An easy-to-follow guide for AI/ML enthusiasts

How Do You Initialize Weights in a Deep Neural Network?

Sanjay N Kumar

Data scientist | AI ML Engineer | Statistician | Analytics Consultant

Why We Need to Initialize Weights 🤔

Imagine you are baking 🍰

If you add *too much sugar*, it’s too sweet.

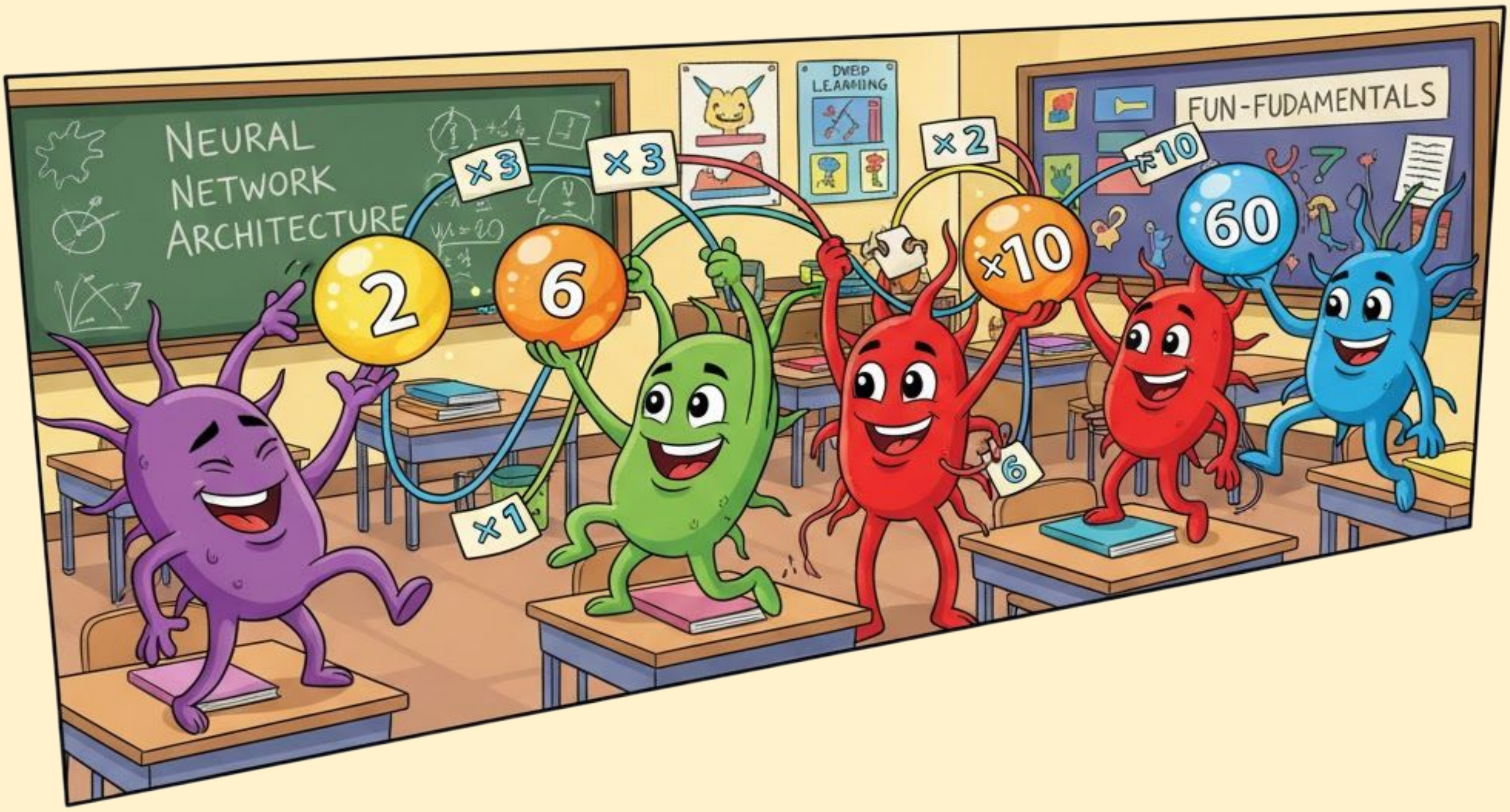
If you add *too little*, it’s tasteless.

➡️ Neural networks are similar!

Weights decide how **“sweet” (strong) or “weak”** the signals between neurons are.

We must start with the ***right amount*** — not too big, not too small.

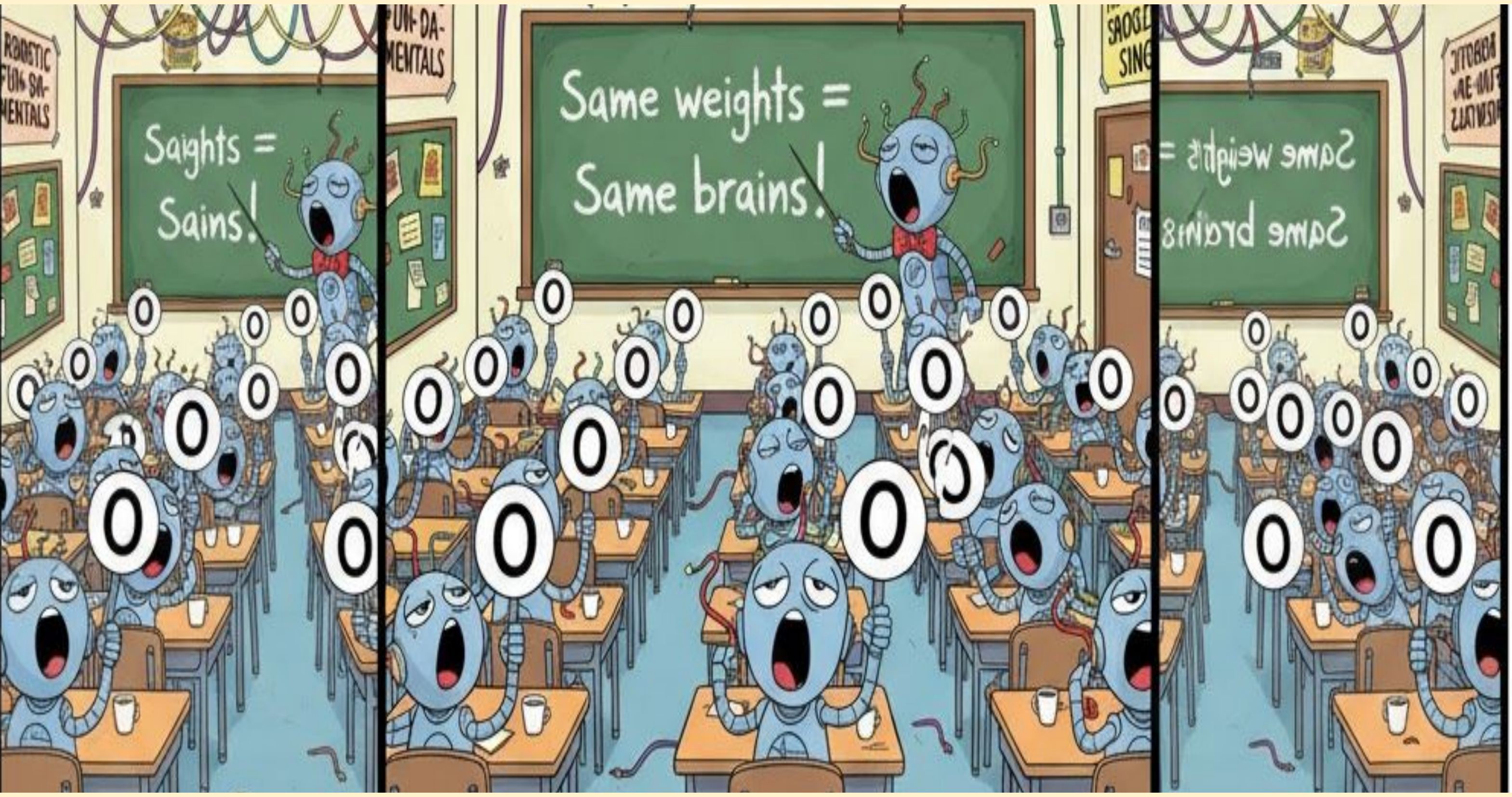
What Are Weights? ⚙️



Each connection between neurons has a number — called a **weight**.

It tells how strong the connection is.
If input = 2 and weight = 3, output = 2×3 = 6.
Simple math! ✨

Why Random Weights Are Needed 🧩

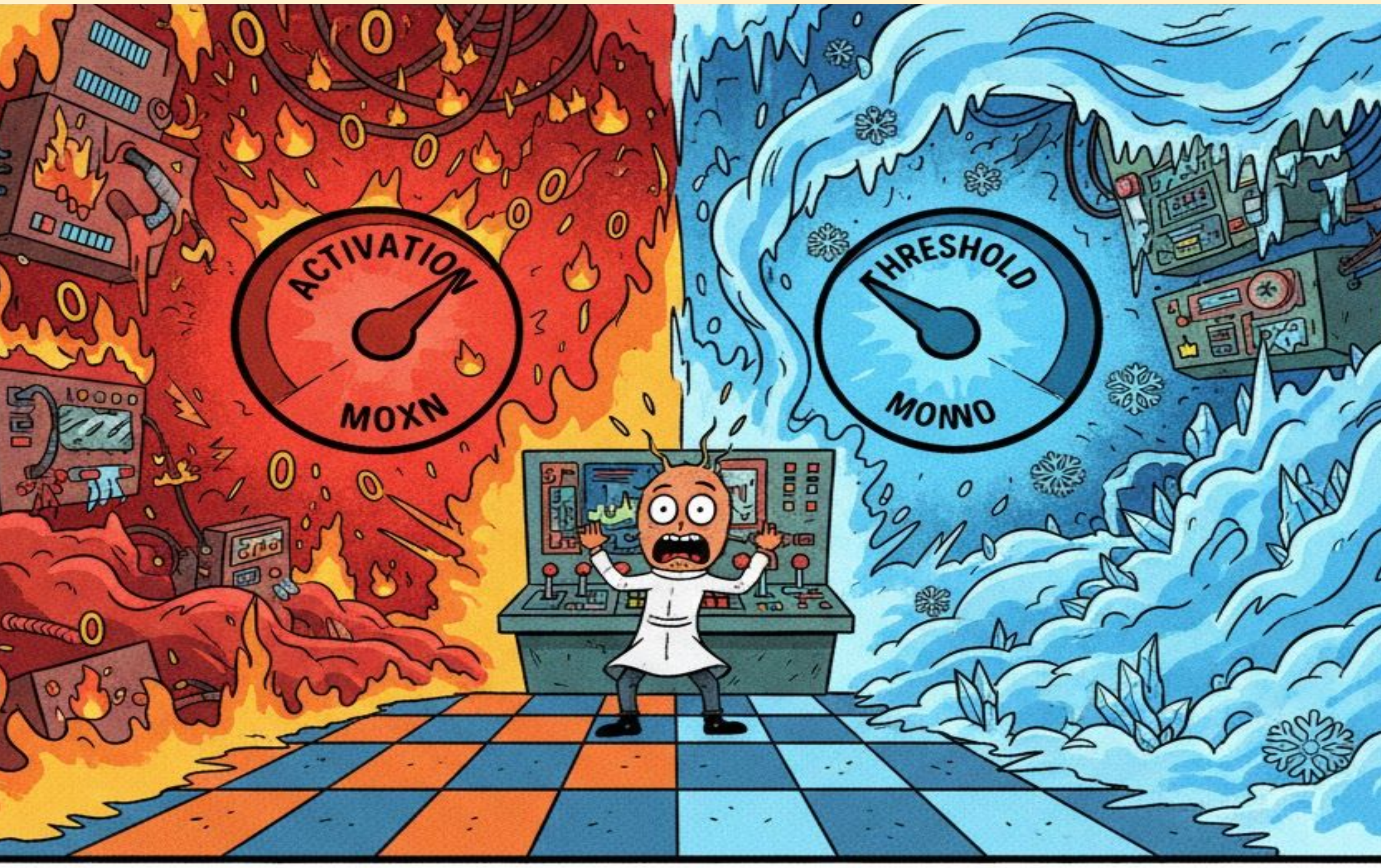


If all weights start as **zero**, all neurons learn the same thing! 😴

They become *clones*.

So we start with ***random small numbers*** — so every neuron learns differently.

The Problem With Randomness 🌄



But random values can go wrong too! 😱

If weights are **too large**, output blows up 🔥
If weights are **too small**, everything becomes almost zero 🧊

This makes learning very slow or unstable.

So, We Need Smart Initialization! 💡

We need a rule that says —

"Start weights just right, not too big or too small."

That's where **Xavier (Glorot)** and **He Initialization** come in!

Xavier / Glorot Initialization — Intuition ✴️

👉 Made for networks using **Sigmoid** or **Tanh** activation.

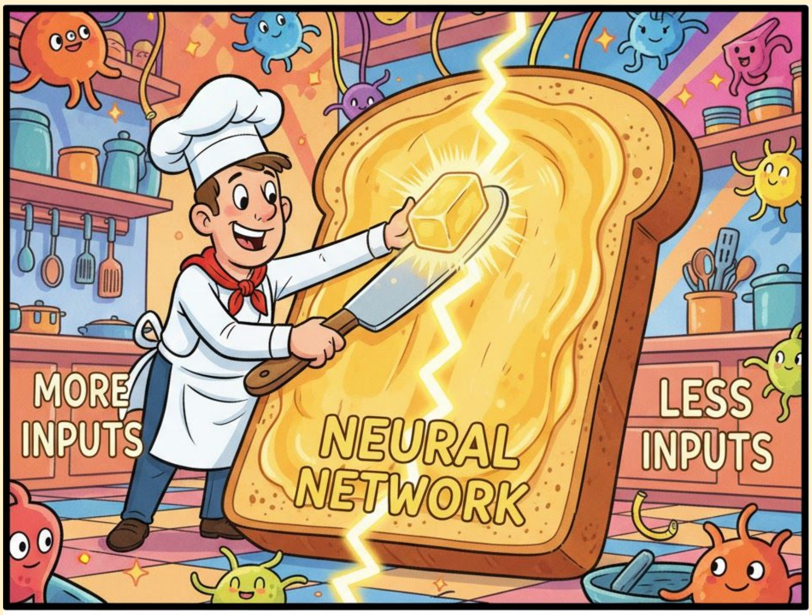
Idea: Keep the signal's strength the same as it flows forward and backward.

Too strong → explosion

Too weak → vanish

🎯 Xavier balances both sides.

Simple Math (Xavier) 🧮



Xavier sets weights based on number of input and output neurons:

$$Var(w) = \frac{2}{(n_{in} + n_{out})}$$

Means:

➡️ If many inputs, make weights smaller.

➡️ If few inputs, make weights slightly bigger.

Just like spreading butter evenly on toast 🍞

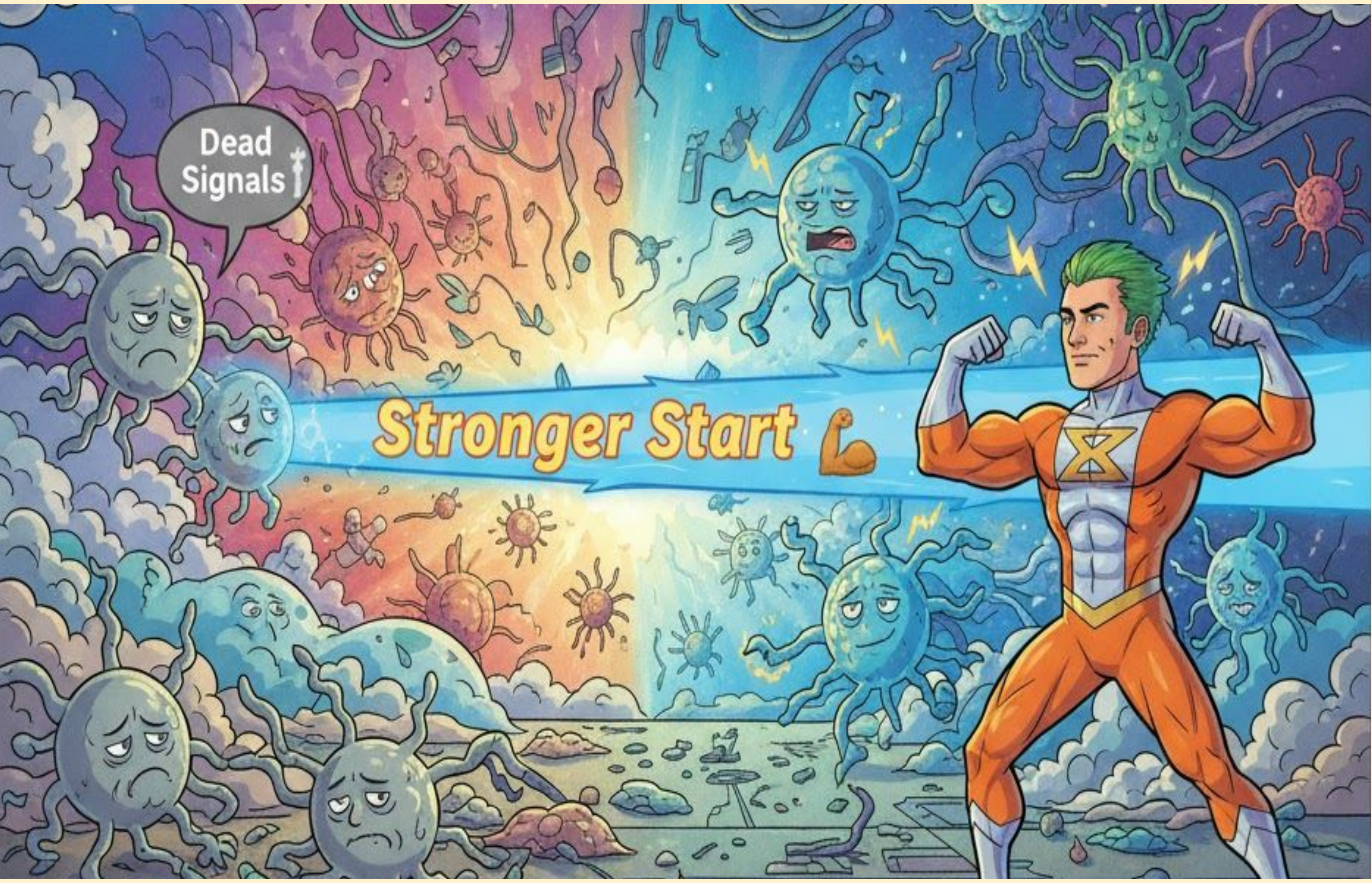
Real-Life Example (Xavier) 🍔

If you pour too much ketchup on a burger 🍔 — it’s messy.

If too little — dry!

Xavier finds ***just the right amount*** so every layer gets a balanced taste of information. 😄

He Initialization — Intuition ⚡



👉 Made for **ReLU** activations.
ReLU keeps only positive values, so half the signals die off. ⚰️

To fix this, He Initialization gives a *stronger start*. 💪

Simple Math (He) 🧮

He Initialization uses:

$$Var(w) = \frac{2}{n_{in}}$$

That “2” makes it slightly stronger than Xavier — to wake up those neurons that might get stuck at zero!

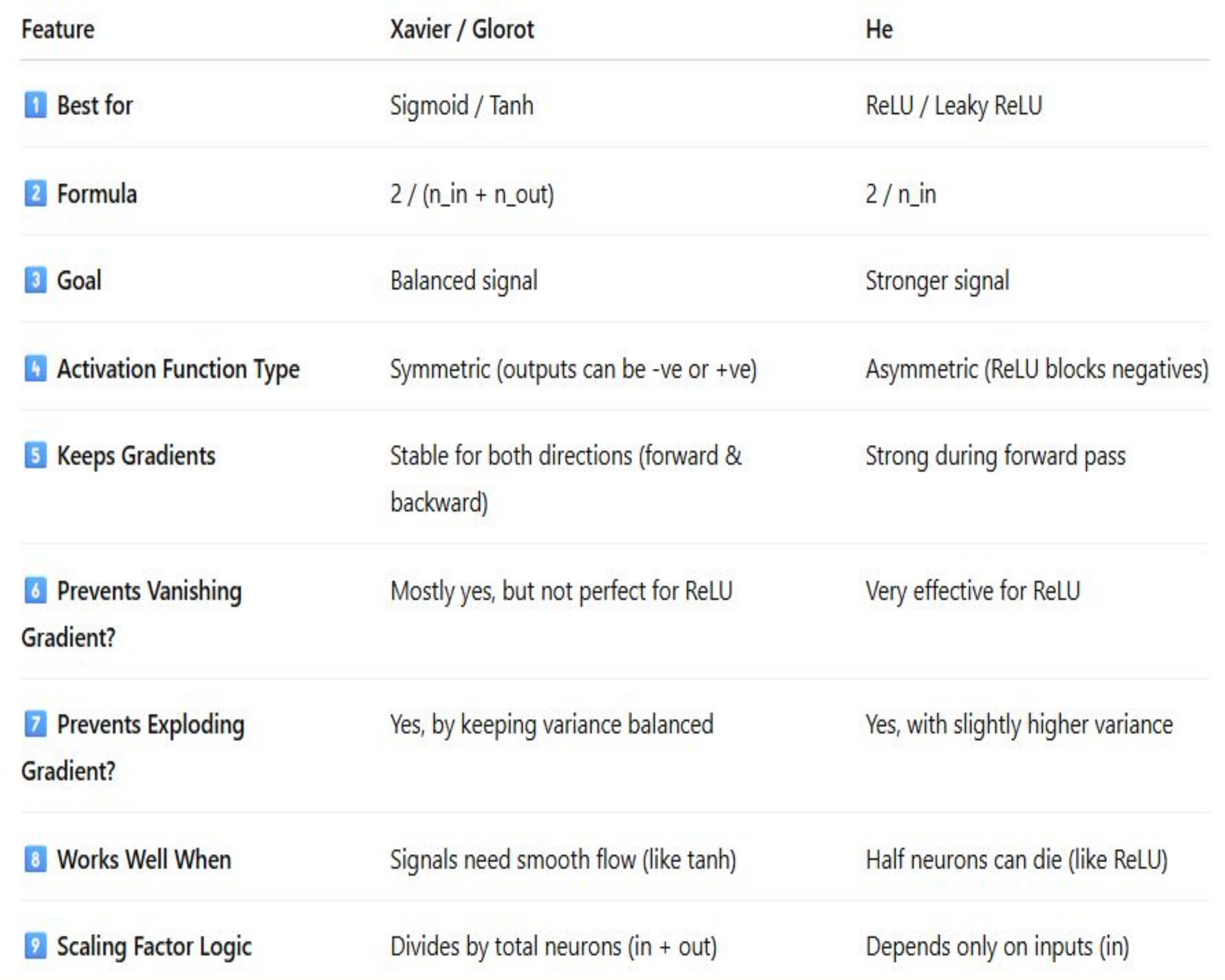
Real-Life Example (He) 🧠

Imagine you’re trying to wake sleepy students 💤

If you speak softly, half stay asleep.
So you speak ***a little louder*** 🔊

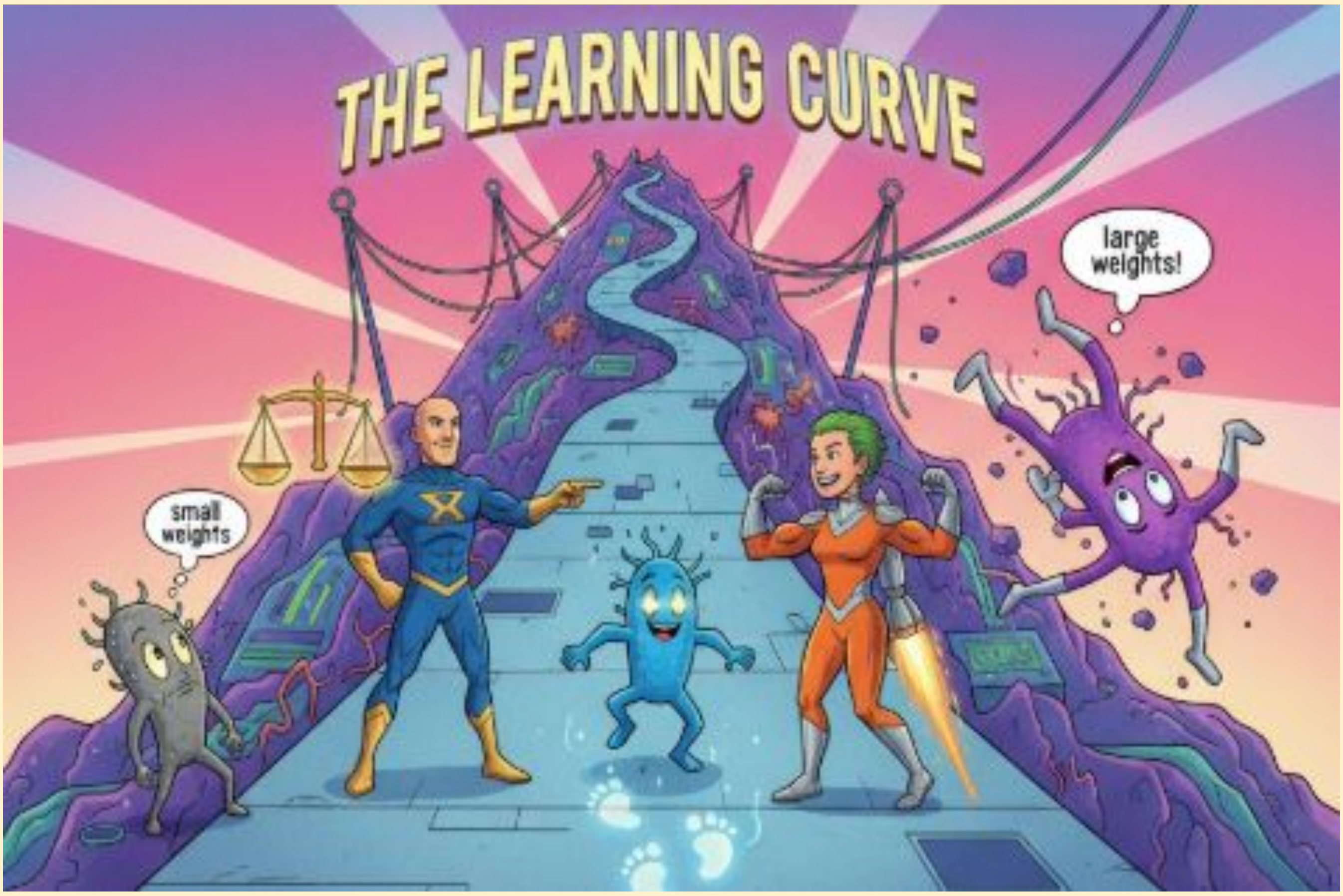
That’s what He Initialization does — gives a louder push to start learning faster!

Comparing Xavier vs He



Feature	Xavier / Glorot	He
1️⃣ Best for	Sigmoid / Tanh	ReLU / Leaky ReLU
2️⃣ Formula	2 / (n_in + n_out)	2 / n_in
3️⃣ Goal	Balanced signal	Stronger signal
4️⃣ Activation Function Type	Symmetric (outputs can be -ve or +ve)	Asymmetric (ReLU blocks negatives)
5️⃣ Keeps Gradients	Stable for both directions (forward & backward)	Strong during forward pass
6️⃣ Prevents Vanishing Gradient?	Mostly yes, but not perfect for ReLU	Very effective for ReLU
7️⃣ Prevents Exploding Gradient?	Yes, by keeping variance balanced	Yes, with slightly higher variance
8️⃣ Works Well When	Signals need smooth flow (like tanh)	Half neurons can die (like ReLU)
9️⃣ Scaling Factor Logic	Divides by total neurons (in + out)	Depends only on inputs (in)

Simple Visual Idea 🧠



Think of climbing a hill 🏔️

- If steps are too small → takes forever (small weights)

- If steps are too big → you fall (large weights)
  Xavier & He help you take *perfect steps* — steady and safe. 👣

Summary 💬

✅ Weights decide how the network learns.

✅ Bad initialization = bad learning.

✅ Xavier keeps things balanced.

✅ He gives more energy for ReLU layers.

✅ Both make training faster and stable! ⚡

The Power of a Good Start! ⚡

Ignite your network with the *right spark*.
Let Xavier and He Initialization guide your neurons —
from randomness to rhythm, from chaos to clarity! ✨

💬 **Train smart. Start right. Learn deep.**

Reach out — let's build intelligence that learns with balance, power, and purpose!

Sanjay N Kumar

Data scientist | AI ML Engineer | Statistician | Analytics Consultant

https://www.linkedin.com/in/sanjaytheanalyst360/

sanjaytheanalyst360@gmail.com