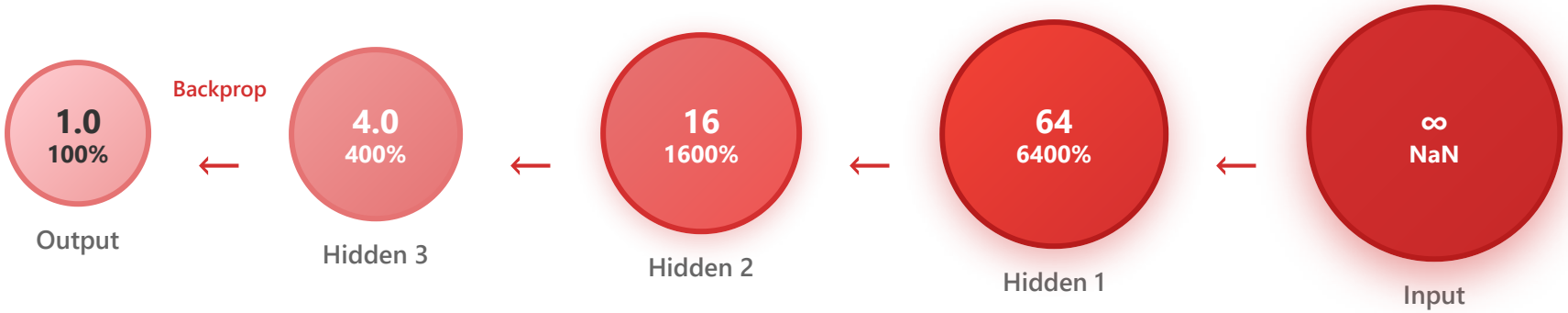
Exploding Gradient Problem



MATHEMATICAL CAUSE

$$\text{gradient} \propto \prod(\partial a_l/\partial z_l) \rightarrow \text{Product of many terms} > 1$$

Consequences

- Unstable parameter updates
- Loss becomes NaN or infinity
- Training loss spikes dramatically
- Weights become very large

Gradient Clipping	Xavier/He Init	Batch Normalization

📈 Activation Functions & Gradient Explosion Risk

⚠ High Risk

Sigmoid (σ)

$\sigma'(z) = \sigma(z)(1-\sigma(z))$

Max gradient: 0.25

⚠ Medium Risk

Tanh

$\tanh'(z) = 1 - \tanh^2(z)$

Max gradient: 1.0

✅ Lower Risk

ReLU

$\text{ReLU}'(z) = 1 \text{ if } z>0 \text{ else } 0$

Gradient: 0 or 1

$0.25^5 = 0.00098$ ❌

Poor weight init → explode

But large W → still explode

🔢 Chain Rule Multiplication: How Gradients Explode

Backpropagation through layers:

$$\partial L/\partial W_1 = \partial L/\partial a_4 \times \mathbf{\partial a_4/\partial z_4} \times \partial z_4/\partial a_3 \times \mathbf{\partial a_3/\partial z_3} \times \partial z_3/\partial a_2 \times \mathbf{\partial a_2/\partial z_2} \times \partial z_2/\partial a_1 \times \mathbf{\partial a_1/\partial z_1} \times \partial z_1/\partial W_1$$

Red terms: activation derivatives (σ', tanh', ReLU')

Example with poorly initialized weights:

❌ Bad Case (W ~ 2.0)

```
Layer 1: grad × W₁  × σ'
= 1.0 × 2.0 × 0.25 = 0.5
Layer 2: 0.5 × 2.0 × 0.25 = 0.25
Layer 3: 0.25 × 2.0 × 0.25 = 0.125
Layer 4: 0.125 × 2.0 × 0.25 = 0.0625
```

If W=3: 3×0.25=0.75 → vanish!
If W=5: 5×0.25=1.25 → explode! 💥

✅ Good Case (Xavier Init)

```
W ~ N(0, 1/√n)
Keeps gradient variance stable

E[∂L/∂W] ≈ constant
No explosion or vanishing
```

💥 Real Explosion Scenario

Initial Weight		5 Layers		Gradient Scale
W = 2.5	×	**n = 5**	→	**$2.5^5 = 97.7$**

With 20 layers: $2.5^{20} = 9.09 \times 10^8$ → NaN! 🔥