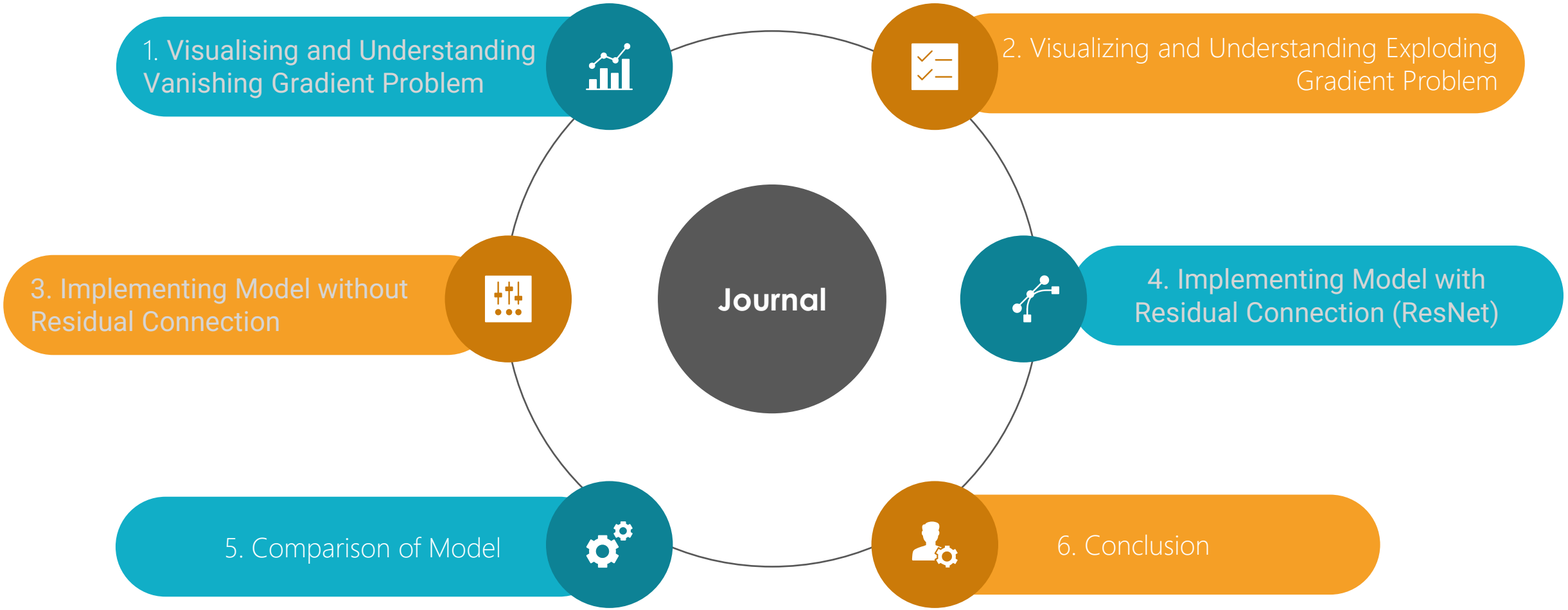
Residual Connections in Deep Neural Networks

Understanding Gradient Challenges in Neural Networks: The Efficacy of Residual Blocks (ResNet)

Subject: 32513 Advanced Data Analytics Algorithms, Machine Learning

Presented by: Gribesh Dhakal (24594374)

Journal Analysis



Problem Analysis

Gradient in Neural Networks

Represents the change in loss with respect to a change in model weights.

Calculation

Derived using the chain rule in calculus during the backpropagation process.

Vanishing Gradient

Gradients become very small; earlier layers learn slowly, leading to prolonged training.

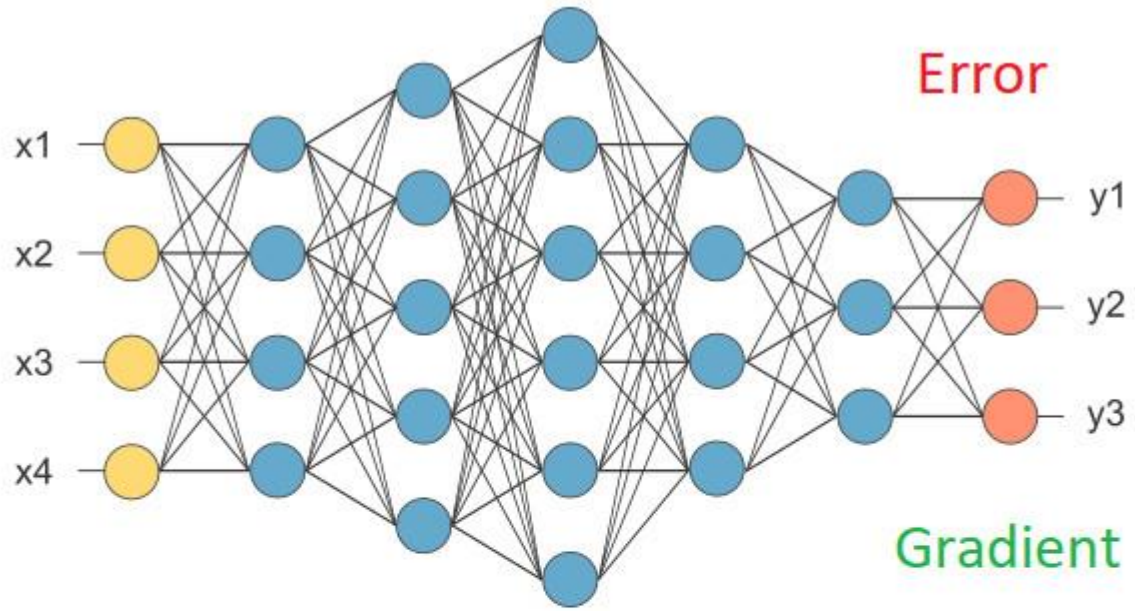
Exploding Gradient

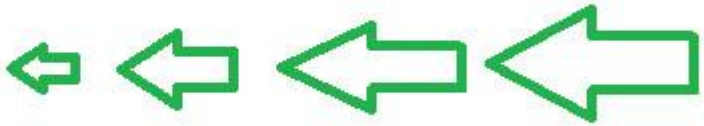
Gradients grow very large; can cause unstable training and large weight updates.

Impact on Training

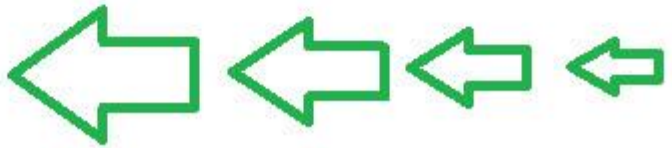
Affected convergence, potential underfitting (for vanishing) or erratic behaviour (for exploding).

Gradient Problem



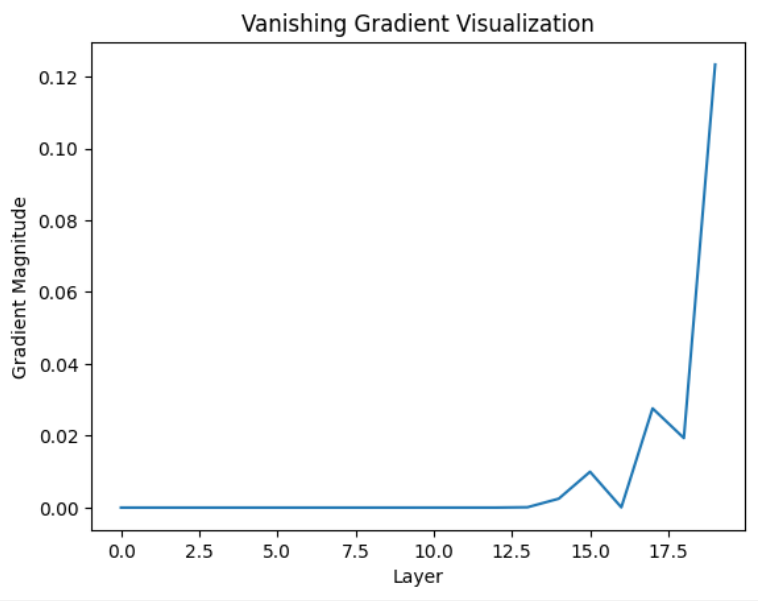


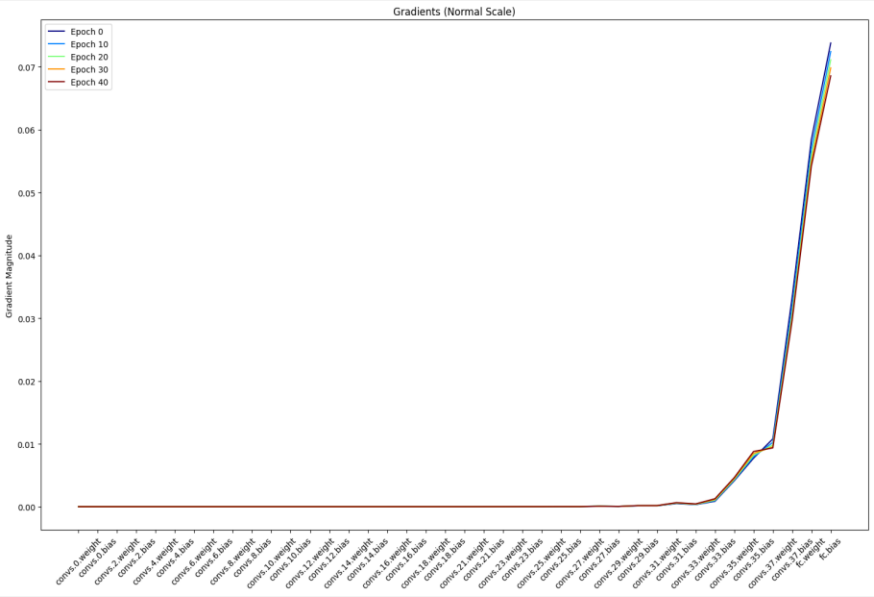
Vanishing Gradient

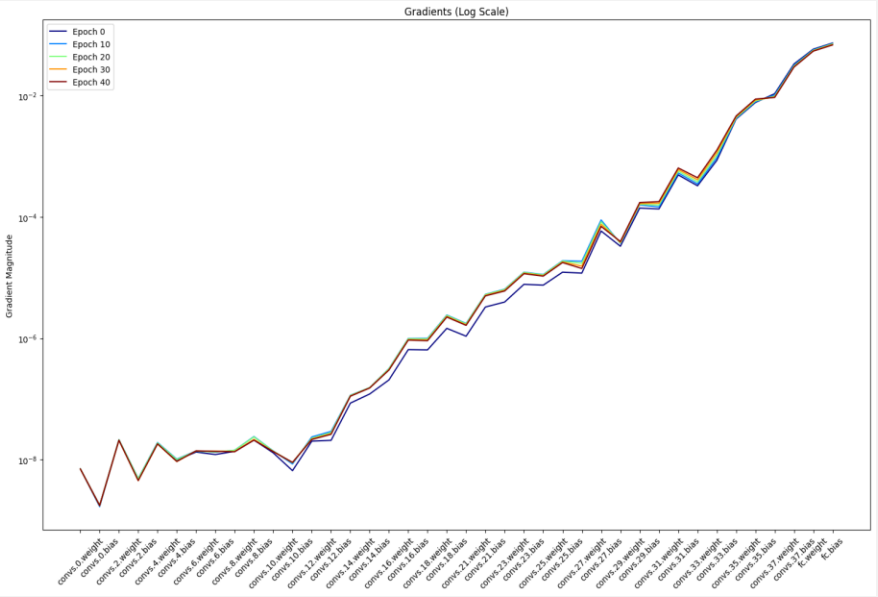


Exploding Gradient

Vanishing Gradient Analysis







Variation of Gradient Magnitude

Simple CNN (Normal Scale)

<0.12

Normal Scale

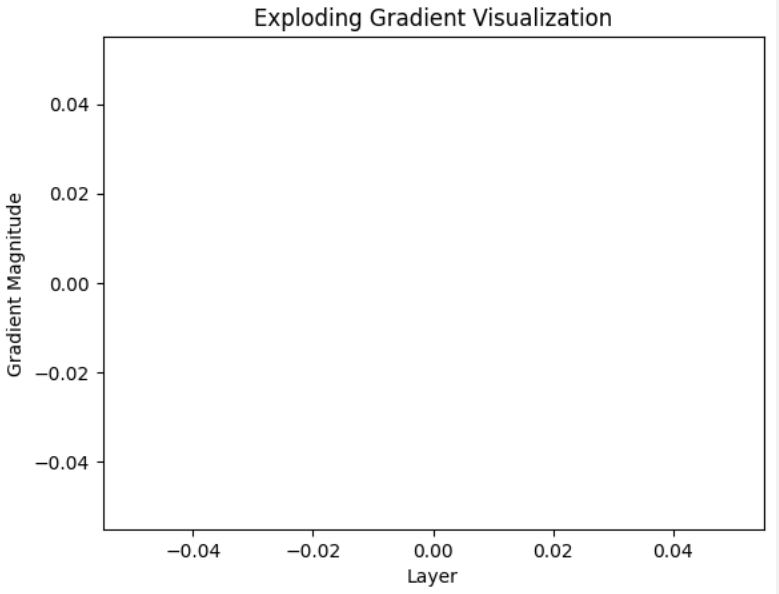
<0.07

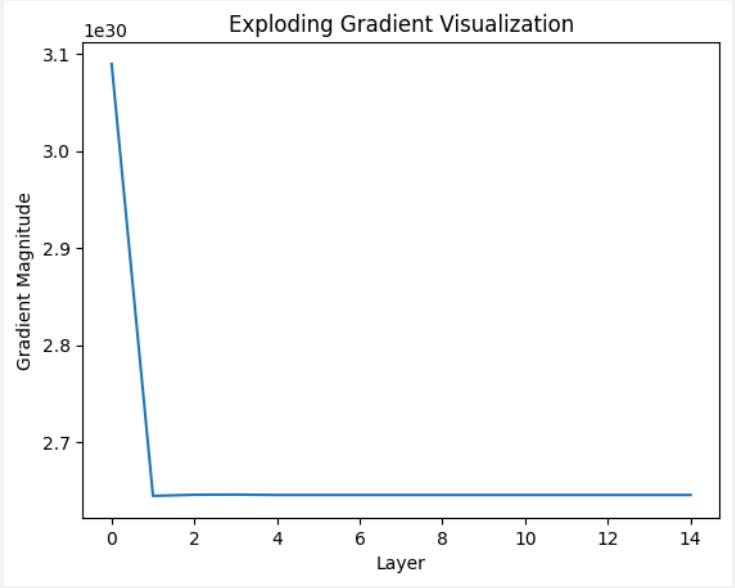
Log Scale

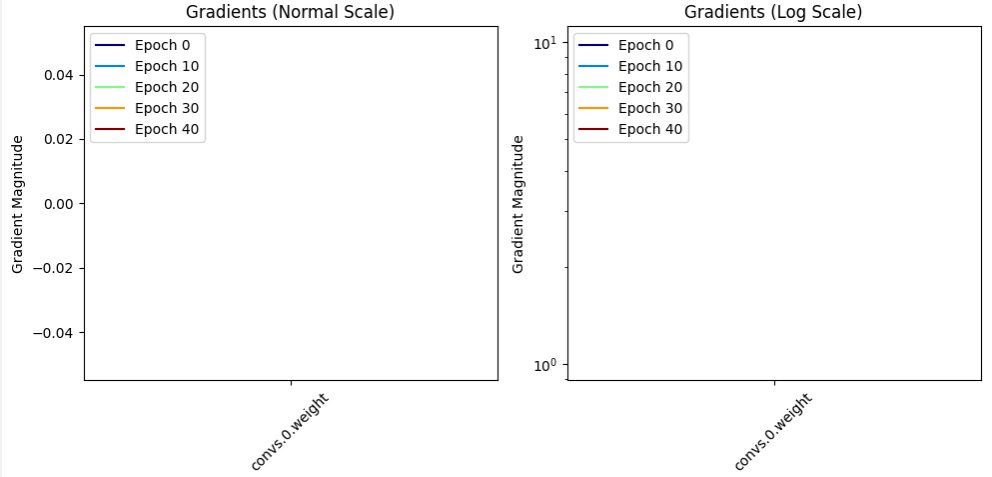
10^{-2} to 10^{-8}

As we move from last layer to first, gradient diminishes.

Exploding Gradient Analysis







Simple CNN

NaN or ∞

Simple CNN

~1e30

Complex CNN

NaN or ∞

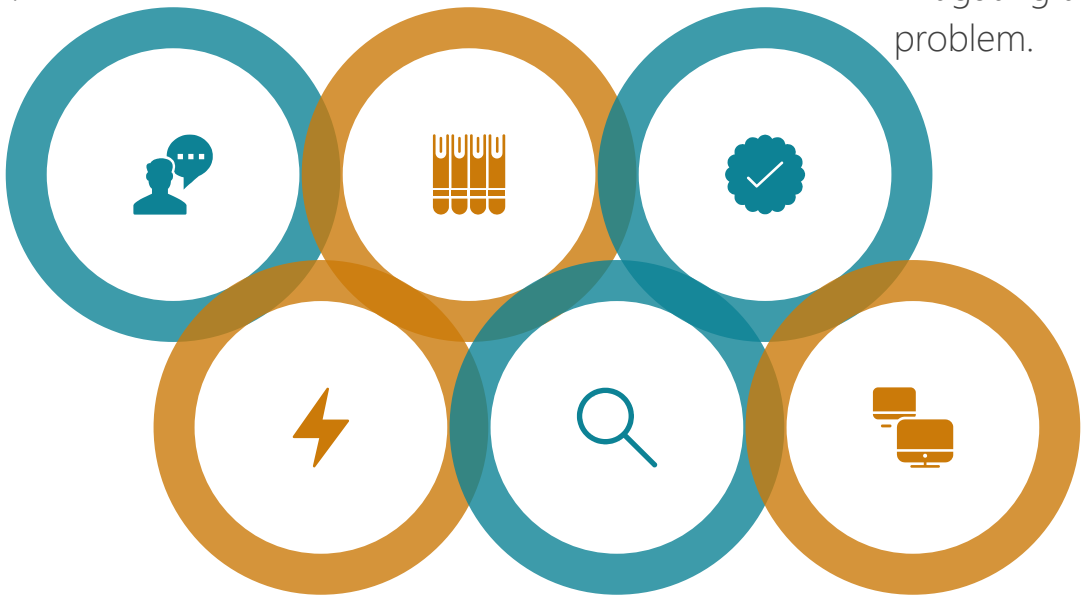
Gradient Increase Sharply as we move from last layer to first layer.

Residual Connection

1. Concept:
Allows activation from one layer to bypass one or more layers and directly connect to a later layer.

2. Shortcuts
Uses "skip connections" or "shortcuts" to jump over certain layers.

3. Addressing Vanishing Gradient:
Improves backpropagation by providing an alternate path, mitigating the vanishing gradient problem.



4. Training Deep Networks:
Enables training of much deeper networks by ensuring adequate gradient flow.

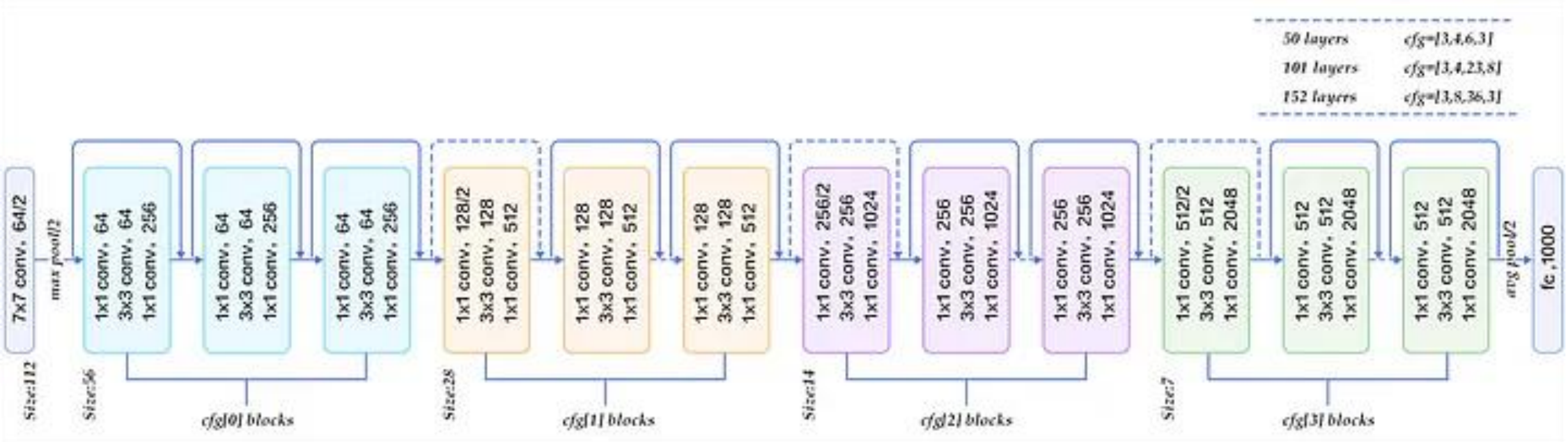
5. Preserving Identity:
Allows the model to learn identity functions which help in stacking layers without hindrance.

6. Enhanced Feature Propagation:
Promotes the reuse of features, thus making the network more efficient.

Different ResNet Models

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				

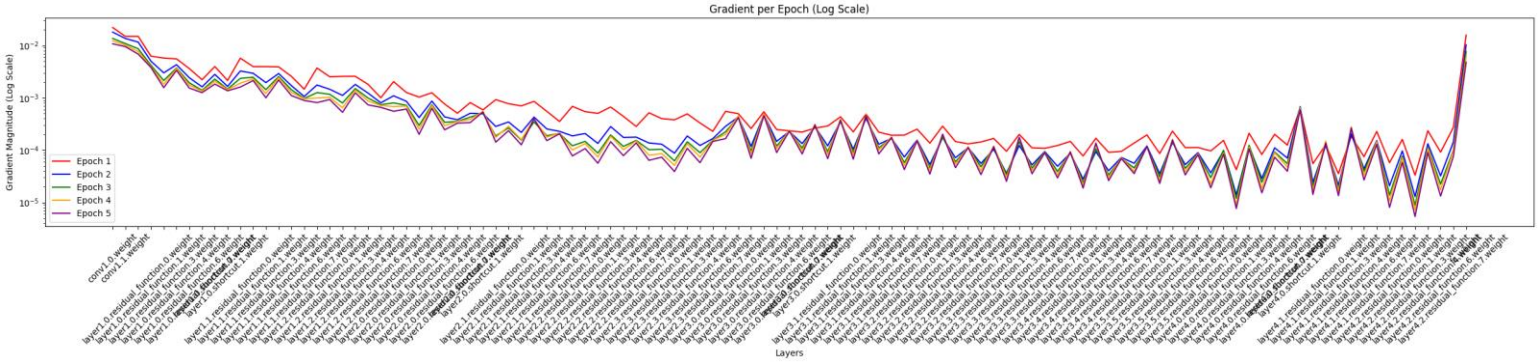
ResNet50 Architecture



Comparison
of Model with and without Residual Block

Without Residual Blocks	With Residual Blocks
› Often slower due to gradient issues. › Plateaus or even degrades with increased depth. › Susceptible to both vanishing and exploding gradients.	› Faster convergence and training due to skip connections assisting gradient flow. › Consistently high accuracy, benefiting from increased depth especially on datasets I have used. › Mitigates the vanishing gradient problem; more stable gradient flow.
› Longer, requiring more epochs for comparable performance. › Performance degradation with increasing depth. › Loss fluctuations and potential stagnation during training.	› Quicker training times due to efficient gradient flow and faster convergence. › Can handle increased depth without performance deterioration; exploits depth for complex datasets. › More stable training with fewer instances of loss fluctuations; consistent gradient flow aids in stability.

Conclusion

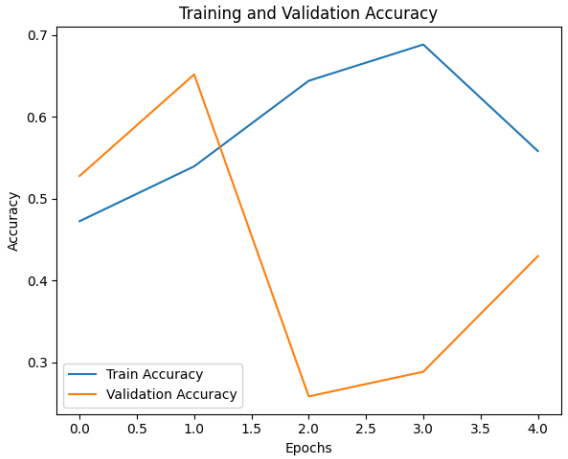


Significance of Gradients

Gradients are pivotal in training deep neural networks. Their magnitude directly impacts the speed and stability of the learning process.

Challenges with Deep Networks

As networks become deeper, they are prone to issues like vanishing and exploding gradients. These issues can severely impede model performance and training efficiency.



Predicted: Tomato_healthy | Actual: Tomato_healthy

Residual Connections' Efficacy

Implementing residual connections, as seen in ResNet architectures, effectively addresses the gradient problems, enabling the successful training of ultra-deep networks and ensuring feature preservation across layers.

Thank You