**Title: Analysis of Activation Functions and Network Depth in Deep Neural Networks using Sine Wave Data**

---

**1. Introduction**

This report explores the implementation of a simple Deep Neural Network (DNN) to learn a sine wave pattern. The focus is on evaluating different activation functions (Sigmoid, Tanh, ReLU), comparing ReLU variants (ReLU, Leaky ReLU, Parametric ReLU), and analyzing the impact of network depth on performance.

---

**2. Dataset Overview**

- **Source**: Synthetic sine wave data provided in CSV format.
- **Structure**: Single column named "Wave" with 5001 float values representing a sine wave.
- **Preprocessing**:
  - A sliding window approach (window size = 50) is used to create input-output pairs.
  - MinMaxScaler is applied to normalize both input features and output labels.
  - Data is split into training (80%) and testing (20%) sets.

---

**3. Activation Functions Evaluation**

**Objective**: Compare the learning performance of models using Sigmoid, Tanh, and ReLU.

- **Model Architecture**:
  - Input layer (50 features)
  - Two hidden layers with 64 neurons each
  - Output layer with 1 neuron

- **Results**:
  - **Sigmoid** and **Tanh** showed higher validation loss and slower convergence due to the vanishing gradient problem.
  - **ReLU** significantly outperformed both, achieving lower loss and faster training.

- **Conclusion**:
  ReLU mitigates the vanishing gradient issue, making it more effective for deeper

networks.

---

**4. ReLU Variants Comparison**

**Objective**: Evaluate how different ReLU variants affect learning performance.

- **Variants Tested**:
  - **ReLU**
  - **Leaky ReLU** (alpha = 0.01)
  - **Parametric ReLU** (trainable alpha)

- **Architecture**: Same as above with 2 hidden layers.

- **Findings**:
  - **Leaky ReLU** and **PReLU** slightly outperformed ReLU.
  - These variants handle the "dying ReLU" issue better by allowing small gradients for negative inputs.

- **Conclusion**:
  ReLU variants can offer slight improvements, especially in deeper or more complex models.

---

**5. Impact of Network Depth**

**Objective**: Assess how increasing depth influences learning accuracy.

- **Tested Depths**: 1 to 5 hidden layers
- **Activation Function**: ReLU

- **Results**:
  - **1-2 layers**: Underfitting, higher loss.
  - **3-4 layers**: Optimal performance, lowest loss.
  - **5 layers**: Performance plateaued or slightly worsened due to increased complexity.

- **Conclusion**:
  Increasing depth improves performance up to a point, after which overfitting or training difficulties may arise. A balance is required to avoid underfitting or overfitting.

---

**6. Final Remarks**

This experiment highlights the importance of:
- Choosing effective activation functions (e.g., ReLU over Sigmoid/Tanh).
- Considering advanced ReLU variants for deeper models.
- Tuning network depth to avoid underfitting or overfitting.

Future improvements may include using techniques such as dropout, batch normalization, or residual connections to enhance deeper network performance.