🔁 Deep Learning Review Notes — Targeted Gaps

✅ 1. Activation Derivatives

Sigmoid

- Definition:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

- Derivative:

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

- Notes: Derivative is small for large |x| → vanishing gradients.

Tanh

- Definition:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

- Derivative:

$$\frac{d}{dx}\tanh(x) = 1 - \tanh^2(x)$$

- Notes: Zero-centered → often better than sigmoid.

ReLU vs GELU

Feature	ReLU	GELU (used in Transformers)
Formula	$\max(0, x)$	$\mathrm{GELU}(x) \approx 0.5x(1 + \tanh(\sqrt{2/\pi}(x + 0.0447x^3)))$
Derivative	0 (x < 0), 1 (x > 0)	Smooth, always non-zero
Gradient Behavior	Harsh cutoff	Probabilistic, soft cutoff
Problem	Dead neurons	No dead zones

🚀 2. Optimizer Theory

Adam Optimizer (Adaptive Moment Estimation)

1. Gradient:

$$g_t = \nabla_\theta L(\theta_t)$$

2. First moment estimate (mean):

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$

3. Second moment estimate (uncentered variance):

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

4. Bias correction:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}, \quad \hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

5. Parameter update:

$$\theta_t = \theta_{t-1} - \eta \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$$

- Defaults: $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$

AdamW (Decoupled Weight Decay)

- **Old approach (Adam + L2):**

$$g_t \leftarrow g_t + \lambda \cdot \theta$$

- **AdamW decouples it:**

$$\theta \leftarrow \theta - \eta \cdot \text{AdamUpdate} - \eta \cdot \lambda \cdot \theta$$

- **Why it matters**: Regularization is applied directly to weights, not gradients → more consistent behavior.
- **Used in**: All modern transformer training (BERT, GPT, T5, etc.)

📉 3. Learning Rate Scheduling

Why Use a Schedule?

- Large LR: fast but unstable
- Small LR: slow but stable
- Schedules give you the **best of both** (start warm, then cool)

Linear Warmup

- Slowly ramp up the LR over the first T_{warmup} steps:

$$\text{lr}_t = \eta \cdot \frac{t}{T_{\text{warmup}}}$$

Cosine Decay

- After warmup, gradually decay using a cosine function:

$$\text{lr}_t = \eta \cdot 0.5 \left(1 + \cos\left(\frac{t - T_{\text{warmup}}}{T_{\text{total}} - T_{\text{warmup}}} \cdot \pi\right)\right)$$

- Smoothly transitions learning rate to near zero by the end of training.

Visual Summary

```
In [ ]:
```