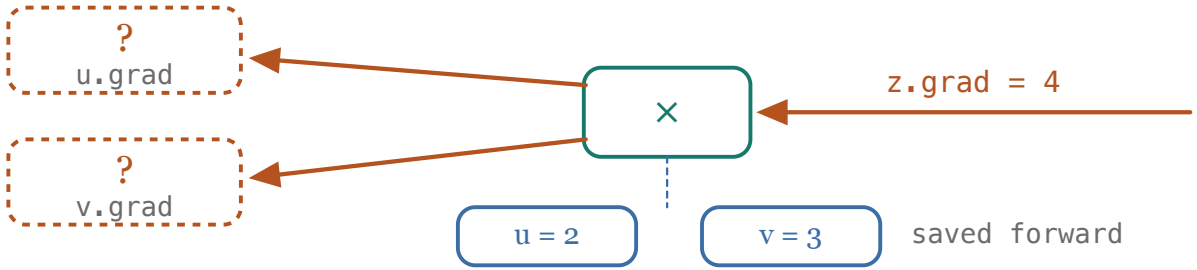
$$z = uv: \qquad \bar{u} \mathrel{+}= v\,\bar{z}, \qquad \bar{v} \mathrel{+}= u\,\bar{z}$$

u, v	both saved going forward	$\bar{z}$	the upstream factor
$v\,\bar{z}$	into $\bar{u}$: the *other* input	`why saved`	each side needs the other

VOTE

Automatic Differentiation

A missing upstream factor



$z = uv$ with $u = 2$, $v = 3$, and $\bar{z} = 4$ arriving. What lands in $\bar{u}$?

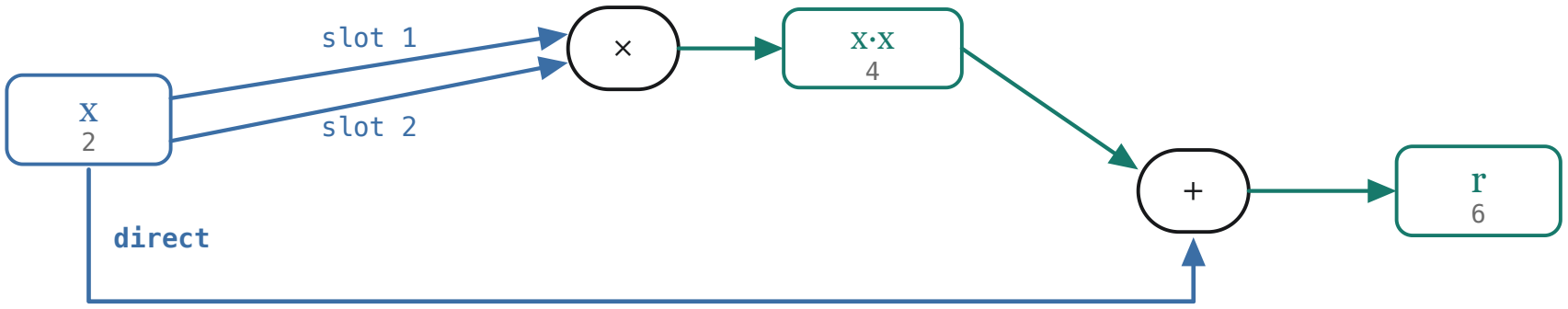
A 3

B 8

C 12

Automatic Differentiation

One variable, several uses



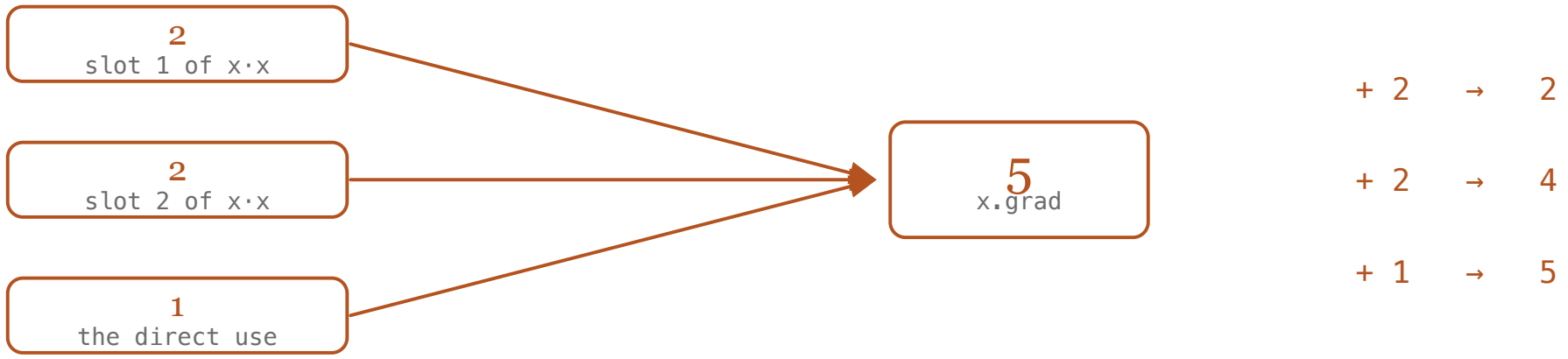
three edges out of x, visited once in traversal

$$r = x \cdot x + x$$

x	one node, used three times	$x \cdot x$	two argument slots
$+\,x$	a third, direct use	r	at $x = 2$ it stores 6

Automatic Differentiation

Contributions accumulate



every use reports; the slot adds, never replaces

$$\frac{dr}{dx} = \underbrace{x}_{\text{slot 1}} + \underbrace{x}_{\text{slot 2}} + \underbrace{1}_{\text{direct}} = 5 \quad (x = 2)$$

x the node, used three times $x \cdot x$ two slots, each giving x

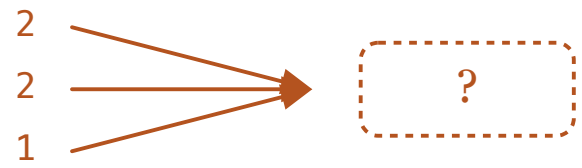
$+\,x$ the direct use, giving 1 dr/dx the sum, at $x = 2$: 5

VOTE

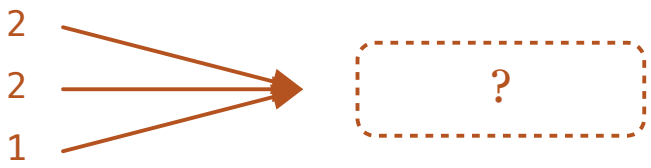
Automatic Differentiation

Assignment or addition?

`x.grad += contribution`



`x.grad = contribution`



For $r = x \cdot x + x$ at $x = 2$, what is dr/dx?

A 1

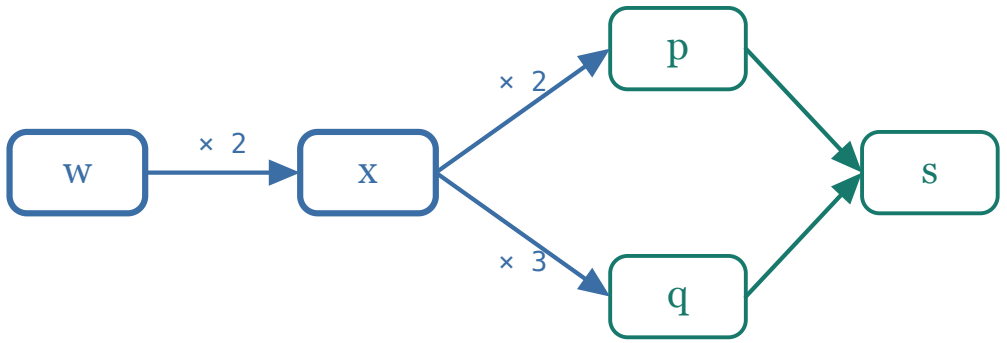
B 3

C 5

Automatic Differentiation

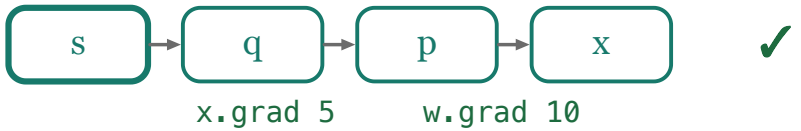
Reverse topological order

forward



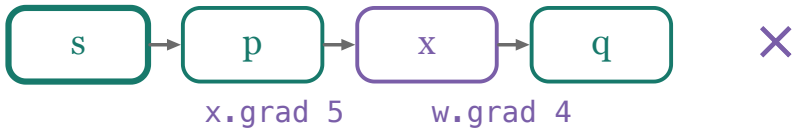
backward

a reverse topological order



✓

an invalid order



✗

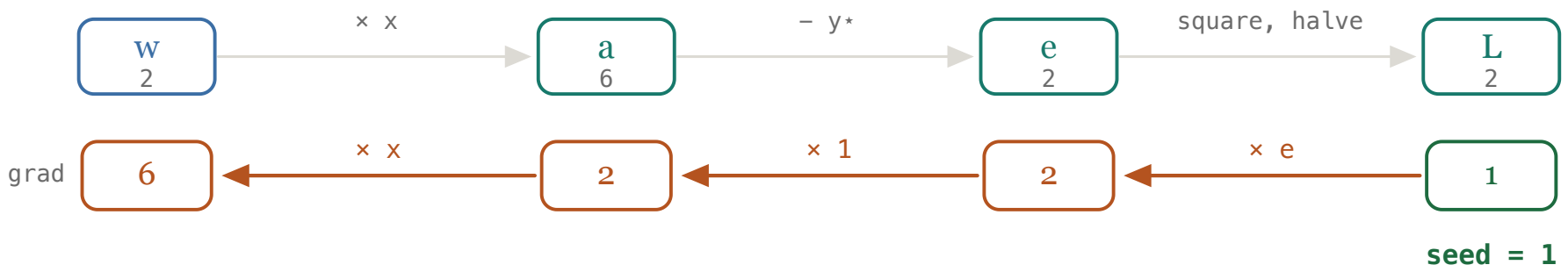
`x ran on a partial adjoint; its callback never runs again`

$$\bar{u} = \sum_{s\,:\,u \to v} \bar{v}\,\frac{\partial v}{\partial s}$$

$s : u \to v$	one input slot; vary it with the others fixed	$\bar{v}$	must be final already
`the sum`	complete when all uses report	`order`	any reverse topological one

Automatic Differentiation

The loss starts with one



the only number we set by hand is the seed

$$\bar{L} = \frac{\partial L}{\partial L} = 1$$

$\bar{L}$ the seed: the one adjoint we set by hand

1 a derivative, not the value of L

the rest start at 0, filled by a local rule

Automatic Differentiation

Trace the backward schedule



forward

```
0   zero every slot, seed  L̄ = 1        →   writes nothing
1   L's callback                        →   writes  ē = 2     a seed runs nothing: 1 into L̄ moves no slot
2   e's callback                        →   writes  ā = 2     each callback writes one slot upstream
3   a's callback                        →   writes  w̄ = 3·2 = 6
```

$$\bar{L} \leftarrow 1, \qquad \text{then run } L,\ e,\ a$$

`seed` a number put in a slot; it runs nothing

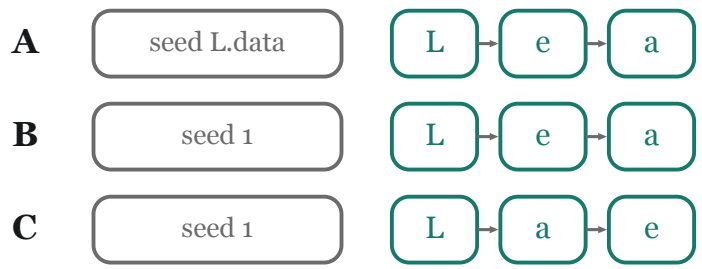
`callback` what writes into the slot below it

One step, one slot. $\bar{w} = 6$ at the end.

VOTE

Automatic Differentiation

Now two runs that break a rule



which run survives the rules?

Zero the slots, then seed and run. Which gives the right $\bar{w}$?

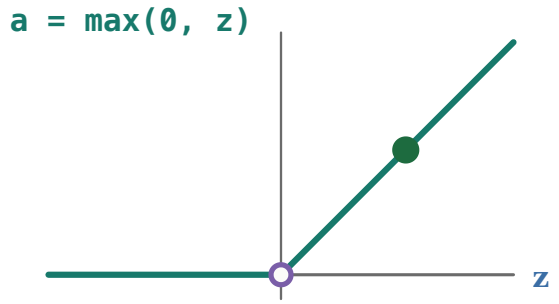
A $\bar{L} = L$: L, e, a

B $\bar{L} = 1$: L, e, a

C $\bar{L} = 1$: L, a, e

Automatic Differentiation

ReLU reuses its forward state



no ordinary derivative at 0
the engine returns 0 by convention

saved z	mask	what leaves
2	1	a.grad
−2	0	0

the mask reads z, never the sign of a.grad

$$a = \max(0, z), \qquad \bar{z} = \bar{a} \cdot \mathbf{1}_{z>0}$$

z the saved preactivation

$\bar{a}$ whatever arrives, any sign

$\mathbf{1}_{z>0}$ 1 if $z > 0$, else 0

at $z = 0$ no derivative; 0 by convention

VOTE

Automatic Differentiation

An open gate can pass a negative gradient



$a = \max(0, z)$ with $z = 2$ saved, and $\bar{a} = -3$ arrives. What is $\bar{z}$?

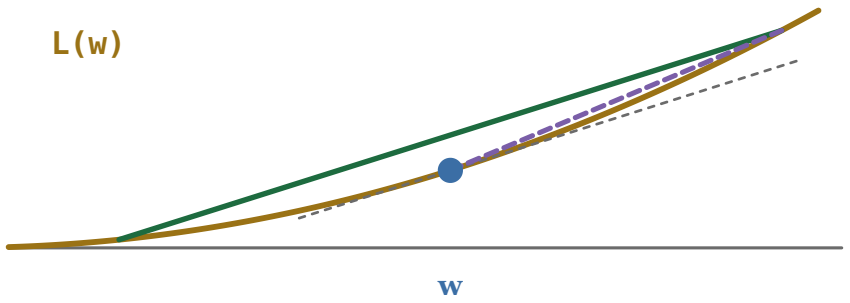
A 0

B -3

C 3

Automatic Differentiation

Checking a computed derivative



tangent

centred 6.000

one-sided 6.045 gap 0.045

and w is put back afterwards

$$\frac{L(w+\tau)-L(w-\tau)}{2\tau} \quad \text{vs} \quad \frac{L(w+\tau)-L(w)}{\tau}$$

τ the probe step, $1/100$ — w restored afterwards

centred exact here — one-sided off by $\tau x^2/2$

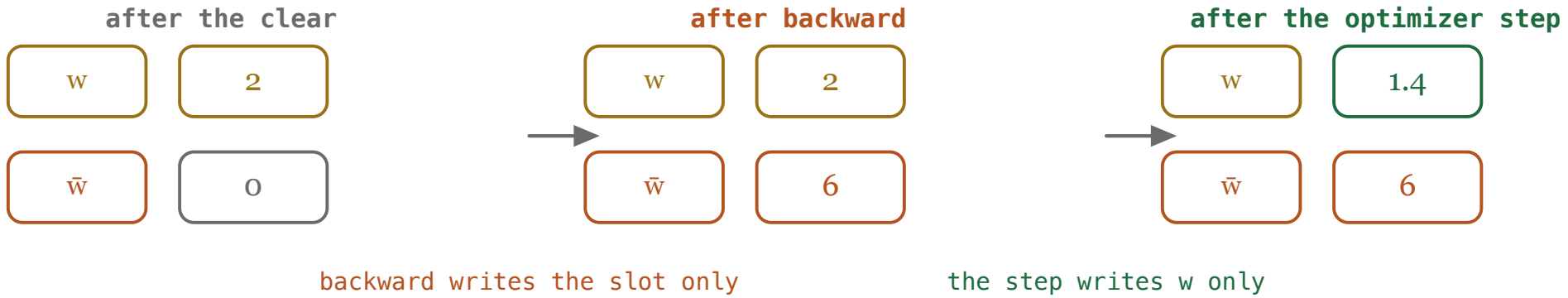
ONE DERIVATIVE, THREE ESTIMATES

By hand $\bar{a}\, x = 6$; the engine's grad, 6; the centred probe, 6.

One-sided: 6.045, over by $\tau x^2/2 = 0.045$. Agreement is evidence only.

Automatic Differentiation

Backward fills slots; plain SGD moves w




```
plain SGD:  w ← 2 − 0.1·6 = 1.4      the slot still reads 6
```

but 6 is the gradient AT w = 2. Nobody has computed the one at 1.4 yet.

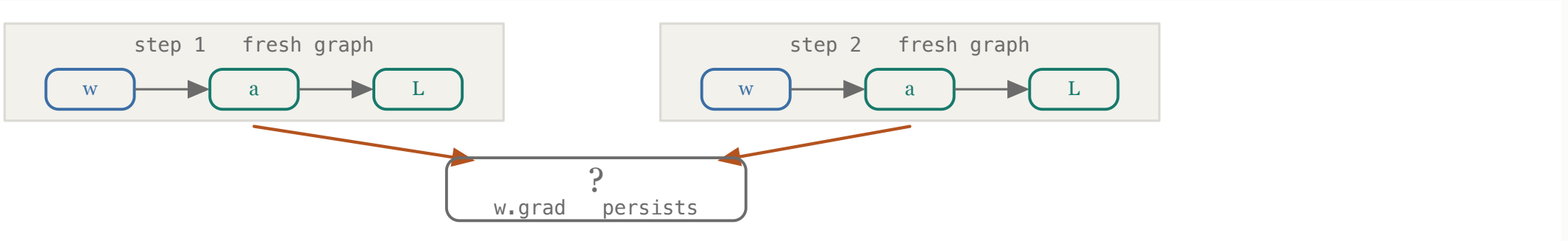
backward: $\bar{w} \mathrel{+}= \ldots$, plain SGD: $w \leftarrow w - \eta\,\bar{w}$

- w the parameter; only the optimizer assigns to it
- η step size, here 0.1; chosen, not computed
- $\bar{w}$ its slot; only backward and the clear write it
- $\bar{w} = 6$ read at $w = 2$; it is not the gradient at 1.4

VOTE

Automatic Differentiation

Why reset gradients?



Two fresh graphs at unchanged w, no clearing. What is in $\bar{w}$?

A this step's gradient only

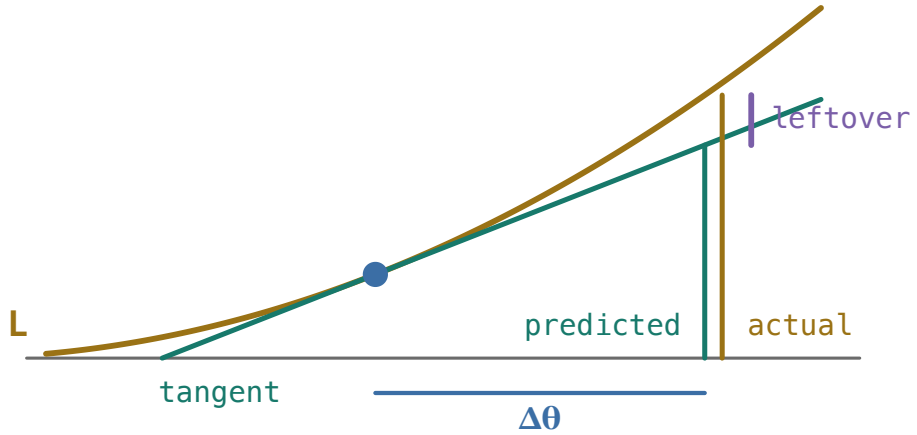
B the old and the new, added

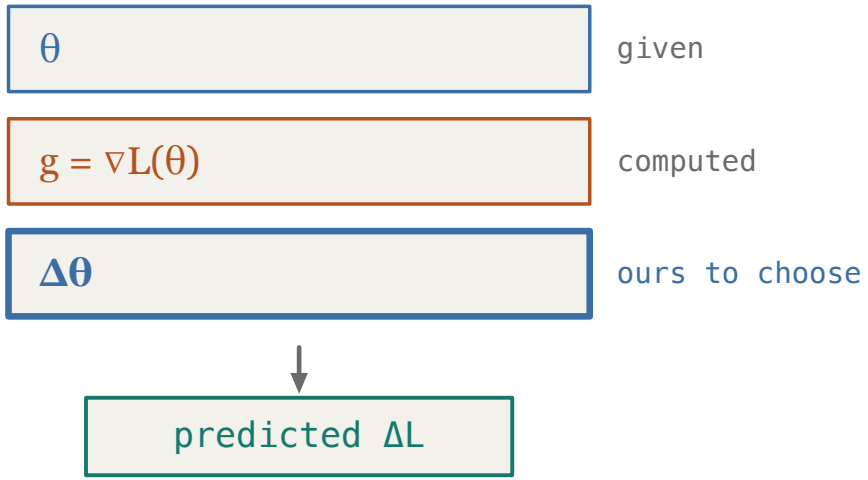
C w updates itself

RECALL

Gradient Descent and Step Size

Choosing a parameter change





$$\Delta L = L(\theta + \Delta\theta) - L(\theta) \approx g^{\mathsf{T}} \Delta\theta$$

$\Delta\theta$ $p \times 1$: the step, and the only thing we choose

g $p \times 1$: $\nabla_\theta L$ at the current θ, fixed while we choose

$\approx$ the linear part only; the leftover is dropped, not zero

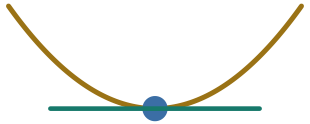
Gradient Descent and Step Size

Why the negative gradient?



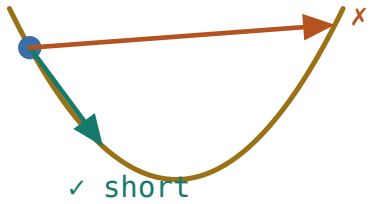
every term is a square, so the sum cannot be negative

if g = 0



prediction 0, and
−ηg moves nothing

if the step is long



the model said down;
the curve went up

$$\Delta\theta = -\eta g \implies g^{\mathsf{T}}\Delta\theta = -\eta\, g^{\mathsf{T}} g = -\eta \sum_{j=1}^{p} g_j^2 < 0$$

$\eta > 0$	our step size, never folded into g
$\sum_j g_j^2$	a sum of squares, so it is ≥ 0
< 0	needs only $g \neq 0$ and $\eta > 0$
$\Delta L < 0$	a different claim: the loss itself also needs a short step

VOTE

Gradient Descent and Step Size

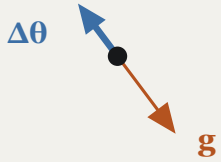
Which direction decreases the local model?

A $\Delta\theta = +\eta\, g$



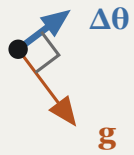
$g^\top \Delta\theta = ?$

B $\Delta\theta = -\eta\, g$



$g^\top \Delta\theta = ?$

C $\Delta\theta \perp g$



$g^\top \Delta\theta = ?$

For $g \neq 0$, $\eta > 0$: which step makes the predicted $g^\top \Delta\theta$ negative?

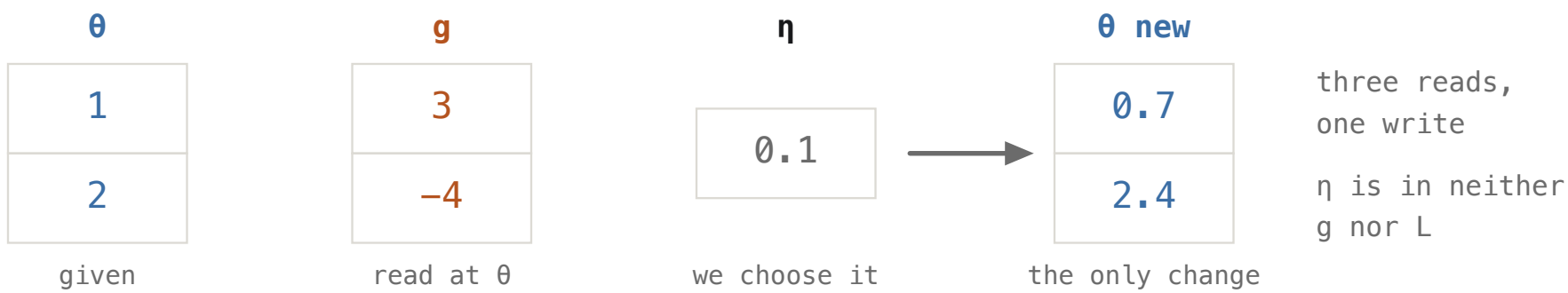
A $\Delta\theta = +\eta g$

B $\Delta\theta = -\eta g$

C a nonzero step orthogonal to g

Gradient Descent and Step Size

The gradient-descent update




```
1 − 0.1 × 3 = 0.7
2 − 0.1 × (−4) = 2.4
```

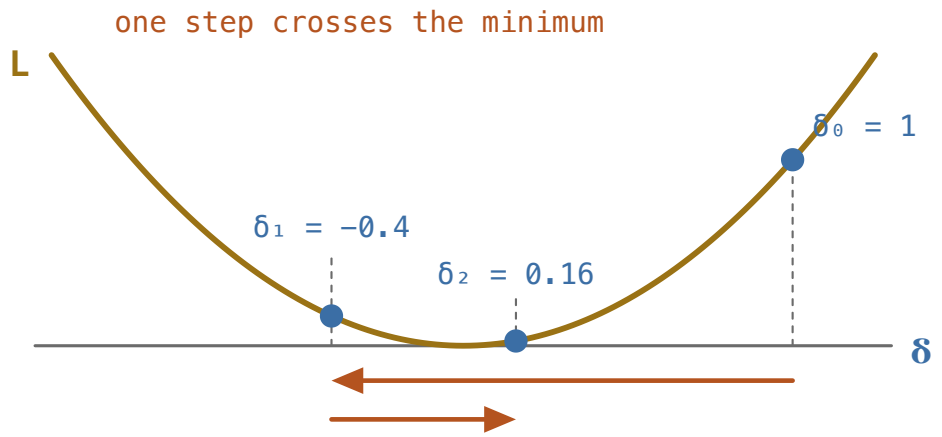
one entry at a time, with the same η

$$\theta_{t+1} = \theta_t - \eta_t g_t, \qquad g_t = \nabla_\theta L(\theta_t)$$

- θ_t the current column, $p \times 1$
- θ_{t+1} the new column, same shape
- g_t read at θ_t, not at θ_{t+1}
- η_t one positive number per step

Gradient Descent and Step Size

A scalar quadratic exposes overshoot



λ = 4, η = 0.35

g = λ δ

δ − η g = δ − ηλ δ

= (1 − ηλ) δ

ηλ = 1.4 > 1 **1 − ηλ = −0.4**

the step is longer than the distance to the minimum

$$L = \tfrac{1}{2}\lambda\delta^2, \quad g = \lambda\delta, \quad \delta_{t+1} = \delta_t - \eta\lambda\delta_t = (1 - \eta\lambda)\,\delta_t$$

δ_t $\theta_t - \theta^\star$, one number here

$\lambda > 0$ the curvature of this quadratic

$1 - \eta\lambda$ the multiplier: it decides every later step

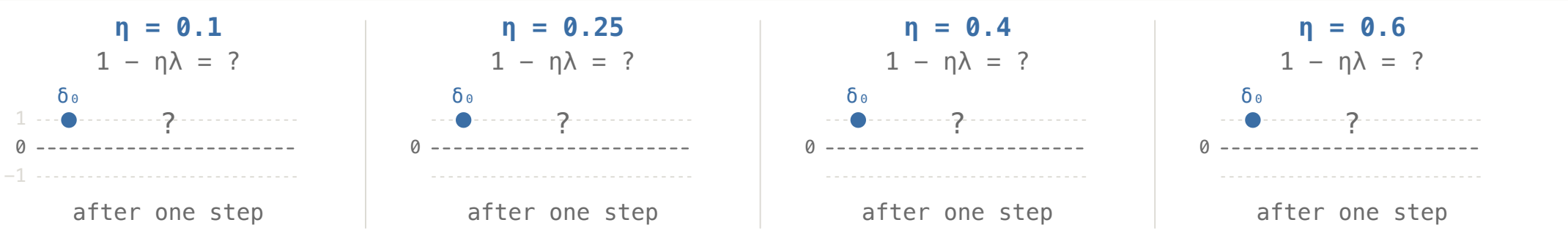
WORKED EXAMPLE

$\eta\lambda = 0.35 \times 4 = 1.4 > 1$, so $1 - \eta\lambda = -0.4$.

VOTE

Gradient Descent and Step Size

Four step-size regimes



Take $\lambda = 4$. Which pair is (converges while oscillating, diverges)?

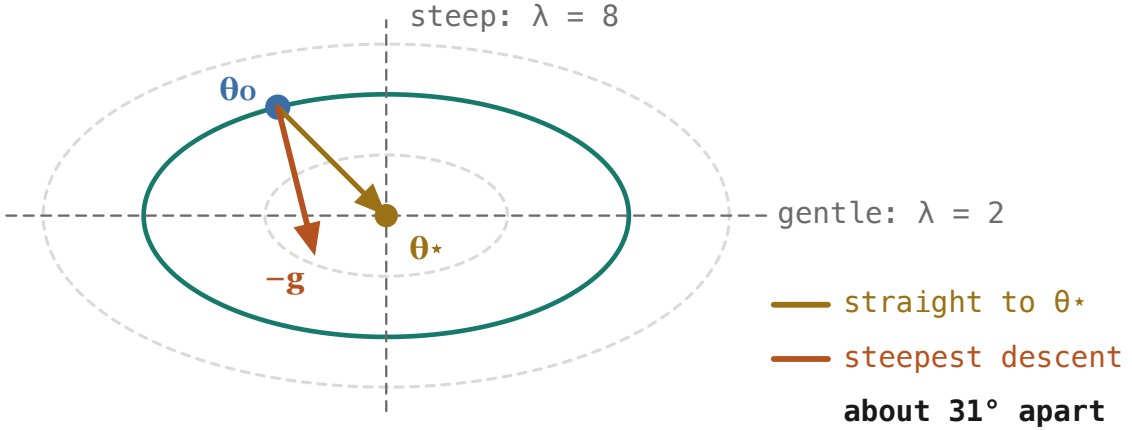
A $(0.4,\ 0.6)$

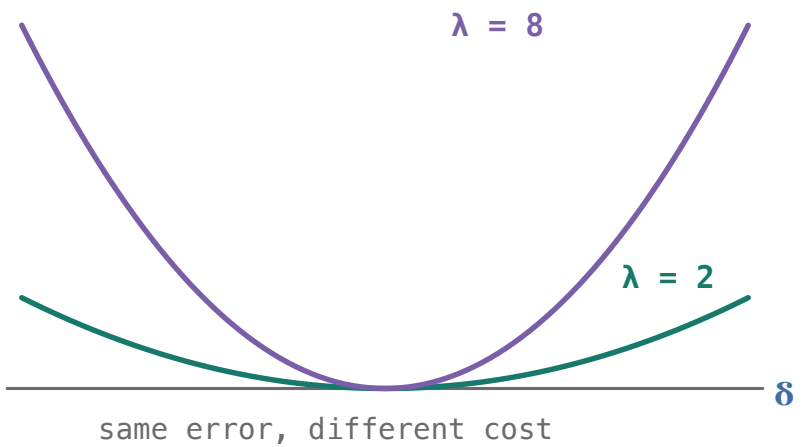
B $(0.1,\ 0.25)$

C $(0.6,\ 0.4)$

Gradient Descent and Step Size

Several curvature directions



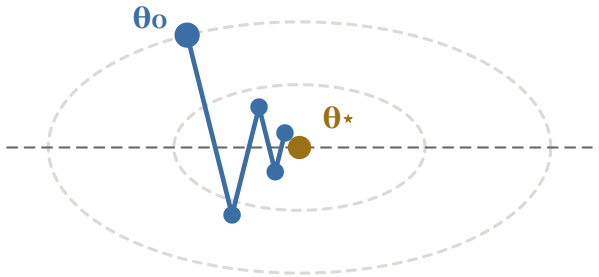


$$L = \tfrac{1}{2}\delta^{\mathsf{T}} H \delta, \qquad g = H\delta, \qquad H = Q\Lambda Q^{\mathsf{T}}$$

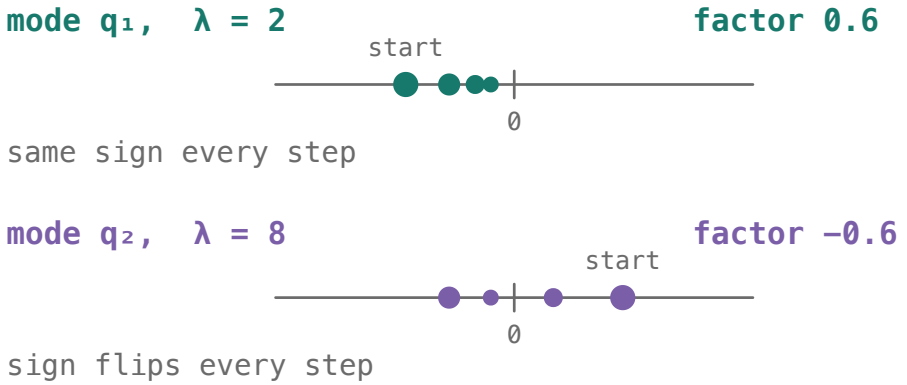
- H symmetric, positive definite, $p \times p$
- Λ $\mathrm{diag}(\lambda_1, \ldots, \lambda_p)$, each $\lambda_i > 0$
- Q orthonormal eigenvectors q_i
- $g = H\delta$ for $\delta \neq 0$, parallel to δ exactly when it is an eigenvector

Gradient Descent and Step Size

Each eigenmode has one multiplier



η = 0.2, four steps



$$\delta_{t+1} = (I - \eta H)\,\delta_t, \qquad \delta_t = \sum_i c_i^{(t)} q_i \;\Rightarrow\; c_i^{(t+1)} = (1 - \eta\lambda_i)\,c_i^{(t)}$$

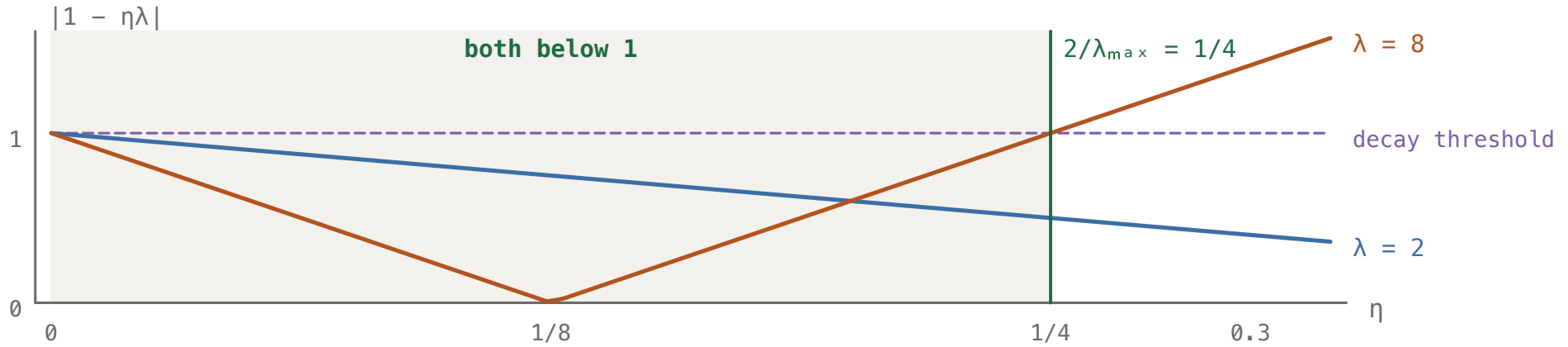
$c_i^{(t)}$ the error's coordinate along q_i

$1 - \eta\lambda_i$ one factor per mode; modes never mix

> If some $\lambda_i = 0$, that factor is 1: the mode keeps its error for every η.

Gradient Descent and Step Size

When does every mode decay?



the steep mode λ = 8 leaves the band first, so it sets the limit

all of this is about one fixed H > 0, and nothing else

$$|1 - \eta\lambda_i| < 1 \text{ for every } i \iff 0 < \eta < \frac{2}{\lambda_{\max}}$$

λ_i the curvatures of one fixed $H \succ 0$

$\lambda_{\max}$ the largest; the steep mode binds first

scope this quadratic only, not any loss, not a network

VOTE

Gradient Descent and Step Size

Which curvature limits the step?

H

2	0
0	8

λ = 2

λ = 8



$H = \mathrm{diag}(2, 8)$. Which η sends the error to zero from every start?

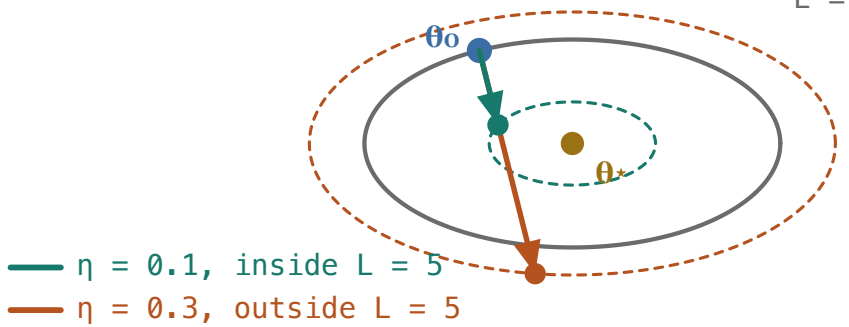
A $0 < \eta < 1$

B $0 < \eta < \frac{1}{4}$

C $0 < \eta \leq \frac{1}{4}$

Gradient Descent and Step Size

A complete two-parameter step



start: $\theta_0 = (0, 0)$, $L = 5$

η	θ_1	$L(\theta_1)$	
0.1	(0.2, −0.8)	0.8	falls
0.3	(0.6, −2.4)	8	rises

same start, same gradient, one number changed

$$g_0 = H(\theta_0 - \theta^\star) = (-2, 8), \qquad \theta_1 = \theta_0 - \eta g_0$$

$\theta^\star = (1, -1)$ the minimiser; $H = \mathrm{diag}(2, 8)$

$\theta_0 = 0$ so $\delta_0 = (-1, 1)$

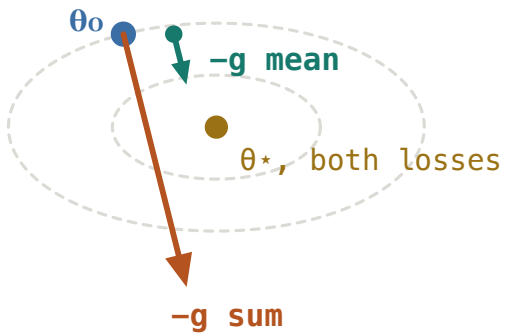
g_0 a column: 8 steep, -2 gentle

WORKED EXAMPLE

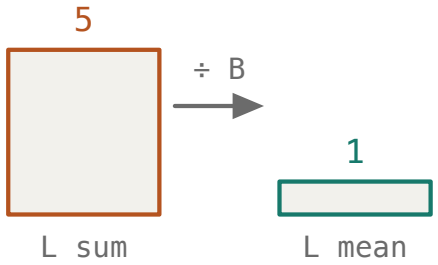
$\eta = 0.1$ takes L from 5 to 0.8. $\eta = 0.3$ raises it to 8: that is **above** the bound, steep factor -1.4, so repeating it diverges.

Gradient Descent and Step Size

Scaling the loss scales the update



same direction, one fifth the length



g sum	g mean
−2	−0.4
8	1.6

B = 5 terms

dividing by five keeps every comparison, so the minimiser is the same point for both losses

$$L_{\text{mean}} = \frac{1}{B} L_{\text{sum}} \quad \Longrightarrow \quad g_{\text{mean}} = \frac{1}{B} g_{\text{sum}}$$

B how many terms the sum has; $B = 5$ here

$\theta^\star$ unmoved: dividing by a positive B keeps every comparison

g_{mean} every entry $1/B$ as large

`every term` is scaled, not just one of them

VOTE

Gradient Descent and Step Size

Which pair reproduces the step?

```
subtract:  η sum × g sum  =  0.1 × (−2, 8)  =  (−0.2, 0.8)
g mean  =  (−0.4, 1.6),    B = 5
```

A	η = 0.1, g mean	subtract η g = ?	
B	η = 0.5, g mean	subtract η g = ?	which row reproduces it?
C	η = 0.5, g sum	subtract η g = ?	

Which pair gives the same update $\theta - \eta g$ as the sum loss?

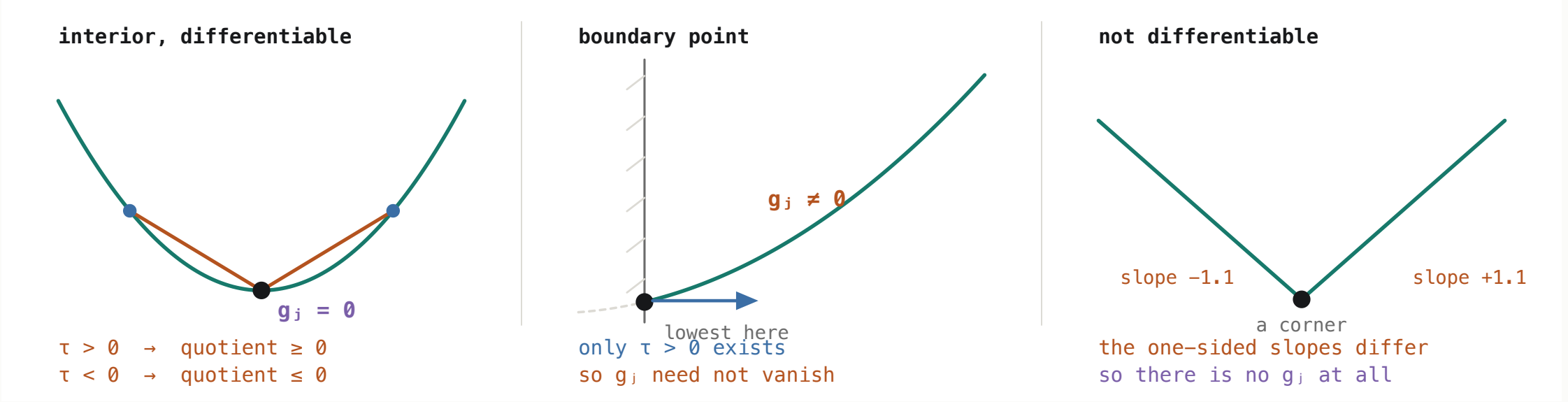
A 0.1 with g_{mean}

B 0.5 with g_{mean}

C 0.5 with g_{sum}

Convexity and Optimality

What zero gradient tells us



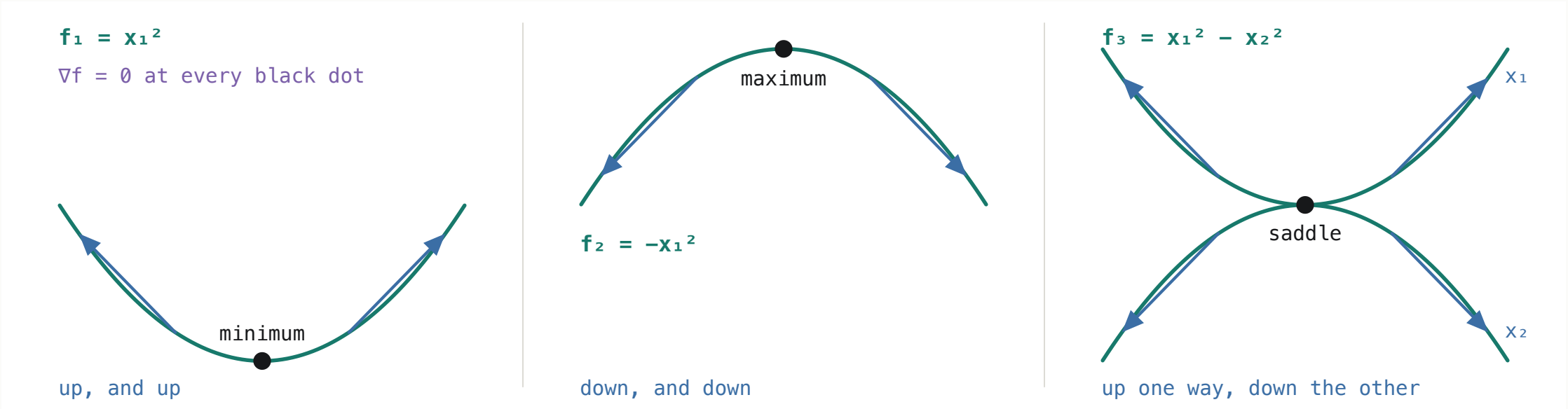
interior, L differentiable, local minimum :

$$\frac{L(\theta + \tau e_j) - L(\theta)}{\tau} \begin{cases} \geq 0, & \tau > 0 \\ \leq 0, & \tau < 0 \end{cases} \Longrightarrow g_j = 0$$

- e_j the j-th coordinate direction
- g_j one limit, so ≥ 0 and ≤ 0 at once
- τ a sufficiently small nonzero probe step, either sign
- a corner no limit, so no g_j to set to zero

Convexity and Optimality

Stationary points have different roles



$$f_1 = x_1^2, \qquad f_2 = -x_1^2, \qquad f_3 = x_1^2 - x_2^2, \qquad \nabla f_k(0) = 0$$

- f_1 up along $+x_1$ and along $-x_1$: a minimum
- f_2 down along both: a maximum
- f_3 up along x_1, down along x_2: a saddle
- $\nabla f_k(0)$ the same zero in all three, so it cannot tell them apart

VOTE

Convexity and Optimality

Is a PSD Hessian at one point enough?

For $f(x) = -x^4$ at 0: $f'(0) = f''(0) = 0$, so $0 \succeq 0$ passes. Is 0

A. a local minimum

B. a local maximum

C. neither

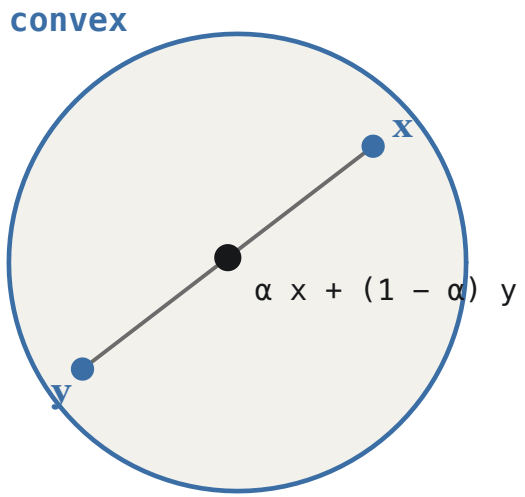
three readings at x = 0, the same for both

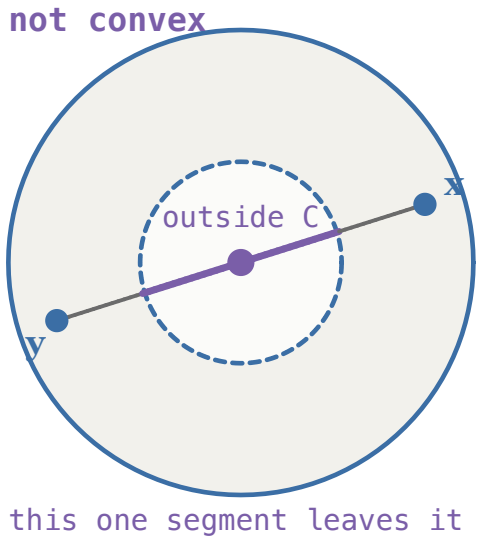
	f(x) = −x⁴	g(x) = x⁴
value	0	0
slope	0	0
curvature	0	0

DEFINITION

Convexity and Optimality

A convex domain contains segments





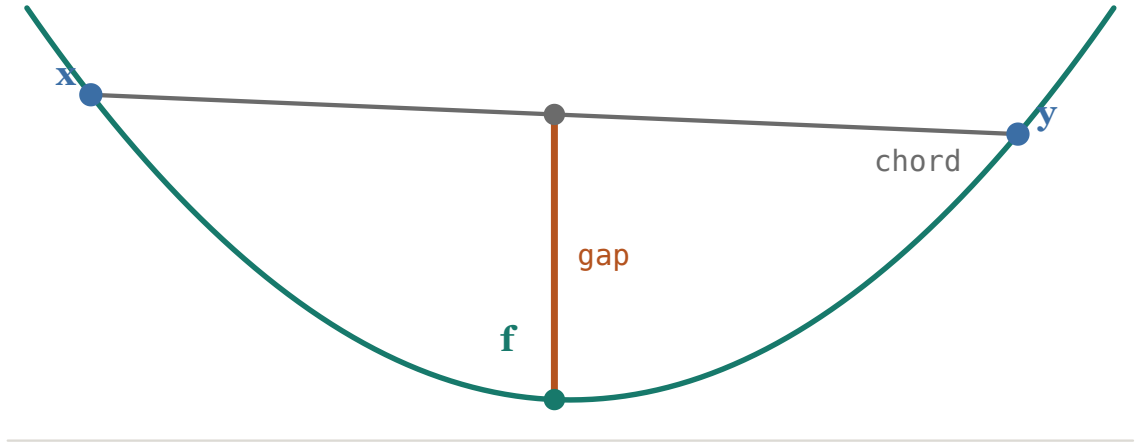
$$x, y \in C, \quad \alpha \in [0,1] \qquad \Longrightarrow \qquad \alpha x + (1-\alpha) y \in C$$

C	a set of points, the domain a convex function needs
$\alpha x + (1-\alpha) y$	every point of the segment, not only its two ends
α	where on the segment you stand: 0 gives y, 1 gives x
not convex	one segment leaving C is enough

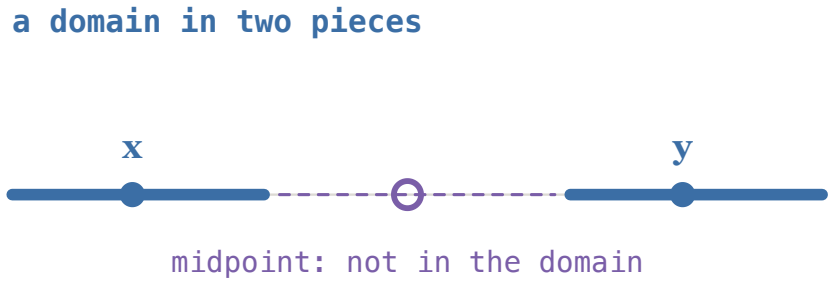
DEFINITION

Convexity and Optimality

A convex function lies below its chords



the chord is never below f



so there is nothing to compare

$$f(\alpha x + (1-\alpha)y) \;\leq\; \alpha f(x) + (1-\alpha) f(y) \qquad \text{for all } x, y \in C,\ \alpha \in [0,1]$$

left side the function at the interpolated point

right side the height of the chord there

the gap chord minus function, never negative

C convex, or some segment leaves it and the left side has no value

VOTE

Convexity and Optimality

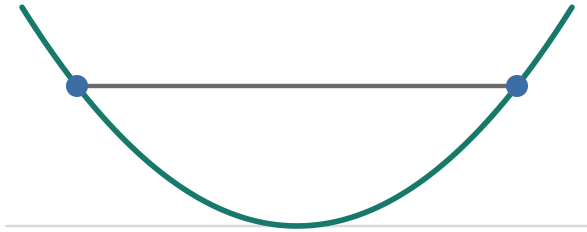
Which picture satisfies the definition?

At $x = -1$ and $y = 1$ with $\alpha = \frac{1}{2}$: which obeys $f(0) \leq \frac{f(-1)+f(1)}{2}$?

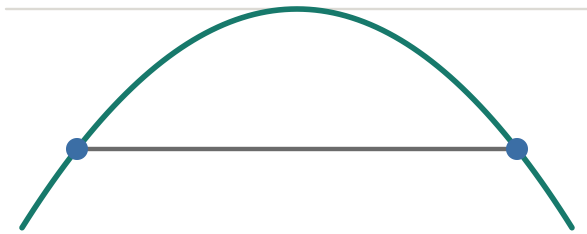
A $f = x^2$

B $f = -x^2$

C both



A: f = x²



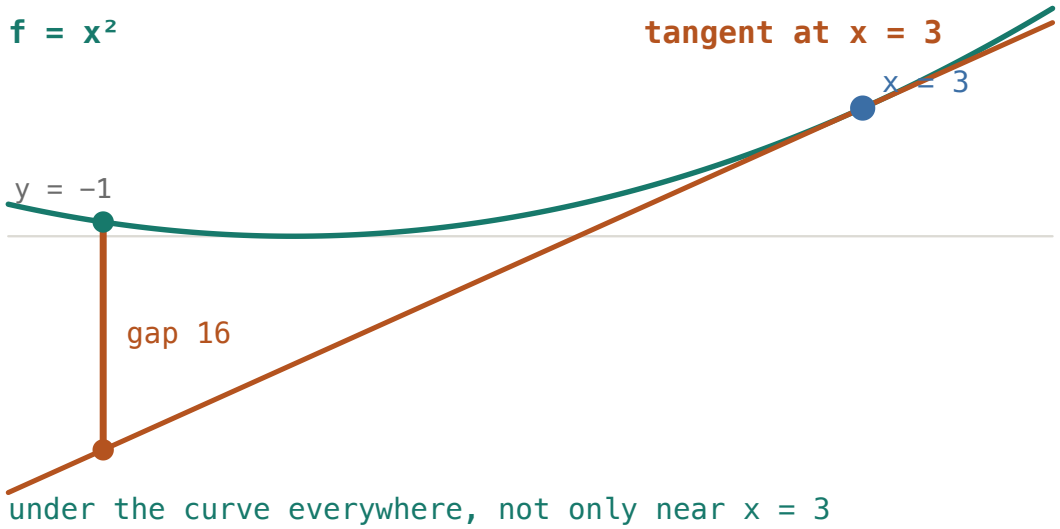
ends at x = −1 and x = 1

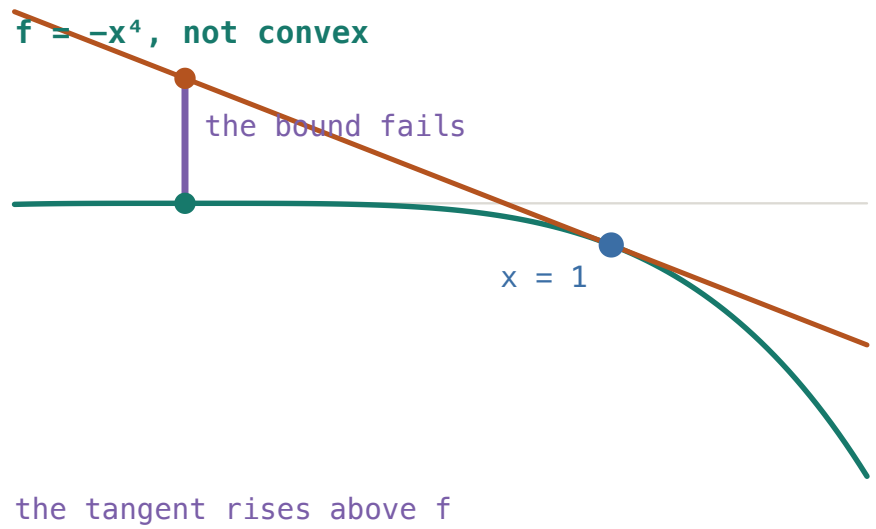
B: f = −x²

KEY INEQUALITY

Convexity and Optimality

A supporting tangent gives a global bound





$$f(y) \;\geq\; f(x) + \nabla f(x)^{\mathsf{T}}(y - x) \qquad \text{for every } x, y \in C$$

$f(y)$ the value anywhere else in C

$f(x) + \nabla f(x)^{\mathsf{T}}(y - x)$ the tangent's height at y

C convex, with f convex and differentiable on it

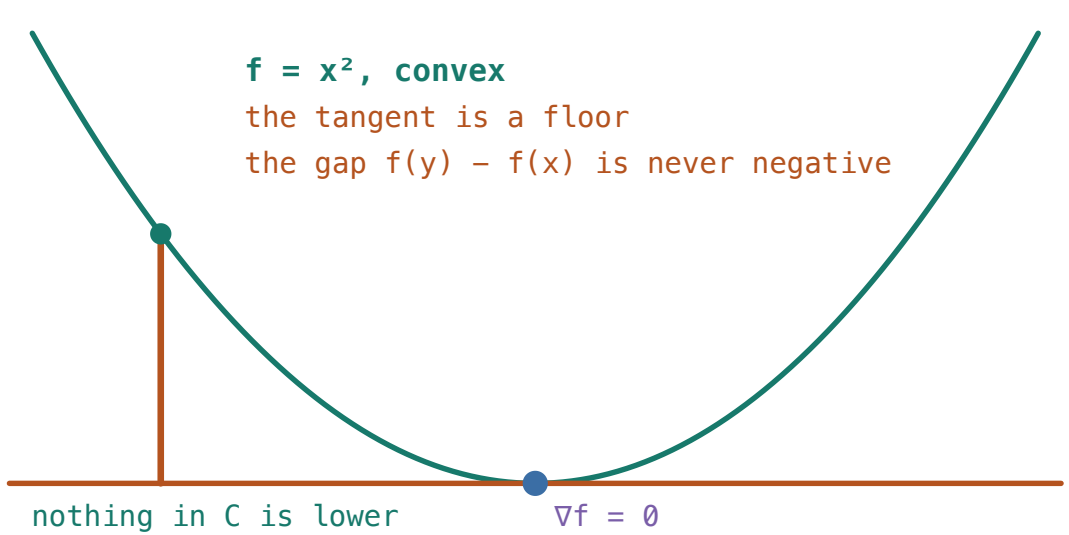
WORKED EXAMPLE

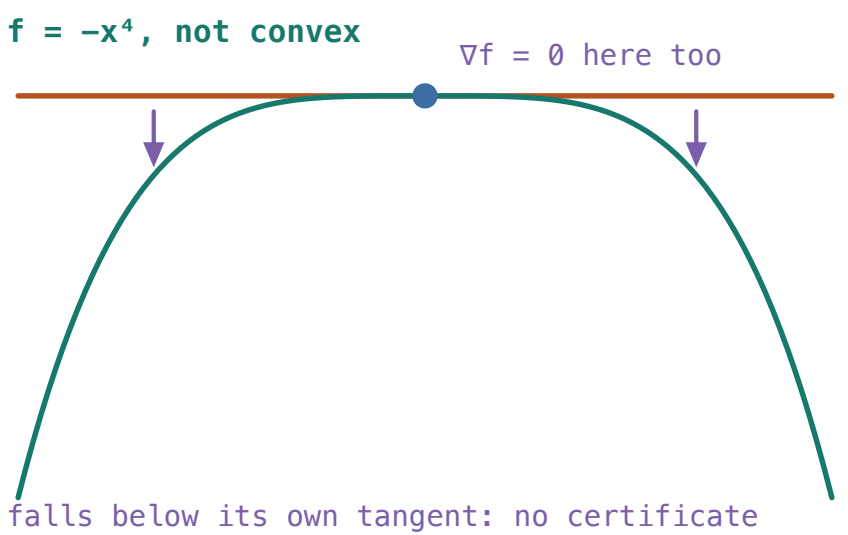
$f = x^2$ at $x = 3$: the tangent is $9 + 6(y - 3)$. At $y = -1$ it reads -15, while $f(-1) = 1$. The gap is 16.

CERTIFICATE

Convexity and Optimality

Zero gradient now certifies a minimum





$$\nabla f(x) = 0 \implies f(y) \geq f(x) + 0^{\mathsf{T}}(y - x) = f(x) \quad \text{for every } y \in C$$

$\nabla f(x) = 0$ the hypothesis, put into the tangent bound

$f(y) \geq f(x)$ global: nothing in C is lower

$0^{\mathsf{T}}(y - x)$ the whole linear term, gone

convex the assumption doing the work; without it $\nabla f = 0$ is necessary only, and only in the interior

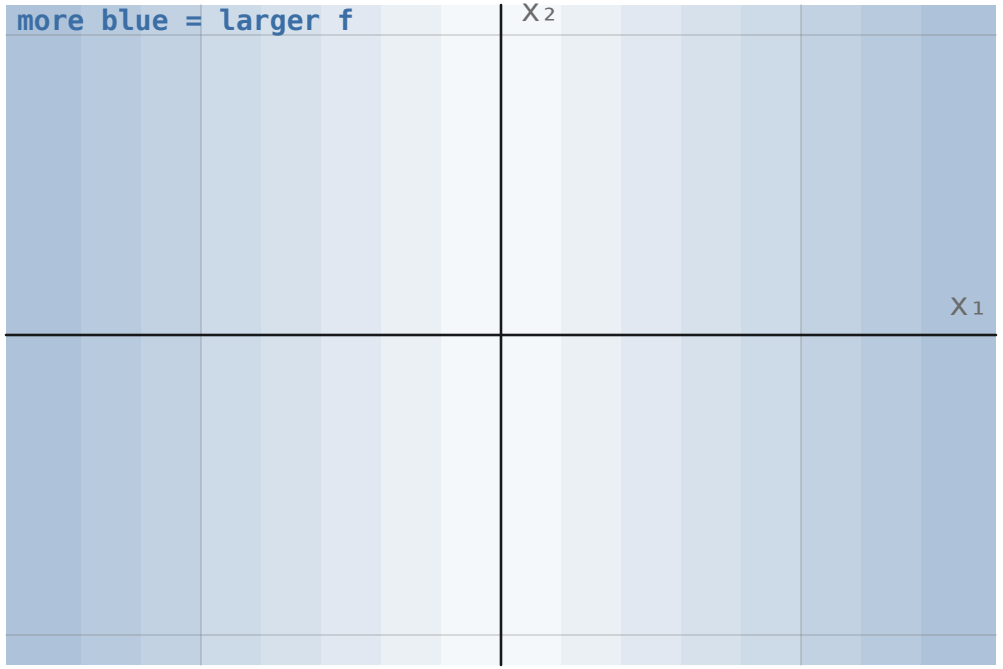
VOTE

Convexity and Optimality

Does convex mean a unique minimizer?

$f(x_1, x_2) = x_1^2$ is convex. Which set is exactly its set of minimizers?

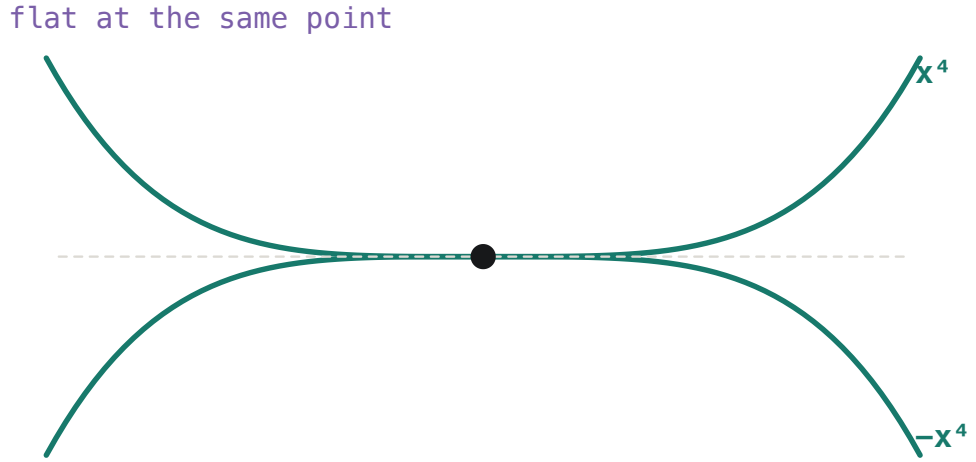
- (A) only $(0, 0)$
- (B) every $(0, t)$
- (C) every point



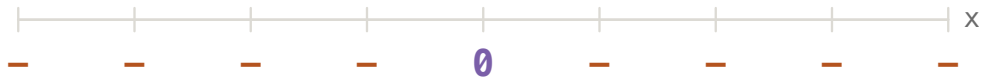
EQUIVALENCE

Convexity and Optimality

Curvature everywhere



$\nabla^2(-x^4) = -12x^2$



negative at every other x: not convex

the dots only sample; the sign is read off the formula

$\nabla^2(x^4) = 12x^2$



nonnegative at every x: convex

$f \in C^2$ on an open convex C : $\quad f$ convex $\iff \nabla^2 f(x) \succeq 0$ for every $x \in C$

$\nabla^2 f(x)$ the Hessian at one point; $\succeq 0$ means PSD there

C^2 twice continuously differentiable; then it is exact

for every x the quantifier that does the work

one point $-x^4$ has $\nabla^2 f(0) = 0$, and $-12x^2 < 0$ everywhere else

Convexity and Optimality

Which conclusion needs which hypothesis

statement	needs	direction
$\nabla L(\theta) = 0$ at an interior local min	interior, differentiable	one way, necessary only
every local min is global	f convex on convex C (no derivative)	one way
$\nabla f(x) = 0 \Rightarrow$ global min on C	f convex AND differentiable on convex C	one way, now sufficient
$\nabla^2 f \succeq 0$ everywhere on C	$f \in C^2$, C open convex	$\Leftrightarrow$ f convex
$\nabla^2 f(x) \succeq 0$ at one x	inconclusive	counterexample $-x^4$ at 0

the first two are different statements, and only the third needs both hypotheses

$$\nabla f(x) = 0 \text{ at an interior local min} \quad \text{vs} \quad \nabla f(x) = 0 \Rightarrow \text{global min}$$

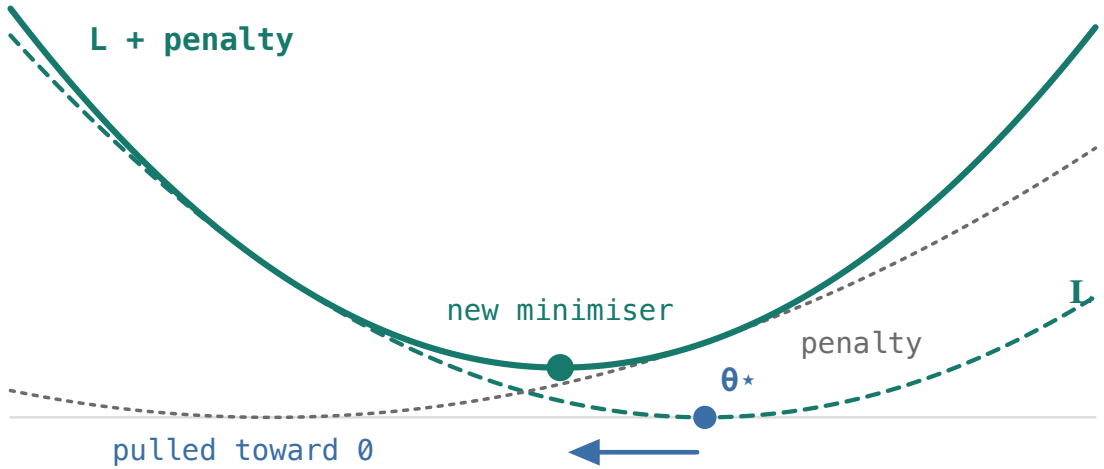
left differentiable, interior; necessary only

right adds convexity; now sufficient and global

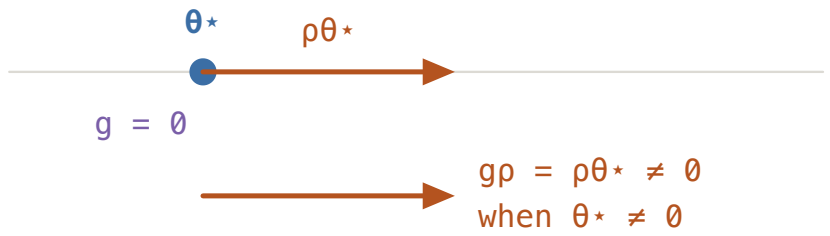
NEW OBJECTIVE

Convexity and Optimality

A quadratic penalty changes the problem



at θ⋆, where L is lowest



so θ⋆ is no longer stationary, unless θ⋆ = 0

$$L_\rho = L + \tfrac{\rho}{2}\|\theta\|^2, \qquad g_\rho = g + \rho\theta, \qquad \rho > 0$$

- L_ρ — L plus a second term
- g_ρ — the new gradient: $\nabla(\|\theta\|^2/2) = \theta$
- $\rho > 0$ — the penalty weight; chosen, not learned
- $\theta^\star$ — where $g = 0$; if $\theta^\star \neq 0$, $g_\rho \neq 0$

A new objective, so in general a new minimiser.

VOTE

Convexity and Optimality

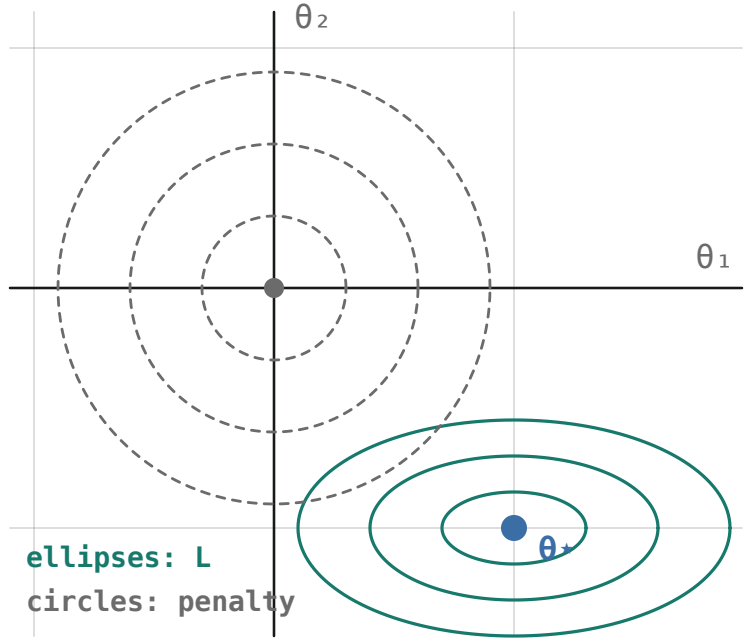
Certifying a quadratic solution

$H = \mathrm{diag}(2, 8)$, $\theta^\star = (1, -1)$, $\rho = 2$. Minimizer of L_ρ?

A $(1, -1)$

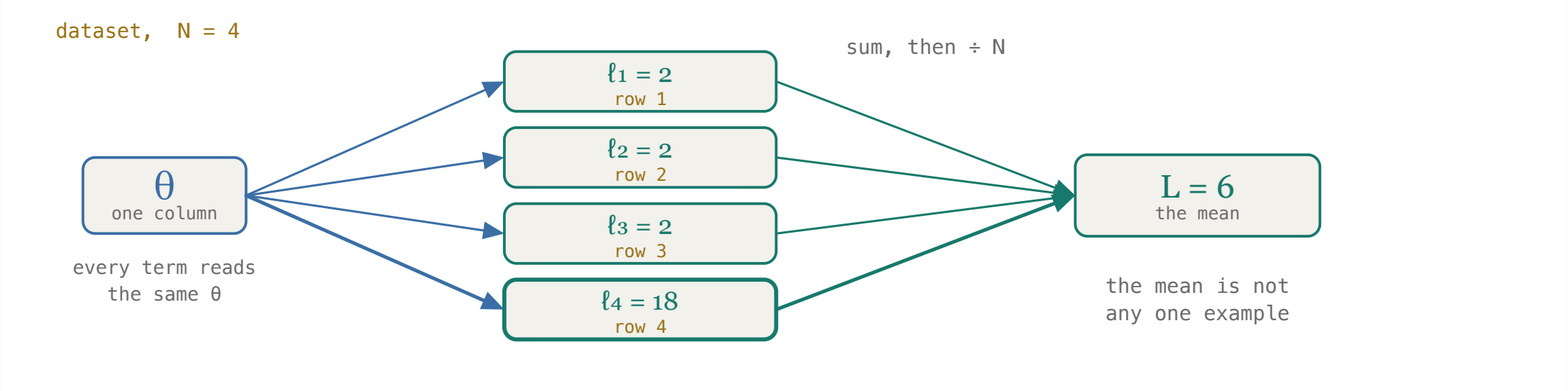
B $(\frac{1}{2}, -\frac{4}{5})$

C $(0, 0)$



Stochastic Gradients and Training

One dataset, many local losses



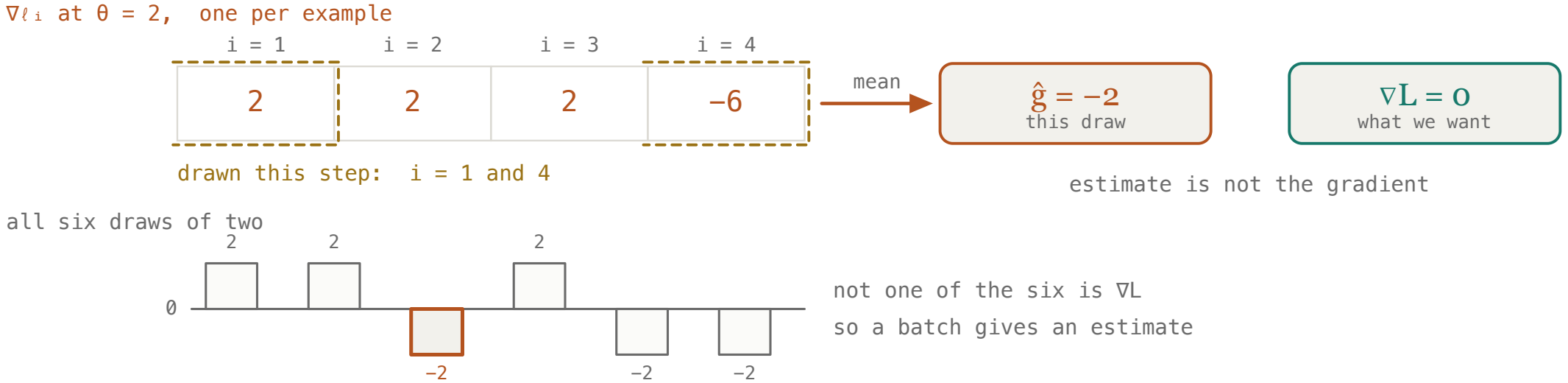
$$L(\theta) = \frac{1}{N} \sum_{i=1}^{N} \ell_i(\theta)$$

N examples in the set, here 4 — $\ell_i(\theta)$ the loss on example i alone

θ read by every term — L one number, whatever N is

Stochastic Gradients and Training

A mini-batch estimates the full gradient



$$\hat{g}_t = \frac{1}{B} \sum_{i \in \mathcal{B}_t} \nabla_\theta \ell_i(\theta_t)$$

$\hat{g}_t$ the estimate, shaped like θ $\quad$ $\mathcal{B}_t$ the indices drawn for step t

B how many, with $B \leq N$ $\quad$ θ_t the same in every term

Drawn uniformly without replacement from $\{1, \ldots, N\}$, independently of θ_t, afresh each step.

VOTE

Stochastic Gradients and Training

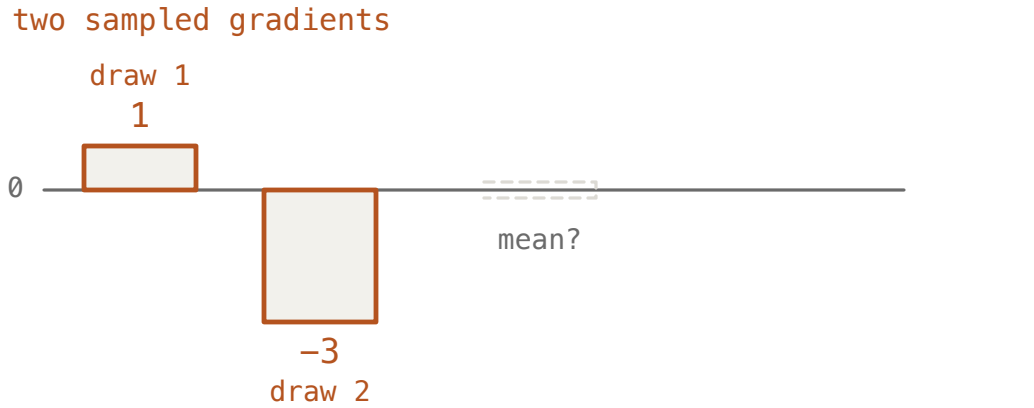
Add gradients before updating

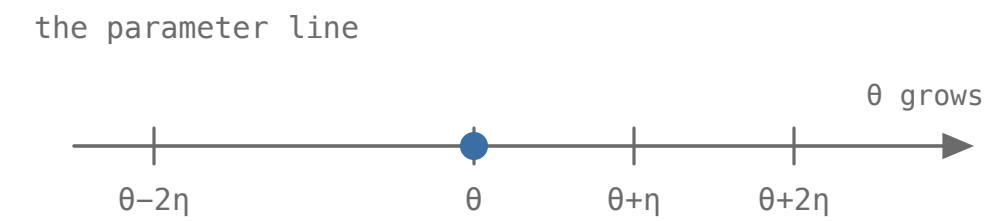
Sampled gradients 1 and -3, with $B = 2$. Under the mean-loss update, where does θ go?

A $\theta - 2\eta$

B $\theta + \eta$

C $\theta + 2\eta$





Stochastic Gradients and Training

Why uniform sampling has the right average

four one-example gradients

i = 1	i = 2	i = 3	i = 4
2	−4	1	5

$\nabla L = 1$
their mean

the full-data gradient

all 4 single draws, w = 1/4

draw i = 1 → 2
draw i = 2 → −4
draw i = 3 → 1
draw i = 4 → 5

average → 1

all 6 draws of two, w = 1/6

−1 3/2 7/2 −3/2 1/2 3

average → 1

not one of the six is 1
that gap is the noise

a nonuniform draw instead, w = 7/10, 1/10, 1/10, 1/10 → 8/5

uniform is what makes the two averages agree

$$\sum_{i=1}^{N} w_i \nabla \ell_i(\theta) = \nabla_\theta L(\theta) \quad \text{when } w_i = \frac{1}{N}$$

w_i probability of drawing index i

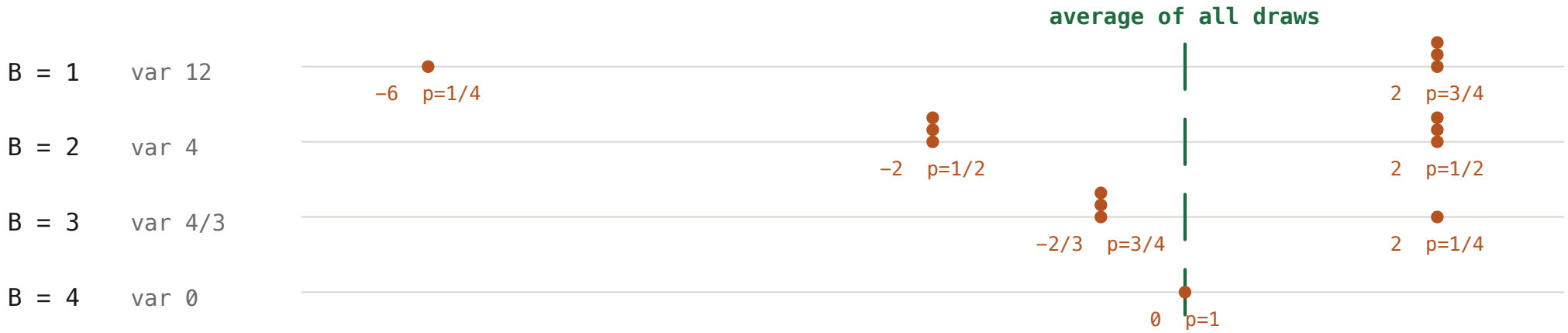
$\nabla \ell_i$ example i's own gradient

∇L the full-data gradient

$w_i \neq \frac{1}{N}$ may change the average

Stochastic Gradients and Training

Unbiased at every B; only the spread changes



θ = 2, c = (0,0,0,8): single-example gradients 2, 2, 2, −6

the tick never moves. The value 2 stays put; its probability falls. The variance falls.

$$\mathbb{E}[\hat{g}_t \mid \mathcal{F}_t] = \nabla L(\theta_t), \qquad \mathrm{Var}(\hat{g}_t \mid \mathcal{F}_t) = \frac{\sigma^2}{B} \cdot \frac{N - B}{N - 1}$$

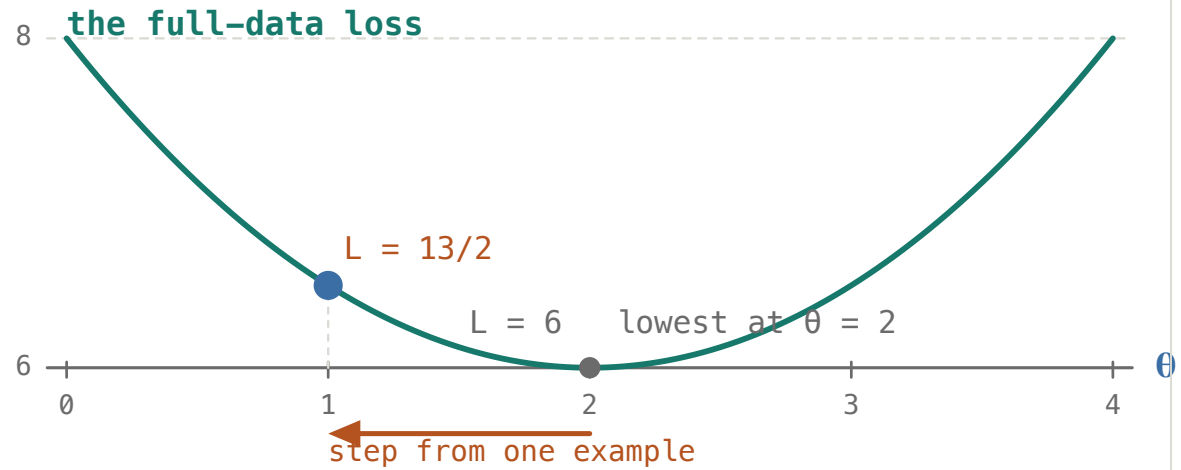
$\mathbb{E}[\cdot \mid \mathcal{F}_t]$ average over batches, given the history; a fresh uniform draw

σ^2 population variance; scalar gradients here

$\frac{N-B}{N-1}$ without replacement, $N > 1$; 0 at $B = N$

Stochastic Gradients and Training

A noisy step can raise the full loss



the four examples

	θ = 2	θ = 1	
i = 1	2	1/2	
i = 2	2	1/2	
i = 3	2	1/2	
i = 4	18	49/2	← the outlier

mean 6 → 13/2

example 1 got better, the mean got worse

$$\ell_i(\theta) = \tfrac{1}{2}(\theta - c_i)^2, \qquad c = (0, 0, 0, 8)$$

c_i example i's own lowest point L the mean of the four

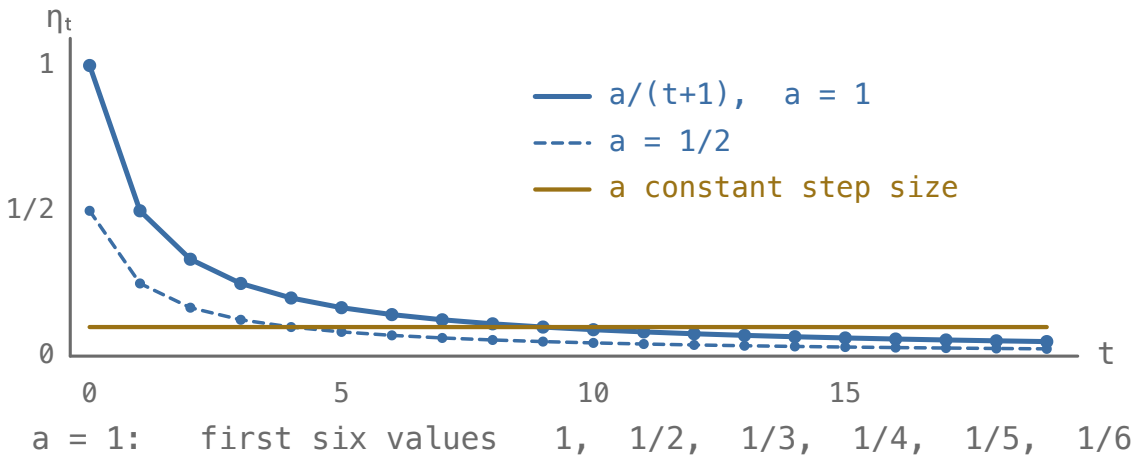
$\nabla \ell_1$ one example's gradient

WORKED EXAMPLE

$\nabla L(2) = 0$, so $\theta = 2$ minimises L, at $L = 6$. Draw example 1, with $\eta = \frac{1}{2}$: $\theta' = 1$ and $L = \frac{13}{2}$.

Stochastic Gradients and Training

First condition: no ceiling fixed in advance



what it says

the budget of step size never runs out, so the bound $G\Sigma\eta_t$ stops giving one fixed ceiling

what it does not say

η_t moves $\eta_t\|\hat{g}_t\|$, not η_t, so $\Sigma\eta_t$ bounds a distance only once $\|\hat{g}_t\| \le G$

it does NOT make the trajectory unbounded, and does not promise the run leaves any ball: if every $\hat{g}_t$ used is 0, θ never moves at all

$$\sum_{t \ge 0} \eta_t = \infty$$

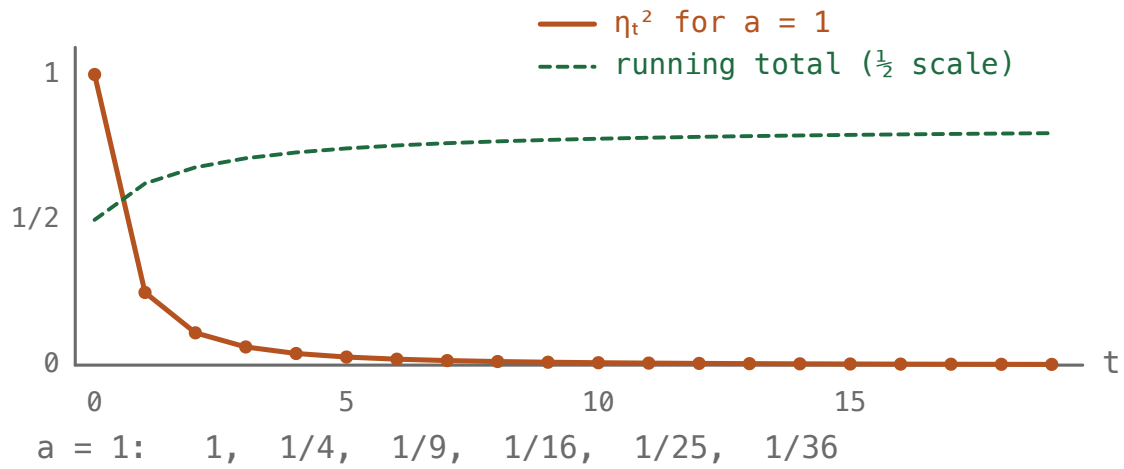
η_t step size at step t, chosen before training

needs $\|\hat{g}_t\| \le G$ before it bounds any distance

$\sum \eta_t$ a sum of step sizes, not a distance travelled

Stochastic Gradients and Training

Second condition: the injected noise settles



what it says

noise contribution: $-\eta_t(\hat{g}_t - g_t)$
under the assumptions below, its signed cumulative sum converges; not a bound on path length or proof of training

what it needs

the error centred given the past, and its conditional second moment bounded

both series are hypotheses, not the theorem

$$\sum_{t \geq 0} \eta_t^2 < \infty$$

$\hat{g}_t - g_t$ the step's error; need not shrink on its own

needs the error centred, and its second moment bounded

$\sum \eta_t^2$ how fast the steps shrink, so the error's effect does

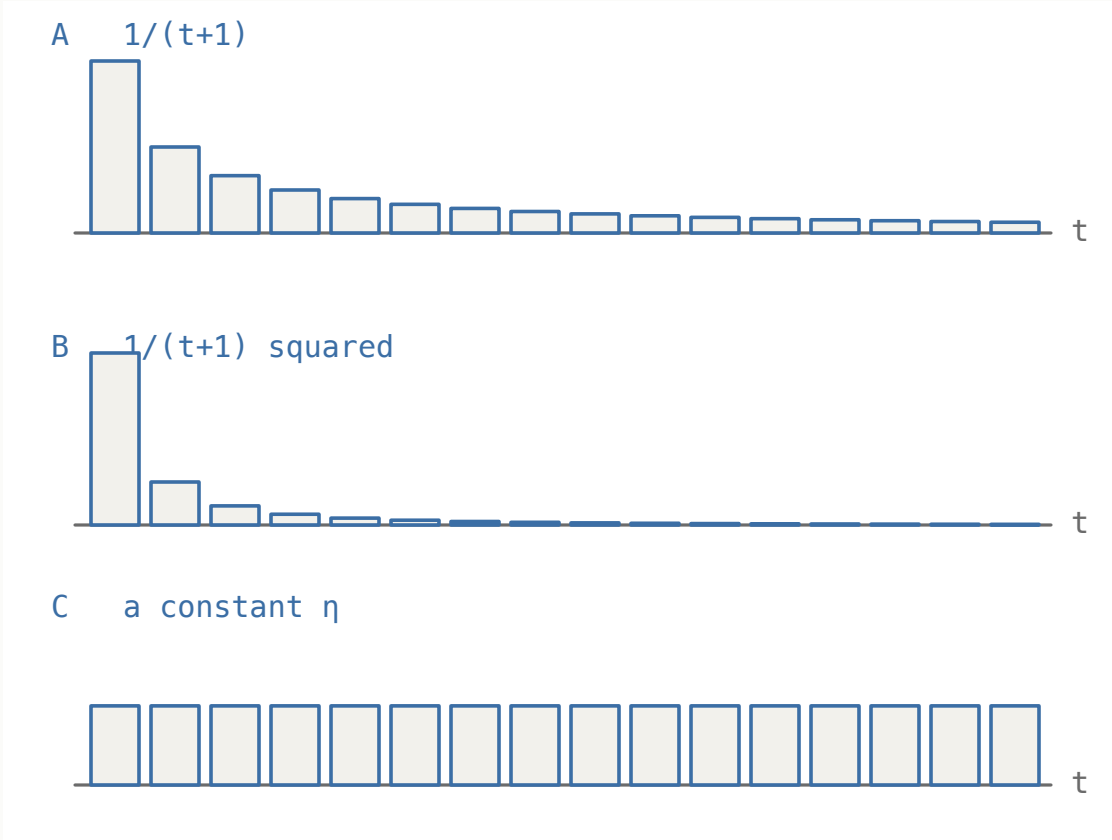
VOTE

Stochastic Gradients and Training

Which schedule passes both series tests?

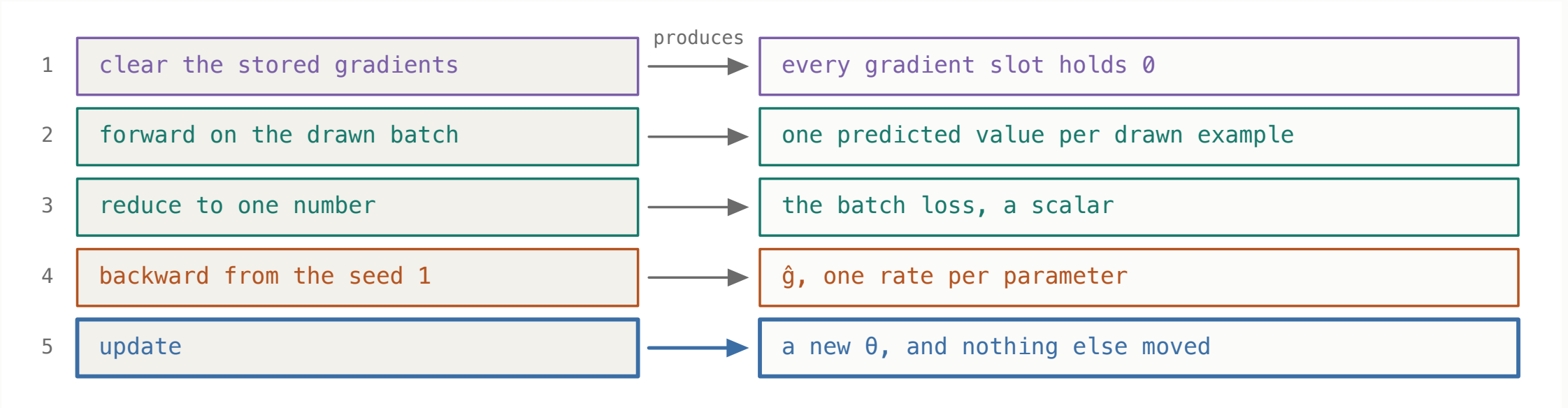
Which η_t passes both $\sum \eta_t = \infty$ and $\sum \eta_t^2 < \infty$?

- **A** $1/(t+1)$
- **B** $1/(t+1)^2$
- **C** a constant $\eta > 0$



Stochastic Gradients and Training

One complete training iteration



$$\theta_{t+1} = \theta_t - \eta_t \hat{g}_t$$

θ_t the column before the step $\hat{g}_t$ what backward produced

η_t the schedule's size at t

Five operations, five different objects, and only the last one assigns to θ.

Stochastic Gradients and Training

Accumulation inside one graph: required

one graph, x used three times

```
r = x*x + x,    x = 2
```



every use contributes one term; the slot adds them

the += is the chain rule here, and is required

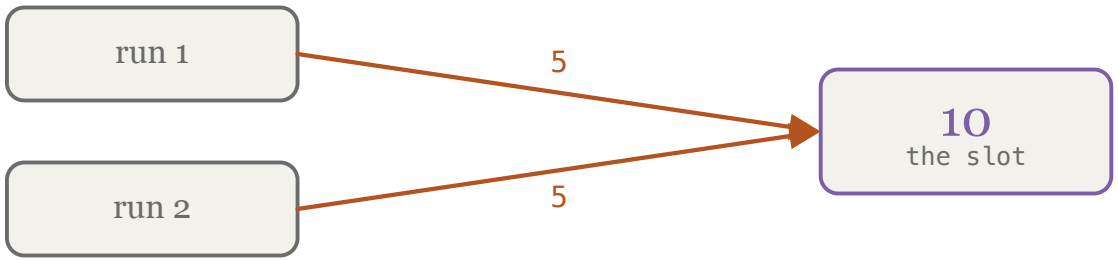
$$\bar{u} \mathrel{+}= \bar{v}\,\frac{\partial v}{\partial s}$$

- $\bar{u}$ the gradient slot for u
- $\bar{v}$ what arrived downstream
- $\partial v/\partial s$ slot s receives u; other slots fixed
- $+=$ add every use; each term carries its own $\partial v/\partial s$

Stochastic Gradients and Training

Accumulation across two backward passes: a leftover

two fresh graphs, same θ, nothing cleared



= ∇(2L), not ∇L

correct, but of 2L

each run alone would give 5

a leftover, and a bug unless you asked for it

$$\text{slot} = \hat{g}_1 + \hat{g}_2 = 10, \qquad \hat{g}_1 = \hat{g}_2 = 5$$

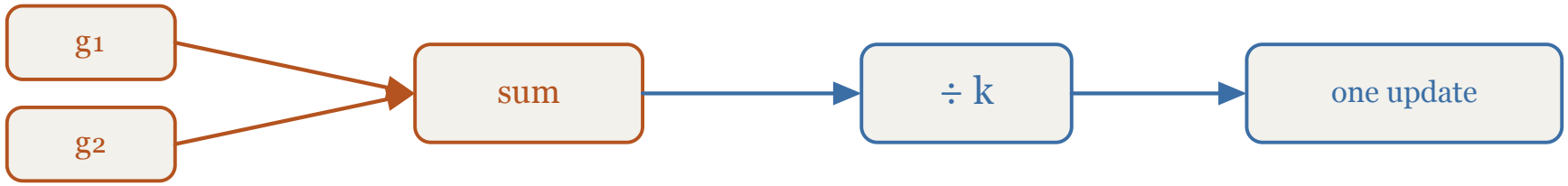
$\hat{g}_i$ run i's gradient, both at the same θ `slot` never cleared between the two

Same arithmetic as the page before, different scope.

Stochastic Gradients and Training

Accumulation on purpose: micro-batches

k micro-batches at one θ, one update



equal sizes: ÷k gives the mean; plain SGD can instead use η/k

sizes 1 and 3, means 0 and 4: ÷k gives 2, the union mean is 3

a policy, not a mistake

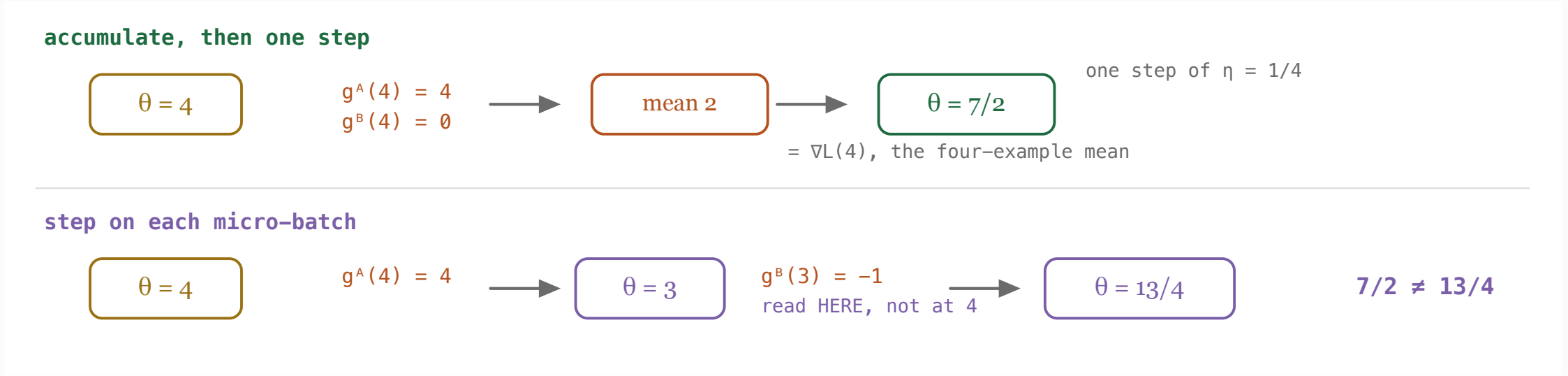
$$\frac{\sum_m B_m \hat{g}_m}{\sum_m B_m} \xrightarrow{B_m \equiv b} \frac{1}{k} \sum_{m=1}^{k} \hat{g}_m$$

$\hat{g}_m$ micro-batch m's mean gradient

B_m its size; equal sizes collapse the weights

Stochastic Gradients and Training

Accumulate once, or step per micro-batch?



$$\theta - \eta\,\frac{g_A(\theta) + g_B(\theta)}{2} \quad \underset{\text{in general}}{\neq} \quad \big(\theta - \eta g_A(\theta)\big) - \eta\, g_B\big(\theta - \eta g_A(\theta)\big)$$

`left` both read at θ; one step on the full-data gradient

`right` g_B read after the first step, so somewhere else

VOTE

Stochastic Gradients and Training

Locate the training-loop bug

Gradients are nonzero, yet θ never changes. Which operation is the loop missing?

- (A) the update
- (B) a second backward
- (C) a larger batch

```
1.  clear the gradient store
2.  forward on the batch
3.  form the loss
4.  backward
5.  ?
```

θ before		θ after	the gradient store
2.00	=	2.00	2

these two are equal, and the store is not zero

Stochastic Gradients and Training

A different omission: clearing, over two steps

both loops update, η = 1/2, θ starts at 2

	store at step 1	θ after step 1	store at step 2	θ ends
clearing	2	1	−7	**9/2**
never clearing	2	1	−5	**7/2**

both read their second gradient at θ = 1, because both moved there

the store is what differs: −7 alone, or −5 with the leftover 2 still in it

$$\text{slot}_t = \sum_{s \le t} \hat{g}_s \quad \text{instead of} \quad \hat{g}_t$$

$\hat{g}_s$ read at the θ the loop is standing on

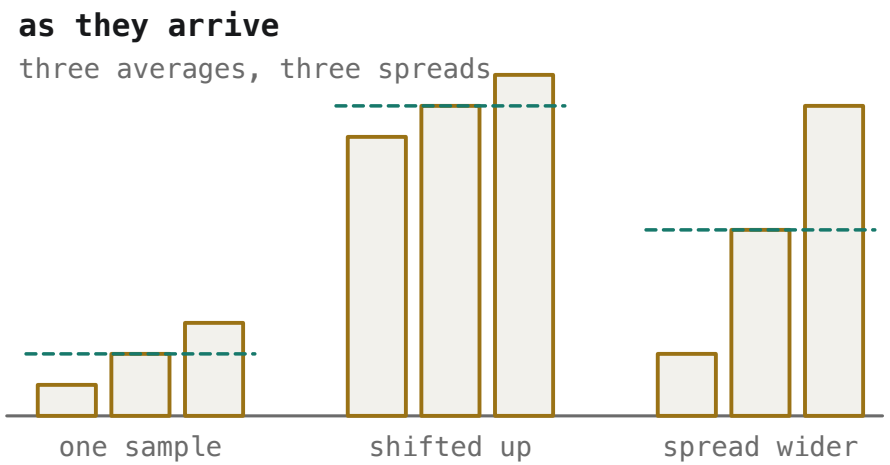
`slot` a running sum when line one is missing

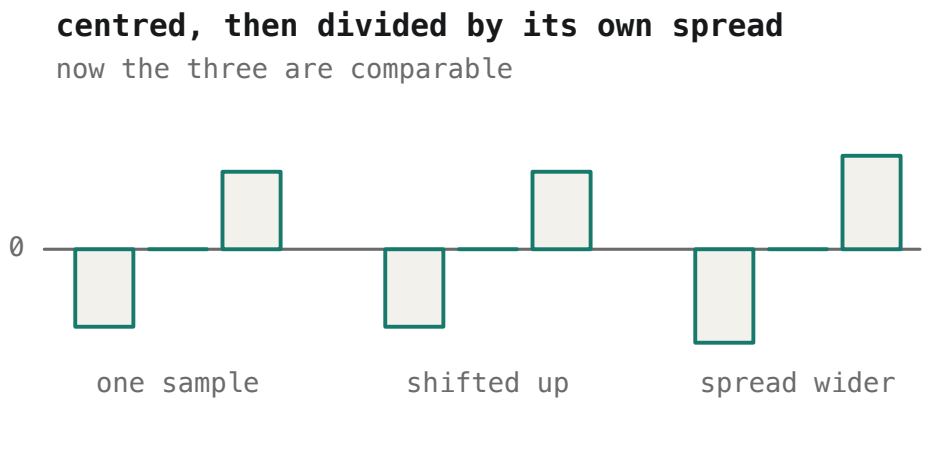
Both loops update. Only line one differs, and they end apart.

NEW BLOCK

Layer Normalization

Controlling feature offset and scale





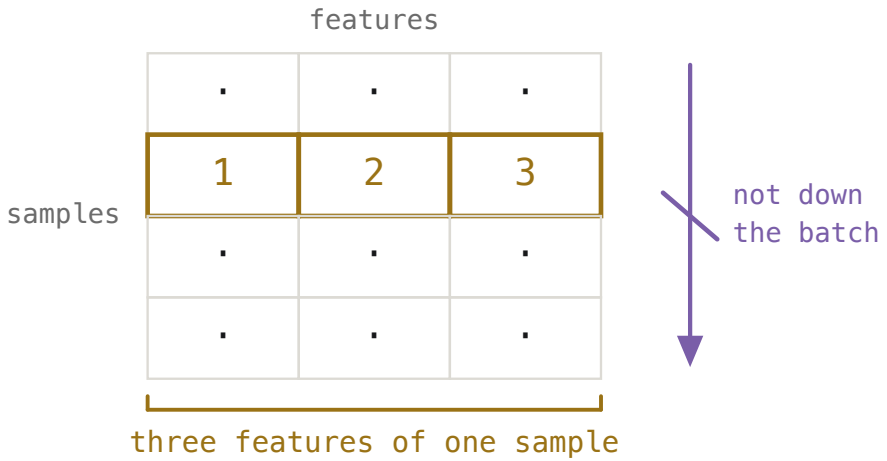
$$h \in \mathbb{R}^d \longmapsto z \in \mathbb{R}^d \quad \text{one sample at a time}$$

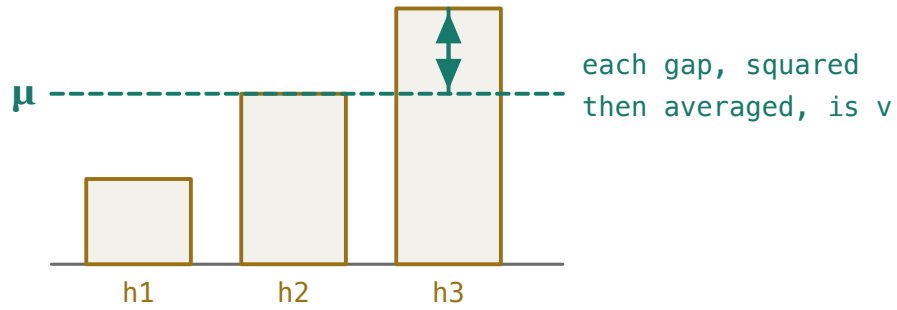
- h one sample's feature column, $d \times 1$
- z what the next layer reads, same shape
- d how many features, never how many samples
- `two per feature` a learned scale and shift, kept at the end

It controls feature offset and scale; faster training is not guaranteed.

Layer Normalization

Mean and variance over features





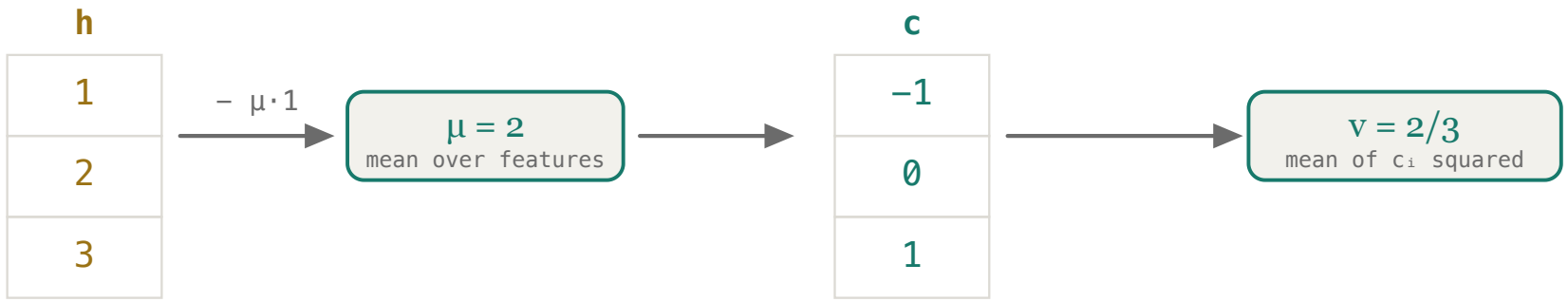
$$\mu = \frac{1}{d}\sum_{j=1}^{d} h_j, \qquad v = \frac{1}{d}\sum_{j=1}^{d}(h_j - \mu)^2$$

h_j	feature j of this one sample
$\mu,\ v$	two scalars, built from this sample alone
$j = 1, \ldots, d$	the sum runs over features; no sample index appears in it

Divided by d, not $d - 1$: the population variance of the d numbers in front of us.

Layer Normalization

Step one: centre the features



−1 + 0 + 1 = 0: centring always gives a column that sums to zero

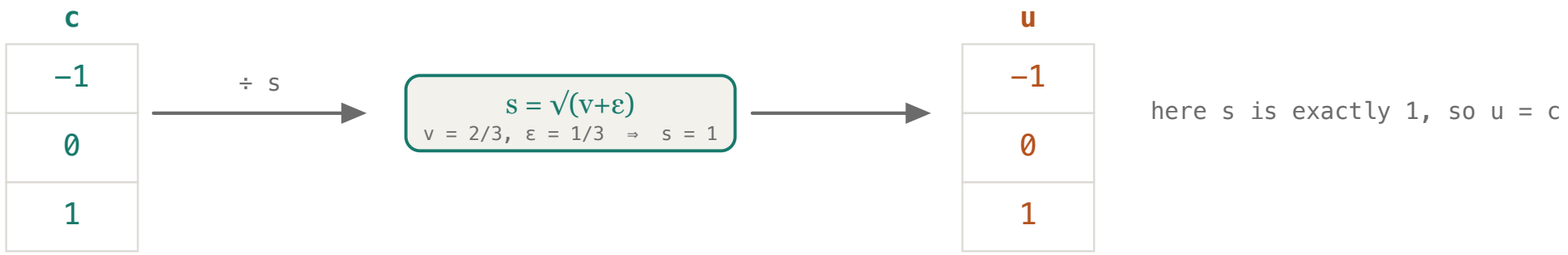
one sample at a time. Nothing here looks at any other sample.

$$\mu = \frac{1}{d}\sum_i h_i, \qquad c = h - \mu\mathbf{1}, \qquad v = \frac{1}{d}\sum_i c_i^2$$

- h one sample's features, $d \times 1$
- μ mean over features, not over the batch
- c the centred features; $\sum_i c_i = 0$ always
- v their mean square, this sample's spread

Layer Normalization

Step two: divide by s, and why ε is there



$$u = \frac{c}{s}, \qquad s = \sqrt{v + \varepsilon}, \qquad \varepsilon > 0$$

- s one positive number for this sample
- u the normalized features; their statistics are the next vote
- ε fixed, so s is never 0 on a flat sample

Layer Normalization

Step three: the learned scale and shift

u		γ		β		z
−1		2		0		−2
0	⊙	−1	+	1	=	1
1		3		−1		2

one component in full

$z_1 = \gamma_1 u_1 + \beta_1$

$= 2 \cdot (-1) + 0$

$= -2$

centring loses μ; learned γ and β cannot restore that missing mean

$$z = \gamma \odot u + \beta, \qquad z_i = \gamma_i u_i + \beta_i$$

- γ, β learned, $d \times 1$ each, one pair for all samples
- z what leaves the block, $d \times 1$
- $\odot$ entry by entry, not a matrix product

VOTE

Layer Normalization

What is exactly zero or one?

```
new h = (0, 0, 3), epsilon = 2
Find mean, centered values, scale.
Use features, not sample rows.
```



New $h = (0, 0, 3)^\top$, $\varepsilon = 2$. Compute c, s, u. Then u has

A mean 0, $\text{mean}(u^2) = 1$

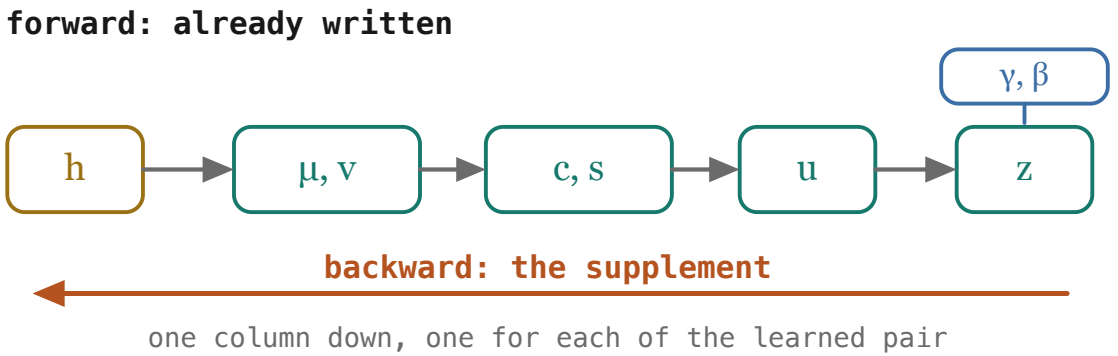
B mean 0, and $\text{mean}(u^2) < 1$

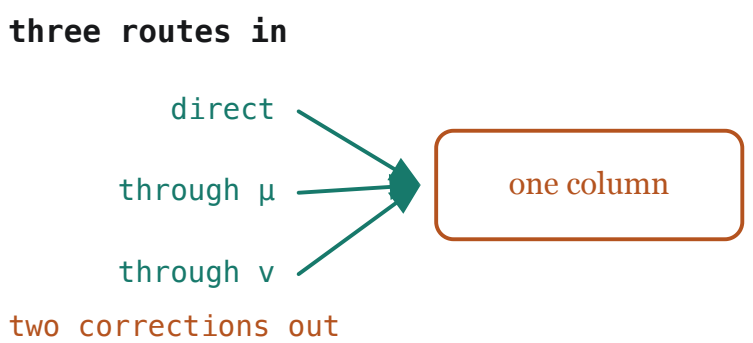
C no fixed mean

SUPPLEMENT

Layer Normalization

Where the backward pass is written down





$$\nabla_z L \longmapsto \nabla_h L,\ \nabla_\gamma L,\ \nabla_\beta L$$

$\nabla_z L$ from above, $d \times 1$

$\nabla_h L$ back down; three routes build it

$\nabla_\gamma L,\ \nabla_\beta L$ $d \times 1$ each, like the pair

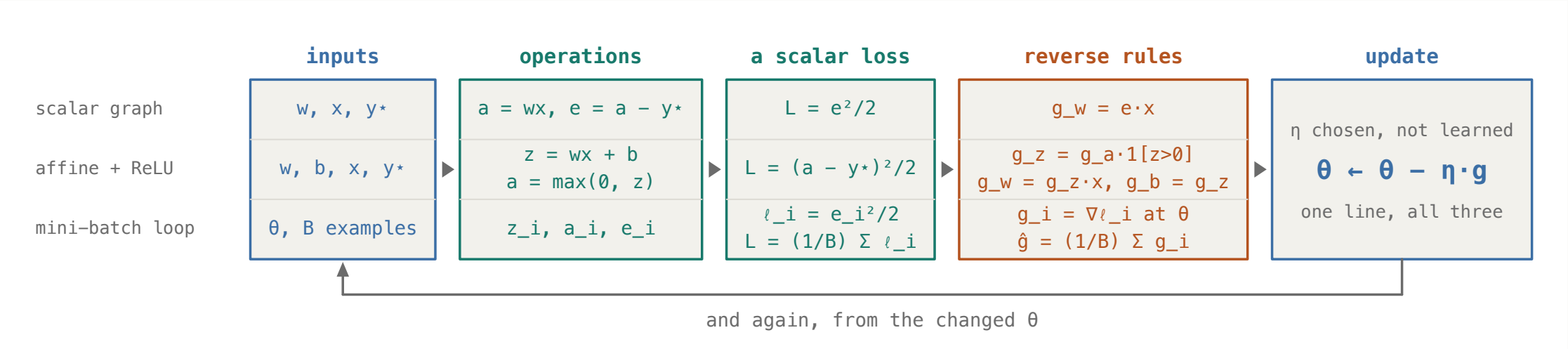
- two correction terms, and a zero-sum test that passes a wrong formula
- wanted again in Lecture 10, and by a gradient check

Eleven pages, with the worked numbers and its own three votes: jingz6676.github.io/rob6013/layernorm-backward.html

PUTTING IT TOGETHER

A Complete Learning Step

The same chain rule in every block



$$\bar{L} = 1, \qquad \bar{u} \mathrel{+}= \bar{v}\,\frac{\partial v}{\partial s}, \qquad \theta_{t+1} = \theta_t - \eta_t g_t$$

- $\bar{L} = 1$ the only seed a reverse pass needs
- $\bar{v}$ what arrived from downstream
- $\partial v / \partial s$ slot s receives u; other slots fixed
- η_t enters here, nowhere earlier

WORKED EXAMPLE

A Complete Learning Step

From data to a checked parameter update

$x = (1, 2),\ y^\star = (1, 3), \quad w = 1,\ b = 0,\ \eta = 0.1$

$$e_i = w x_i + b - y_i^\star, \qquad L = \tfrac{1}{4} \sum_i e_i^2$$

$x_i, y_i^\star$	data
w, b	parameters
e_i, L	error; mean loss

Forward → backward → update → recheck.

$$g_w = \tfrac{1}{2} \sum_i e_i x_i = -1, \qquad g_b = \tfrac{1}{2} \sum_i e_i = -0.5$$

g_w, g_b	shared gradients

$L = 0.25 \ \rightarrow \ (w', b') = (1.1, 0.05) \ \rightarrow \ L' = 0.14625$

VOTE

A Complete Learning Step

Choose the steps for changed data

$x = (1, 3),\ y^\star = (2, 2),\ w = 1,\ b = 0,\ \eta = 0.1$. Find L, g_w, g_b; choose (w', b', L').

$$e_i = w x_i + b - y_i^\star, \qquad L = \frac{1}{4} \sum_i e_i^2$$

e_i error

w, b parameters

L mean loss

A $(0.9, 0, 0.5)$

B $(0.9, 0, 0.425)$

C $(0.95, 0, 0.45625)$

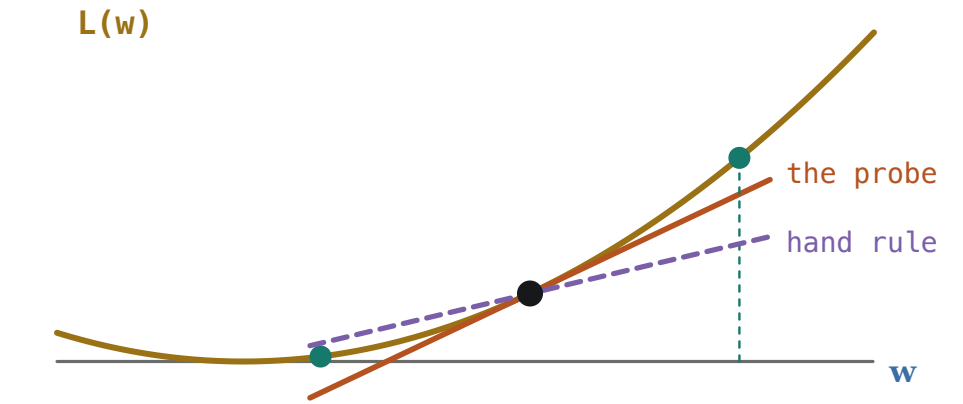
VOTE

A Complete Learning Step

Diagnose before changing the optimizer

Loss rising; the hand rule and a smooth probe disagree. First action?

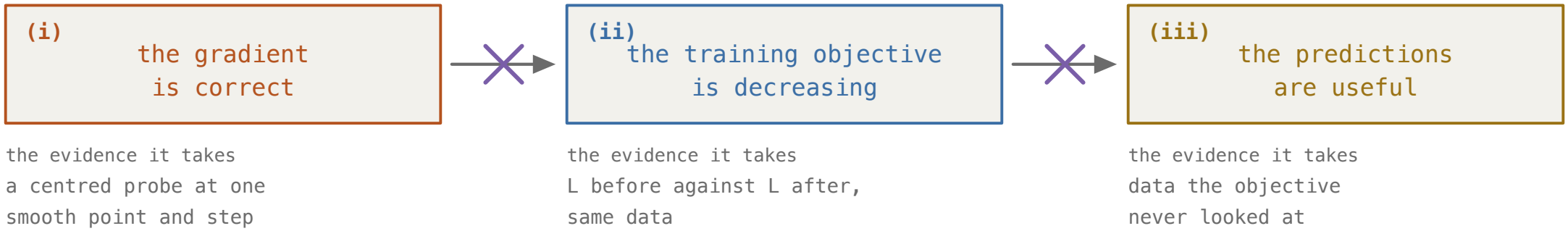
- (A) swap the optimizer
- (B) inspect derivative, fixture, reduction
- (C) raise η





A Complete Learning Step

Three distinct claims about training



$$\text{(i) } g = \nabla_\theta L \qquad \text{(ii) } L(\theta_{t+1}) < L(\theta_t) \qquad \text{(iii) } \ell \text{ at an unseen } x$$

(i) a derivation gives the rule; a probe tests the code

(ii) a step question; η and curvature decide it

(iii) a data question, asked where L never looked

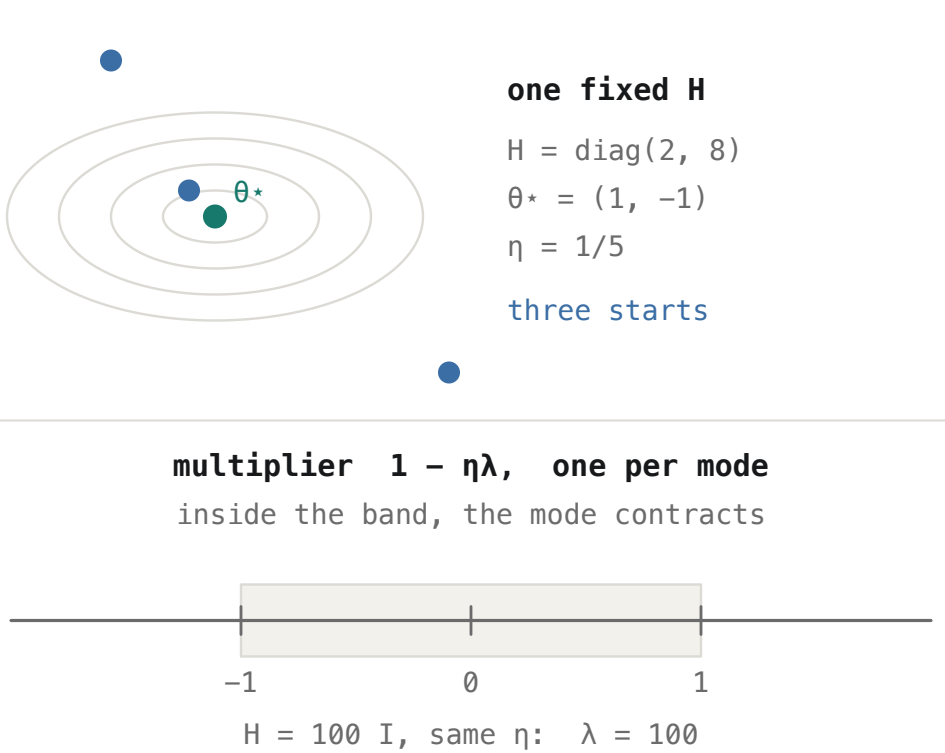
VOTE

A Complete Learning Step

Exit check and next connection

Which guarantee was proved?

- (A) any neural network, $\eta < 2/\lambda_{\max}$ at the start
- (B) a fixed $H \succ 0$ quadratic, that bound
- (C) correct gradients, good test predictions



References

Where these rules come from

1. Boyd & Vandenberghe. *Convex Optimization*, ch. 2–3, 9. Cambridge, 2004.
2. Goodfellow, Bengio & Courville. *Deep Learning*, ch. 8, §8.3.1. MIT Press, 2016. Momentum and Adam, later in that chapter, are optional reading.
3. Baydin et al. *Automatic Differentiation in Machine Learning: a Survey*. JMLR, 2018.
4. Ba, Kiros & Hinton. *Layer Normalization*. arXiv, 2016.

References

Implementations, and the course standards

5. Vaswani et al. *Attention Is All You Need*, §3.2.1. NeurIPS, 2017. The map a later lecture derives; reading ahead, not used here.
6. *PyTorch autograd tutorial*. PyTorch documentation, read 2026-10-04.
7. Karpathy, A. *micrograd*. GitHub. An implementation to read, and to license-check before copying.

Course standards: ROB-GY 6013, *Notation and Conventions* · columns, the gradient row, the batch layout. ROB-GY 6013, *Textbooks and Reading Guide* · what each source is for, and where its notation differs.