Making Deep Networks Trainable

Activations → initialization → normalization → residual connections

Prof. Nipun Batra

ES 667 · Deep Learning · IIT Gandhinagar

Backprop found the gradients; depth may erase them

L4 · BACKPROP

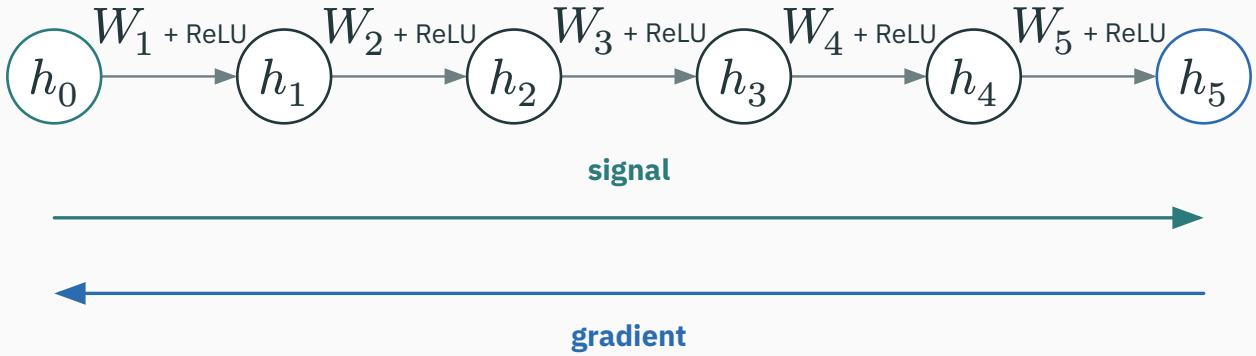
computes $\nabla_\theta \mathcal{L}$

L5 · OPTIMIZATION

chooses the parameter update

L6 · TRAINABILITY

asks whether signals and gradients survive depth



Today the computation rule stays fixed; we redesign how information travels through it.

Commit: which start survives five ReLU layers?

Q QUESTION

ANCHOR · five width-4 layers · ReLU after every affine map · $q_0 = 1 \cdot g_5 = 1$

Which zero-mean weight distribution should keep both end-to-end RMS gauges closest to 1 at initialization?

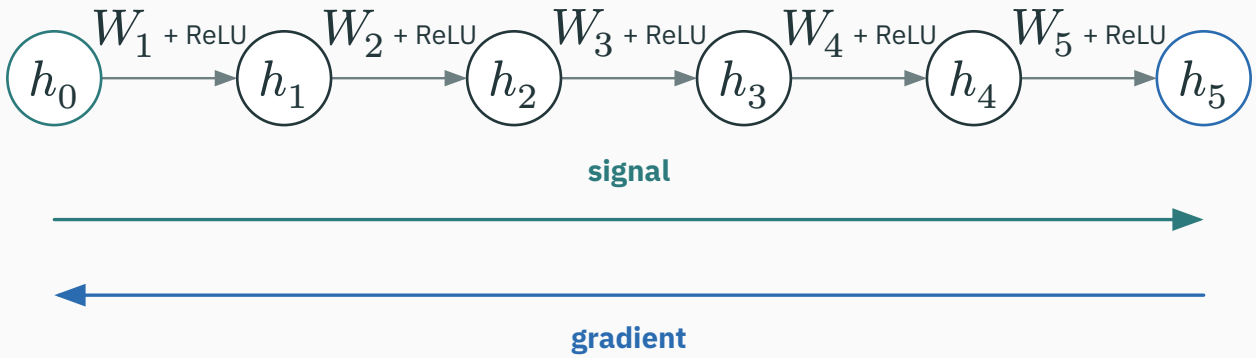
A. Xavier normal with $\mathrm{Var}(w) = \frac{1}{4}$

B. Xavier uniform with the same variance $\frac{1}{4}$

C. He normal with $\mathrm{Var}(w) = \frac{1}{2}$

D. All weights equal to zero

Keep your bet; every section updates one ledger



We will calculate the same chain, then add one control at a time:

activation gate → matched init → normalization → residual path

> Do not change your answer yet. We return to it after the activation gate determines the matching initialization.

Outline: keep signals usable through depth

V VISUAL

Predict the drift → measure both RMS gauges → change the earliest plausible cause → measure again.

forward signal backward gradient chosen gain / reset preserved / identity failure

One gauge, two directions, no width drift

V VISUAL

For width d_ℓ, average over coordinates—not a raw vector norm:

$$q_\ell := \frac{1}{d_\ell} \sum_i \left(h_{\ell,i}\right)^2, \quad r_\ell := \sqrt{q_\ell}$$

$$g_\ell := \frac{1}{d_\ell} \sum_i \left(\overline{h}_{\ell,i}\right)^2, \quad s_\ell := \sqrt{g_\ell}.$$

FORWARD

r_ℓ is activation RMS; q_ℓ is its squared ledger

BACKWARD

s_ℓ is gradient RMS; g_ℓ is its squared ledger

ANCHOR · five width-4 layers · ReLU after every affine map · $q_0 = 1 \cdot g_5 = 1$

A harmless local factor becomes a depth failure

If one layer multiplies a **squared** gauge by c, then five layers multiply it by c^5.

per-layer c	c^1	c^5	RMS factor $\sqrt{c^5}$
0.5	0.5	0.03125	0.17678
1	1	1	1
2	2	32	5.65685

Depth turns a small scale mismatch into a visible vanishing or exploding trace.

The repair order follows causality

V VISUAL

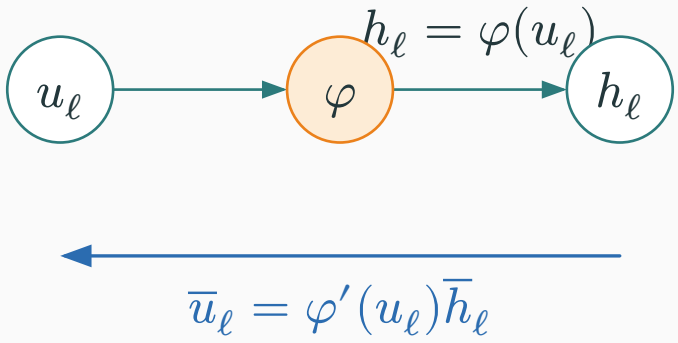
activation	sets the local nonlinear gate
initialization	matches weight gain to that gate at $t = 0$
normalization	re-establishes a reference scale while training
residual connection	adds a short signal and gradient route

Activation comes first because its gate determines the scale that initialization must match.

The activation is a gradient gate

Every nonlinearity acts in both passes

V VISUAL



The curve shapes the representation; its derivative gates the returning gradient.

Sigmoid becomes nearly flat in either tail

$$\sigma(u) = \frac{1}{1 + e^{-u}}, \quad \sigma'(u) = \sigma(u)(1 - \sigma(u)).$$

u	-5	0	5	meaning
$\sigma(u)$	0.0067	0.5	0.9933	bounded output
$\sigma'(u)$	0.00665	0.25	0.00665	gradient gate

Even its best slope is only $\frac{1}{4}$; saturation makes the slope far smaller.

One saturated sigmoid blocks more than 99%

D DERIVATION

Take $u = 5$ and upstream gradient $\overline{h} = 1$.

$$h = \sigma(5) \approx 0.9933$$

$$\sigma'(5) = h(1 - h) \approx 0.00665$$

$$\overline{u} = \overline{h}\sigma'(u) = 1(0.00665) = 0.00665.$$

Only 0.665% of the arriving gradient magnitude crosses this activation.

ReLU keeps one side linear

$$\mathrm{ReLU}(u) = \max(0, u), \quad \mathrm{ReLU}'(u) = \begin{cases} 0 u < 0 \\ 1 u > 0 \end{cases}.$$

u	$\overline{h}$	h	$\overline{u}$
2.3	0.7	2.3	0.7
-2.3	0.7	0	0

ReLU passes the active-side gradient unchanged and blocks the inactive side.

The anchor is half active in expectation

D DERIVATION

ANCHOR · five width-4 layers · ReLU after every affine map · $q_0 = 1 \cdot g_5 = 1$

For a symmetric zero-mean pre-activation z, positive and negative values are equally likely:

$$\mathbb{E}[\mathrm{ReLU}(z)^2] \approx \frac{1}{2}\mathbb{E}[z^2].$$

The same Bernoulli gate appears in the backward squared gauge:

$$\mathbb{E}\left[\left(\mathrm{ReLU}'(z)\overline{h}\right)^2\right] \approx \frac{1}{2}\mathbb{E}\left[\overline{h}^2\right].$$

The gate contributes $\frac{1}{2}$ to both squared-gauge multipliers.

“Dead ReLU” means inactive on every current example

A unit is dead on the current dataset if

$$u_i = w^T x_i + b < 0 \quad \text{for every example } i.$$

Then its incoming parameter gradients are zero because every $\mathrm{ReLU}'(u_i) = 0$.

SUSPECT

large negative bias, overly large update, or poor input scale

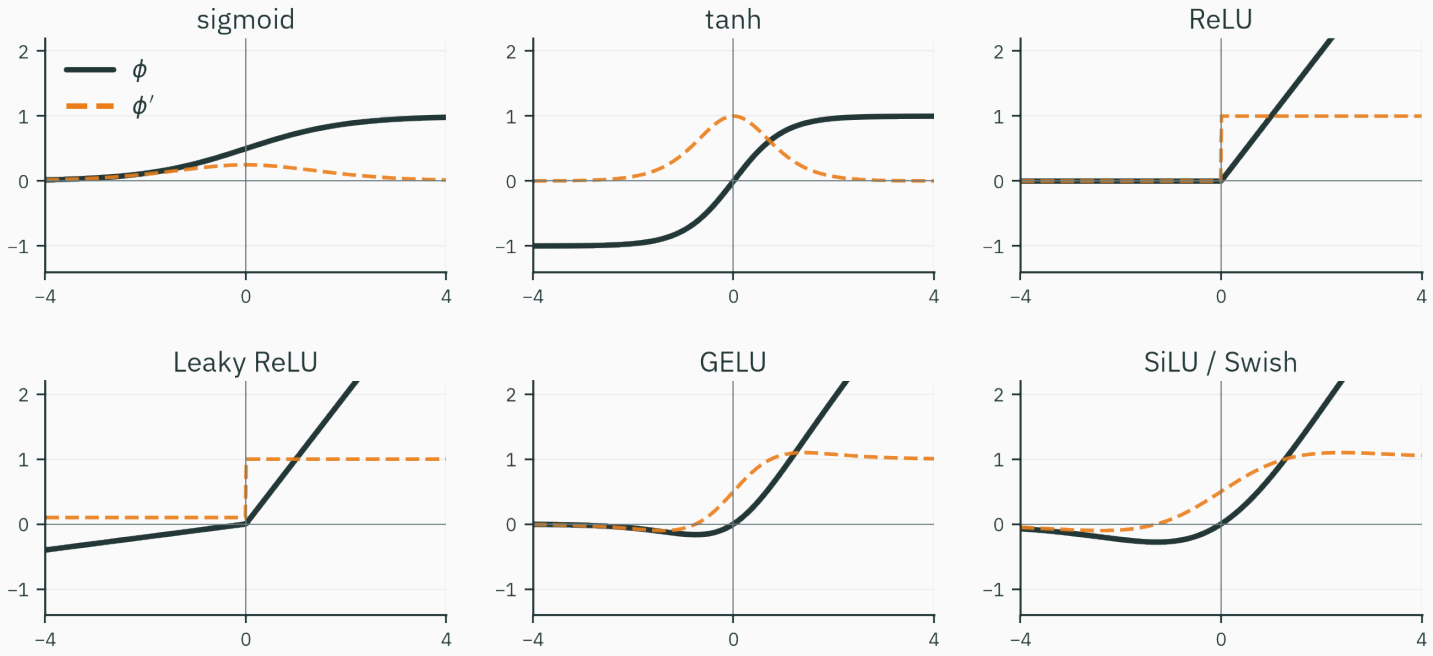
TEST

count examples active for each unit—not only the global zero fraction

> Changing upstream representations may revive it; “can never recover” is too strong.

Compare the gate, not only the curve

V VISUAL



Solid: $\varphi(u)$. Dashed: $\varphi'(u)$, the returning-gradient gate. GELU uses the documented sigmoid approximation $u\sigma(1.702u)$; the other curves are exact away from their kink.

Compare the gate, not only the curve

V VISUAL

Choose the activation for representation; choose initialization to match its gain.

Ledger 1 — the gate is now known

V VISUAL

GAUGE LEDGER · SAME FIVE-LAYER ANCHOR

control	evidence / status · squared q, g; RMS r, s where shown
activation	$c_{\text{gate}} = \frac{1}{2}$ in both squared gauges
initialization	not chosen; weight gain still unknown
normalization	not yet added
residual path	not yet added

ANCHOR · five width-4 layers · ReLU after every affine map · $q_0 = 1 \cdot g_5 = 1$

Initialization must match the gate

Fan-in gives the reproducible scale derivation

D DERIVATION

For one pre-activation $z = \sum_{i=1}^{n} w_i x_i$, assume independent, zero-mean weights and inputs.

$$\mathbb{E}[z^2] = \mathbb{E}\left[\left(\sum_i w_i x_i\right)^2\right]$$

$$= \sum_i \mathbb{E}[w_i^2]\mathbb{E}[x_i^2] + \underbrace{\sum_{i \neq j} \mathbb{E}[w_i]\mathbb{E}[w_j]\mathbb{E}[x_i x_j]}_{0}$$

$$= n_{\text{in}} \operatorname{Var}(w)\, \mathbb{E}[x^2].$$

An affine layer multiplies the squared signal gauge by $n_{\text{in}} \operatorname{Var}(w)$.

The gate tells us which weight variance to choose

ANCHOR · five width-4 layers · ReLU after every affine map · $q_0 = 1 \cdot g_5 = 1$

For width 4 followed by ReLU,

$$c = \underbrace{4\ \mathrm{Var}(w)}_{\text{affine gain}} \times \underbrace{\frac{1}{2}}_{\text{ReLU gate}} = 2\ \mathrm{Var}(w).$$

scheme	$\mathrm{Var}(w)$	one-layer c	five-layer c^5
Xavier	$\frac{1}{4}$	0.5	0.03125
He	$\frac{2}{4} = \frac{1}{2}$	1	1

The same half-active gate that lost energy supplies the missing factor of 2.

Answer: He keeps the anchor gauges near one

A ANSWER

Correct: C — He normal with $\mathrm{Var}(w) = \frac{1}{2}$

Why: For width 4, affine gain $4\left(\frac{1}{2}\right) = 2$ and the ReLU gate contributes $\frac{1}{2}$, so the net squared-gauge factor is $c = 1$.

Xavier normal and Xavier uniform share a variance, so both give $c = 0.5$ here. Their samples are variance-matched alternatives, not identical distributions.

Xavier loses half the squared gauge per ReLU layer

D DERIVATION

ANCHOR · five width-4 layers · ReLU after every affine map · $q_0 = 1 \cdot g_5 = 1$

With $\mathrm{Var}(w) = \frac{1}{4}$,

$$q_5 = (0.5)^5 q_0 = 0.03125, \quad g_0 = (0.5)^5 g_5 = 0.03125.$$

Read the same ledger in the primary RMS gauge:

$$r_5 = \sqrt{q_5} = 0.17678, \quad s_0 = \sqrt{g_0} = 0.17678.$$

Xavier preserves the affine pre-activation scale, but ReLU then removes half the second moment.

He preserves the second moment through ReLU

D DERIVATION

ANCHOR · five width-4 layers · ReLU after every affine map · $q_0 = 1 \cdot g_5 = 1$

With $\mathrm{Var}(w) = \frac{2}{n_{\text{in}}} = \frac{1}{2}$,

$$c = 4\left(\frac{1}{2}\right)\left(\frac{1}{2}\right) = 1.$$

$$q_5 = 1^5 q_0 = 1, \quad g_0 = 1^5 g_5 = 1, \quad r_5 = s_0 = 1.$$

Under the mean-field assumptions, He preserves activation and gradient RMS through the ReLU stack.

The computed trace stays on one honest scale

V VISUAL

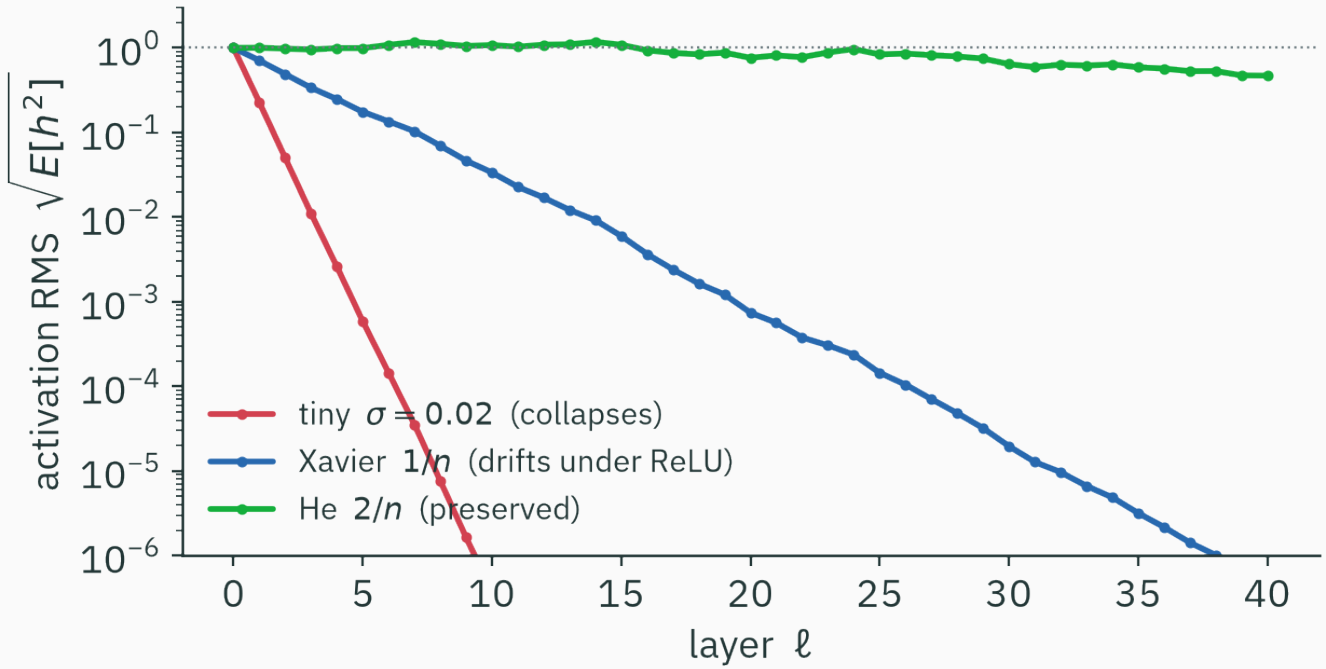
ANCHOR · five width-4 layers · ReLU after every affine map · $q_0 = 1 \cdot g_5 = 1$

squared gauge	$\ell = 0$	1	2	3	5
Xavier 0.5^{ℓ}	1	0.5	0.25	0.125	0.03125
He 1^{ℓ}	1	1	1	1	1

Both rows share the same exact 0–1 scale; no trajectory falls below or leaves the shown range.

A seeded wider simulation shows the same scale story

V VISUAL



Constructed diagnostic, seed 0: width 256, batch 1024, 40 bias-free ReLU layers, Gaussian inputs. Each curve uses newly sampled weights; points are measured activation RMS.

A seeded wider simulation shows the same scale story

V VISUAL

> He keeps the trace order-one in this finite-width draw; Xavier loses roughly one half of the squared gauge per ReLU layer, as predicted.

Xavier samplers match variance, not samples

For $n_{\text{in}} = n_{\text{out}} = 4$, Xavier targets $\text{Var}(w) = \frac{2}{4+4} = \frac{1}{4}$.

NORMAL ALTERNATIVE

$w \sim \mathcal{N}\left(0, \frac{1}{4}\right)$ · unbounded · most mass near 0

UNIFORM ALTERNATIVE

$w \sim \mathcal{U}\left(-\sqrt{\frac{3}{4}}, \sqrt{\frac{3}{4}}\right)$ · bounded · flat density

They have the same mean and variance, so the second-moment calculation matches; higher moments and individual draws do not.

“Variance-matched” is the useful equivalence; the distributions themselves are different.

Convolutions use the fan-in of one output

For a convolution,

$n_{\text{in}} = k_h k_w C_{\text{in}}.$

A 3×3 kernel with 64 input channels has

$$n_{\text{in}} = 3 \cdot 3 \cdot 64 = 576, \quad \text{Std}_{\text{He}} = \sqrt{\frac{2}{576}} \approx 0.0589.$$

> Stride changes output resolution, not how many input values feed one output. A projection shortcut may use a 1×1 convolution with stride > 1 when it must downsample.

Initialization controls only the start

The derivation assumes independent coordinates, symmetric pre-activations, and unchanged weight statistics.

Training breaks these assumptions: correlations grow, weights move, activation means shift, and data are not idealized.

PROMISE	NOT A PROMISE
a sensible RMS scale at $t = 0$	exact preservation for every layer, example, or training step

Initialization earns a healthy start; measurement tells us whether it stays healthy.

Ledger 2 — the anchor starts balanced

V VISUAL

GAUGE LEDGER · SAME FIVE-LAYER ANCHOR

control	evidence / status · squared q, g; RMS r, s where shown
activation	ReLU gate contributes $\frac{1}{2}$
initialization	Xavier: $q_5 = g_0 = .03125$; He: $q_5 = g_0 = 1$
normalization	not yet added
residual path	not yet added

Primary RMS reading: Xavier $r_5 = s_0 = .17678$; He $r_5 = s_0 = 1.$

Normalization resets a reference scale

A normalization layer defines its own measuring group

For values in one chosen group,

$$\mu = \frac{1}{N}\sum_i x_i, \quad \sigma^2 = \frac{1}{N}\sum_i (x_i - \mu)^2$$

$$\hat{x}_i = \frac{x_i - \mu}{\sqrt{\sigma^2 + \varepsilon}}, \quad y_i = \gamma \hat{x}_i + \beta.$$

RESET

$\hat{x}$ has mean near 0 and variance near 1 over that group

LEARN

γ, β restore any useful affine scale and shift

The key design question is not “normalize?” but “which entries share one statistic?”

BatchNorm and LayerNorm choose different axes

V VISUAL

For $X \in \mathbb{R}^{B \times D}$, rows are examples and columns are features.

BatchNorm: down one feature

reduce over the batch axis B

LayerNorm: across one example

reduce over that example's D features

Commit: normalize the slice before calculating

Q QUESTION

For one BatchNorm feature across four examples,

$$x = [1, 2, 3, 4], \quad \gamma = 2, \quad \beta = 1, \quad \varepsilon \approx 0.$$

What are the batch mean and population variance used on this slide?

A. $\mu = 2.5, \sigma^2 = 1.25$

B. $\mu = 2.5, \sigma^2 = \frac{5}{3}$

C. $\mu = 10, \sigma^2 = 5$

D. $\mu = 0, \sigma^2 = 1$

Answer: build the whole BatchNorm slice

A ANSWER

$$\mu = \frac{1+2+3+4}{4} = 2.5.$$

$$\sigma^2 = \frac{(1-2.5)^2 + (2-2.5)^2 + (3-2.5)^2 + (4-2.5)^2}{4} = 1.25.$$

$$\hat{x} = \frac{x - 2.5}{\sqrt{1.25}} \approx [-1.342, -0.447, 0.447, 1.342].$$

$$y = 2\hat{x} + 1 \approx [-1.683, 0.106, 1.894, 3.683].$$

Correct: A — the normalized slice has mean 0, variance 1, and RMS 1 before γ, β.

Learnable scale is allowed to move the gauge

For the normalized slice,

$$\frac{1}{4}\sum_i \hat{x}_i^2 \approx 1 \quad \Rightarrow \quad \mathrm{RMS}(\hat{x}) \approx 1.$$

After $y = 2\hat{x} + 1$,

$$\frac{1}{4}\sum_i y_i^2 \approx 5 \quad \Rightarrow \quad \mathrm{RMS}(y) \approx \sqrt{5}.$$

> Normalization supplies a stable coordinate system; it does not force the learned output RMS to remain 1.

BatchNorm inference must ignore the current batch

During training, BatchNorm uses the current batch and updates running estimates.

During inference, it uses frozen running mean and variance:

$$y = \gamma \frac{x - \mu_{\text{run}}}{\sqrt{\sigma^2_{\text{run}} + \varepsilon}} + \beta.$$

ALLOWED

inference may process one example or a batch for efficiency

FORBIDDEN

an example's prediction changing because current batch-mates changed

`model.eval()` selects running statistics; "inference is always unbatched" is not the rule.

LayerNorm avoids batch dependence

LayerNorm computes one mean and variance from each example's feature vector:

$$\mu_i = \frac{1}{D}\sum_j x_{ij}, \quad \sigma_i^2 = \frac{1}{D}\sum_j \left(x_{ij} - \mu_i\right)^2.$$

property	BatchNorm	LayerNorm
normalizes over	examples per feature	features per example
depends on batch-mates?	training: yes	no
train / inference stats	batch / running	same computation
common home	CNN feature channels	sequence/token models

Choose the axis from the invariance you need, not from the layer's name.

Ledger 3 — the BN slice supplies a scale reset

V VISUAL

GAUGE LEDGER · SAME FIVE-LAYER ANCHOR

control	evidence / status · squared q, g; RMS r, s where shown
activation	ReLU gate contributes $\frac{1}{2}$
initialization	He gives $q_5 = g_0 = 1$ at $t = 0$
normalization	$[1, 2, 3, 4]$: $\mu = 2.5, \sigma^2 = 1.25, \mathrm{RMS}(\hat{x}) = 1$
residual path	not yet added

The reset is defined over the selected axis; learned γ, β may deliberately move the post-normalization RMS.

Residual connections shorten the route

Greater depth can make the training error worse

V VISUAL

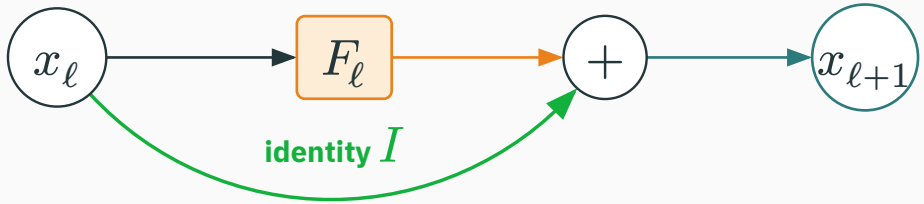
In the original ResNet experiment on CIFAR-10, a 56-layer **plain** network had higher training error than a 20-layer plain network.

He et al., “Deep Residual Learning for Image Recognition,” CVPR 2016, Fig. 1.

Learning a correction makes “do nothing” easy

Reparameterize a desired map H_ℓ as

$$F_{\ell(x)} := H_{\ell(x)} - x \quad \Rightarrow \quad H_{\ell(x)} = x + F_{\ell(x)}.$$



If the useful transformation is close to identity, the branch only has to learn the small correction F_ℓ.

Column-gradient convention fixes the transpose

D DERIVATION

Let $J_{F_\ell} = \partial \frac{F_{\ell(x_\ell)}}{\partial} x_\ell$ and store gradients as column vectors.

$$x_{\ell+1} = x_\ell + F_{\ell(x_\ell)}$$

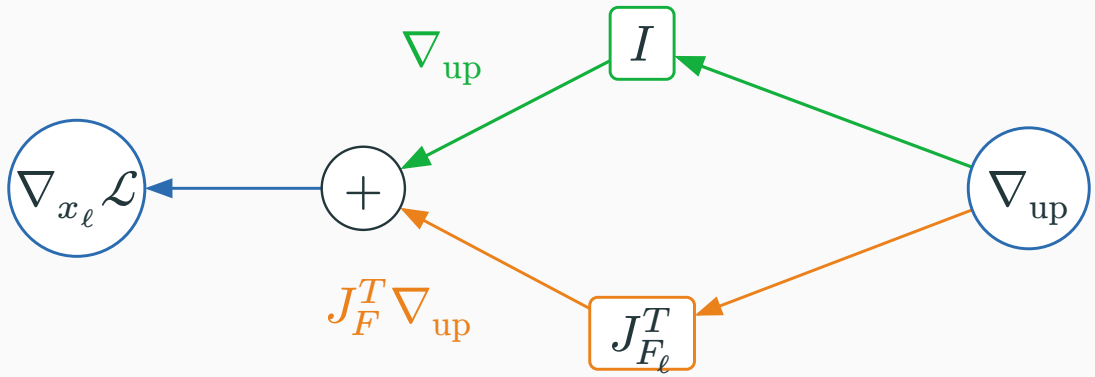
$$\nabla_{x_\ell} \mathcal{L} = \left(I + J_{F_\ell}\right)^T \nabla_{x_{\ell+1}} \mathcal{L}$$

$$= \underbrace{\nabla_{x_{\ell+1}} \mathcal{L}}_{\text{identity contribution}} + \underbrace{J_{F_\ell}^T \nabla_{x_{\ell+1}} \mathcal{L}}_{\text{residual contribution}} .$$

Backward applies the transposed local Jacobian; the two contributions add at the block input.

The arriving gradient splits into two contributions

V VISUAL



IDENTITY

∇_{up} arrives without a branch matrix multiply

RESIDUAL

$J_F^T\nabla_{\text{up}}$ may reinforce or cancel it

A shortcut adds a route; it does not guarantee that every total gradient is large.

Commit: compare five plain and residual blocks

Q QUESTION

Let one scalar branch have $F'(x) = -0.1$ and let the top gradient magnitude be 1.

After five blocks, which pair of input-gradient magnitudes is correct?

A. plain 0.5; residual 1.5

B. plain 10^{-5}; residual 0.9^5

C. plain 0.1; residual 0.9

D. plain 0; residual 1 exactly

Answer: the shortcut changes the local multiplier

A ANSWER

path	one block	five-block magnitude	squared gauge
plain branch F	$\lvert -0.1 \rvert$	$\lvert -0.1 \rvert^5 = 0.00001$	10^{-10}
residual $I + F$	$\lvert 1 - 0.1 \rvert = 0.9$	$0.9^5 = 0.59049$	$0.59049^2 = 0.34868$

Correct: B — the residual RMS magnitude is 0.59049; its squared ledger is 0.3487.

Residual structure improves the route, not every ingredient

failure	why shortcut alone is insufficient	check
branch explodes	J_F can dominate the identity	branch/output RMS
branch cancels identity	$J_F \approx -I$ in a direction	identity vs residual contribution
shape changes	x and $F(x)$ cannot be added	projection shape/stride
training diverges	learning rate still controls update size	update-to-weight RMS

> Initialization, normalization, residual scaling, and learning rate remain coupled design choices.

Ledger 4 — the short path completes the anchor

V VISUAL

GAUGE LEDGER · SAME FIVE-LAYER ANCHOR

control	evidence / status · squared q, g; RMS r, s where shown
activation	ReLU gate contributes $\frac{1}{2}$
initialization	He gives $q_5 = g_0 = 1$ at $t = 0$
normalization	$\mu = 2.5, \sigma^2 = 1.25$, normalized RMS $= 1$
residual path	$F' = -.1$: plain RMS 10^{-5}; residual RMS $.59049$ $(g = .3487)$

Each intervention answers a different reason the gauges can drift.

Measure the system, not the slogan

One ledger connects all four controls

V VISUAL

control	mechanism	anchor evidence	what it does not guarantee
activation	local gate	ReLU contributes $\frac{1}{2}$	no dead units
initialization	starting gain	He changes $c : .5 \to 1$	scale later in training
normalization	axis-wise reset	$[1, 2, 3, 4] \to \mathrm{RMS} = 1$	post-γ, β RMS
residual	short additive route	$10^{-5} \to .59049$	no cancellation/ explosion

Use the ledger to explain a mechanism, then measure the actual network.

Diagnose with symptom → suspect → test

symptom	first suspect	smallest informative test
activation RMS shrinks by depth	gate/init mismatch	log pre/post RMS by layer
early gradient RMS is tiny	saturation or long path	log gradient RMS and shortcut terms
many inactive ReLU units	bias/update/input scale	active-example count per unit
train/eval predictions differ	BatchNorm statistics	same input in both modes
first non-finite value	scale or update explosion	locate first bad layer and step

Change one cause-relevant control at a time so the experiment can identify the cause.

Measure the same RMS gauges in code

I INTERACTIVE

```
def rms(x):
    return x.detach().float().square().mean().sqrt().item()

for name, h in activations.items():
    print(name, "activation_rms", rms(h))

for name, p in model.named_parameters():
    if p.grad is not None:
        print(name, "gradient_rms", rms(p.grad),
              "finite", bool(p.grad.isfinite().all()))
```

`square().mean().sqrt()` matches r_ℓ, s_ℓ; a raw `norm()` would grow with width and break the ledger.

Three labs, one predict → run → explain loop

I INTERACTIVE

1 · autopsy	01_signal_gradient_autopsy.ipynb
2 · propagate	02_variance_propagation.ipynb
3 · compare	03_xavier_vs_he.ipynb

For each notebook: **commit to a trace → run → compare with the ledger → explain any mismatch**.

> The three filenames above are direct Colab links to the public repository.

Misconception check: which claim survives?

Q QUESTION

Which statement is defensible without extra assumptions?

A. He exactly preserves every layer's variance throughout training

B. BatchNorm inference must contain only one example

C. Residual shortcuts guarantee gradients cannot vanish

D. Each control targets a mechanism; the actual gauges must still be measured

Answer: mechanisms guide; measurements decide

A ANSWER

Correct: D — Each control targets a mechanism; the actual gauges must still be measured

Why: He is an initialization approximation, inference may be batched, and a residual branch can reinforce or cancel the identity contribution.

Revisit your opening bet

ANCHOR · five width-4 layers · ReLU after every affine map · $q_0 = 1 \cdot g_5 = 1$

choice	five-layer squared gauges	five-layer RMS gauges
Xavier normal / uniform	$q_5 = g_0 = .03125$	$r_5 = s_0 = .17678$
He normal	$q_5 = g_0 = 1$	$r_5 = s_0 = 1$
all-zero weights	symmetry is unbroken	units learn the same feature

The winning bet was He because ReLU's half-active gate required twice Xavier's variance.

Exit ticket: prescribe the smallest repair

For a 30-layer ReLU MLP, you observe activation RMS $1.0 \to 0.6 \to 0.35 \to \cdots$ and early-layer gradient RMS near zero.

Write three lines:

1. **symptom:** which gauge drifts, and in which direction?
2. **suspect:** which local multiplier could explain it?
3. **test:** which single controlled change or measurement distinguishes the cause?

> A strong answer names the activation/init pair first, verifies per-layer RMS, then considers normalization or a shorter residual route.

Depth is trainable when signal and gradient RMS remain usable along the routes that matter.

Next: L7 asks whether a trainable network also **generalizes**—regularization and the train/test gap.

Further variants and provenance

RMSNorm resets magnitude without mean subtraction

For one example with D features,

$$\mathrm{RMS}(x) = \sqrt{\frac{1}{D}\sum_{j=1}^{D} x_j^2 + \varepsilon}, \quad y_j = \gamma_j \frac{x_j}{\mathrm{RMS}(x)}.$$

KEEPS	DROPS
feature-wise learned scale and RMS denominator	mean subtraction and learned shift in the basic form

> $\varepsilon > 0$ prevents division by zero and limits amplification when the input RMS is extremely small.

Projection shortcuts can change width and resolution

If $F_{\ell(x_\ell)}$ and x_ℓ have different shapes, use

$$x_{\ell+1} = P_\ell x_\ell + F_{\ell(x_\ell)}.$$

In CNNs, P_ℓ is often a learned 1×1 convolution:

- stride 1 can change channel count;
- stride > 1 can change channels **and** downsample spatial resolution.

The shortcut must match shape before addition; it need not be a literal identity tensor.

Placement decides whether the shortcut is truly clean

POST-ACTIVATION

$x_{\ell+1} = \varphi\big(x_\ell + F_{\ell(x_\ell)}\big)$; the final φ' gates both paths

PRE-ACTIVATION

norm and activation live in F_ℓ; the additive route stays unobstructed

> Fixup is a separate carefully scaled initialization strategy that trains deep residual networks without normalization; it is evidence that these controls are coupled, not mandatory in one fixed recipe.

BatchNorm for convolution reduces over B, H, W

For $X \in \mathbb{R}^{B \times C \times H \times W}$, BatchNorm computes one mean and variance per channel:

$$\mu_c, \sigma_c^2 \quad \text{over } B \cdot H \cdot W \text{ values of channel } c.$$

axis	reduced?	meaning
batch B	yes	examples share channel statistics
channel C	no	one statistic pair per channel
space H, W	yes	locations act as additional samples

At inference the frozen per-channel running statistics are used, regardless of inference batch size.

Generalized He gain handles a leaky slope

For $\varphi(u) = \max(u, au)$ with a symmetric pre-activation,

$$\mathbb{E}[\varphi(z)^2] \approx \frac{1 + a^2}{2}\mathbb{E}[z^2].$$

Matching the affine gain gives

$$\mathrm{Var}(w) = \frac{2}{(1 + a^2)n_{\text{in}}}.$$

negative slope a	gain factor	$\mathrm{Var}(w)$
0	$\frac{1}{2}$	$\frac{2}{n_{\text{in}}}$
0.1	0.505	$\frac{2}{1.01n_{\text{in}}}$
1	1	$\frac{1}{n_{\text{in}}}$

Primary references and provenance

- Glorot & Bengio (2010), Understanding the difficulty of training deep feedforward neural networks.
- He et al. (2015), Delving Deep into Rectifiers.
- Ioffe & Szegedy (2015), Batch Normalization.
- Ba, Kiros & Hinton (2016), Layer Normalization.
- He et al. (2016), Deep Residual Learning for Image Recognition.
- Zhang & Sennrich (2019), Root Mean Square Layer Normalization.

The activation catalogue and seeded signal-flow trace are generated by `lecture6/diagrams/l6_figs.py`; other diagrams and numeric traces are native Typst vectors computed from the equations shown.