

The background of the slide is filled with various 3D geometric shapes in a light beige or off-white color. These shapes include spheres, cubes, cylinders, cones, a torus (donut shape), a rectangular frame, and wavy lines. They are scattered across the slide, creating a complex, abstract composition. The lighting is soft, giving the shapes a slight shadow and a three-dimensional appearance.

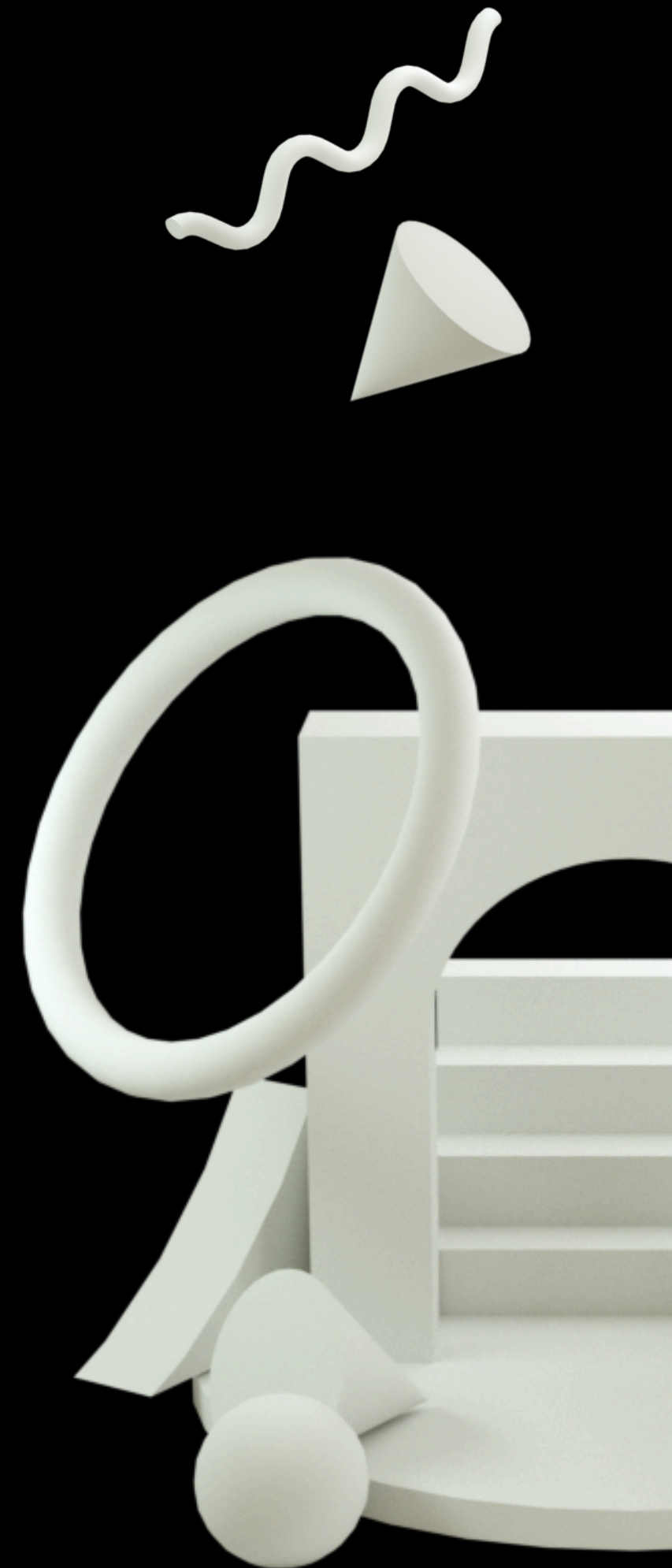
Deep Neural Networks

Activation Functions and Depth Analysis

Tigist W.


Introduction

- This mini project explores the impact of different activation functions and the role of network depth in ANNs.
- We compare Sigmoid, Tanh, and ReLU activations.
- Investigate ReLU variants: Leaky ReLU and Parametric ReLU.
- Analyze network performance for different depths.





Activation Functions Overview

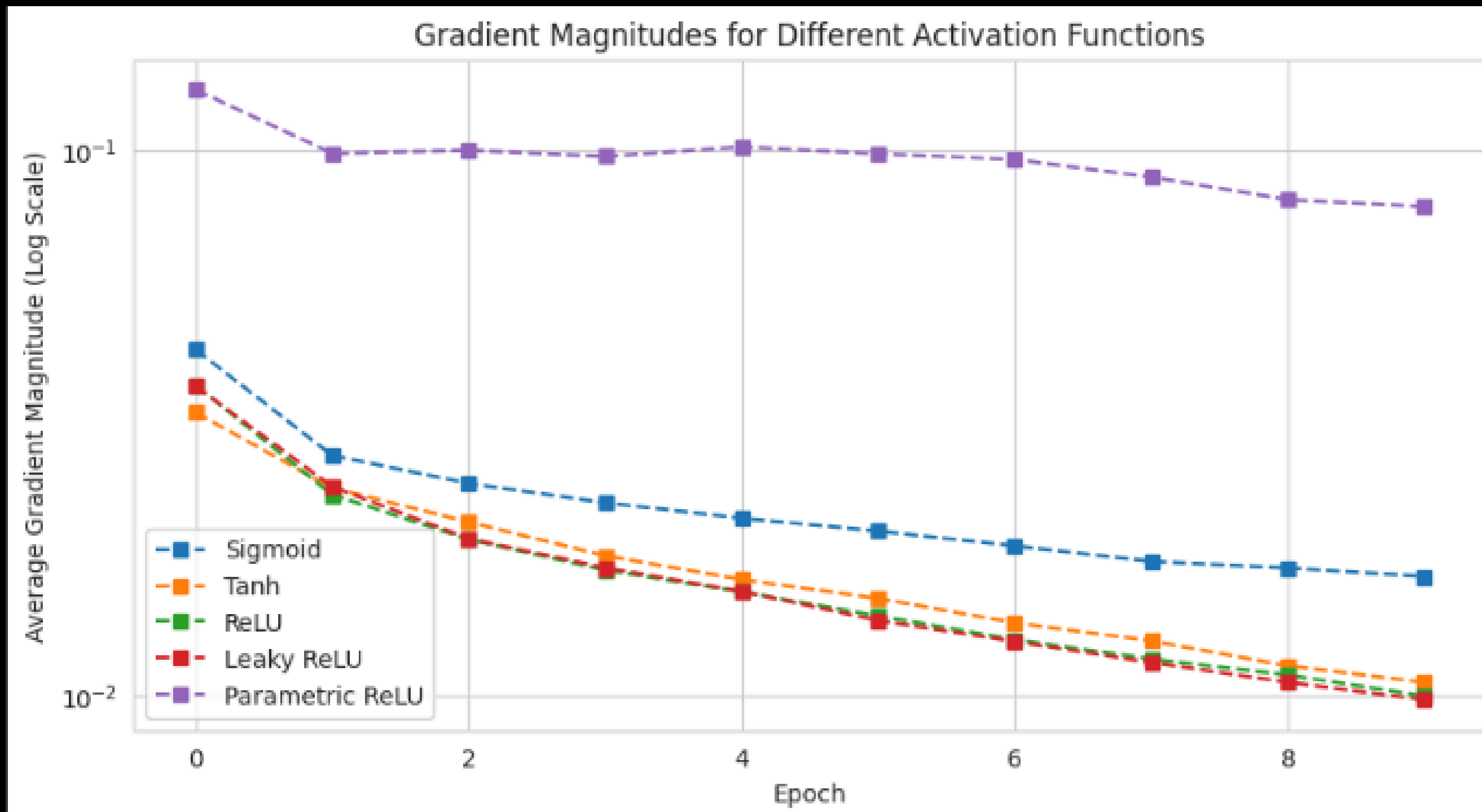
- **Sigmoid:** Output range $(0,1)$; suffers from vanishing gradients.
 - **Tanh:** Output range $(-1,1)$; still suffers from vanishing gradients.
 - **ReLU:** Allows unbounded positive values; mitigates vanishing gradient issues.
 - **Leaky ReLU & Parametric ReLU:** Fix the dying ReLU problem by allowing small negative slopes.
- 

Experiment Setup

- **Dataset: MNIST (Handwritten digits classification)**
- **Model: Fully Connected DNN**
- **Evaluation: Loss, gradient magnitude, and accuracy**
- **Tools: PyTorch, Matplotlib for visualizations**

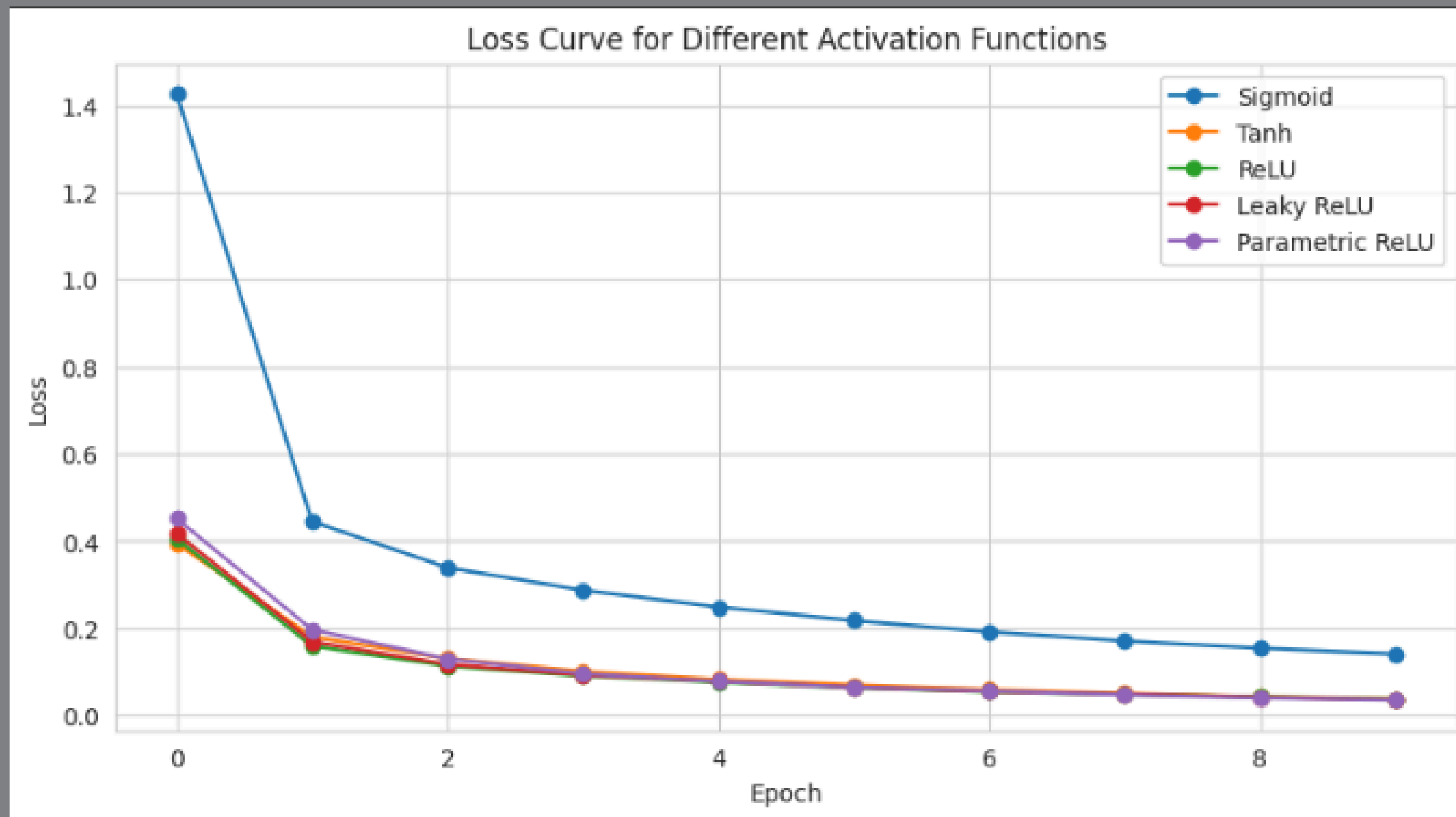
Activation Function Comparison

Gradient Magnitude Analysis



Performance Visualization

Loss curves: Show that ReLU converges faster than Sigmoid & Tanh.



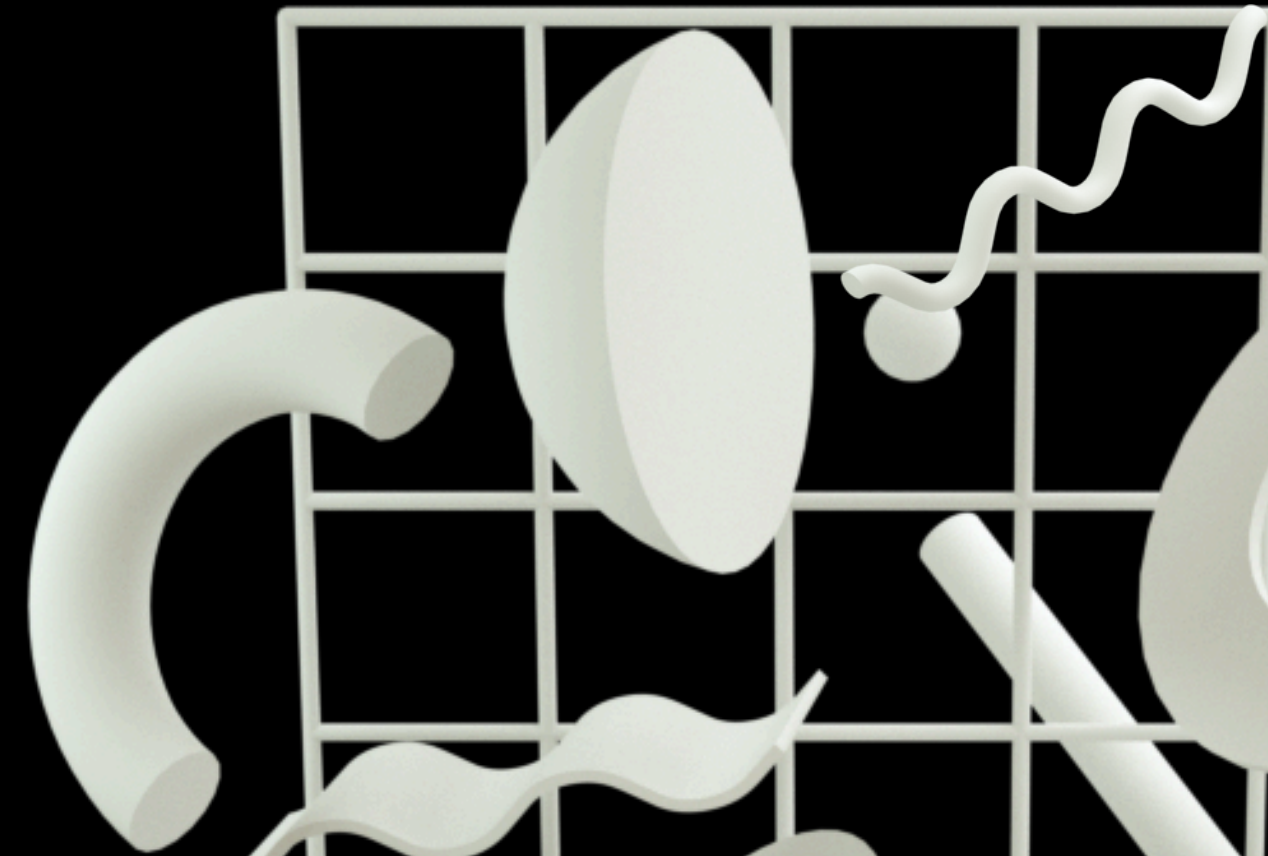
Depth of ANNs

- **Testing different network depths:**
 - **2-layer shallow network**
 - **3-layer moderate network**
 - **5-layer deep network**
 - **7-layer very deep network**
 - **9-layer extra deep network**

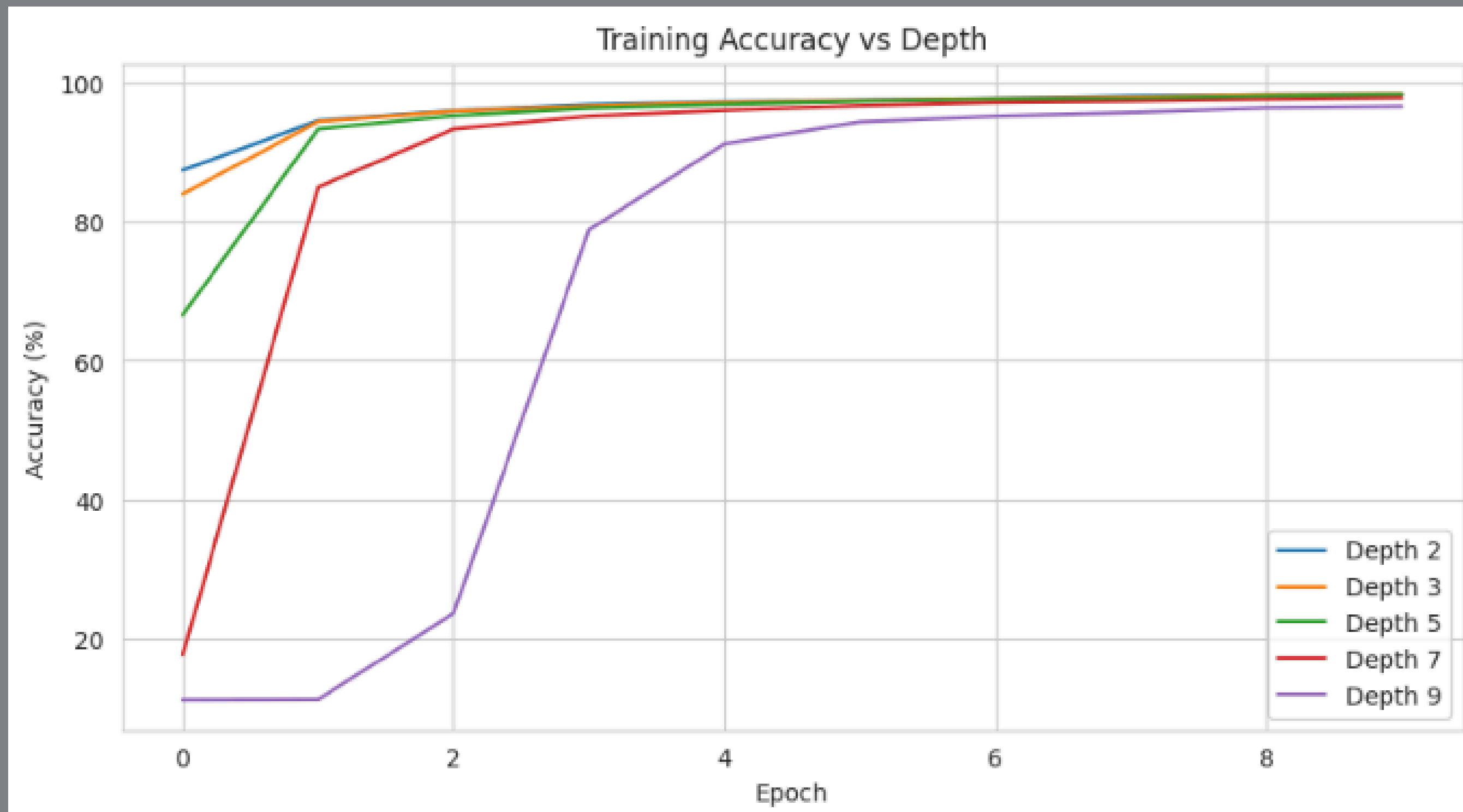


Depth vs. Performance

- **Shallow networks: Struggle with feature extraction.**
- **Deep networks (5-7 layers): Achieve optimal accuracy.**
- **Extra deep networks (10+ layers): Show degradation due to vanishing gradients & overfitting.**



Depth vs. Performance





Thank You!