NAME: ADAMO IDOWU IBRAHIM

**Part 1: Understanding Neural Networks**

**1. What is a Neural Network?** A neural network is a computational model inspired by the structure and function of the human brain. It consists of layers of interconnected nodes or "neurons" that can process information using a connectionist approach to computation. In machine learning, neural networks are used to approximate functions that can relate inputs to outputs in complex, non-linear ways. They learn from examples, adjusting internal parameters (weights and biases) to minimize error on tasks like classification, regression, or even generation of data.

**2. What are Neurons in Neural Networks?** In neural networks, neurons are the fundamental units of computation. Each neuron receives input from other neurons or directly from the input data, processes this input through a weighted sum, adds a bias, then applies an activation function to produce an output. Neurons are used to detect patterns or features within the data, transforming inputs into something that the next layer can further interpret or act upon.

**3. What is an Activation Function?** An activation function determines whether a neuron should be activated or not based on its input. It introduces non-linear properties to the network, allowing it to learn and perform more complex tasks. Without activation functions, neural networks would only be capable of linear transformations. Important for tasks requiring non-linear decision boundaries, activation functions help in learning complex patterns by squashing, thresholding, or transforming the weighted sum of the neuron.

**4. What is Backpropagation?** Backpropagation is a method used to calculate the gradient of the loss function with respect to the weights of the network. It works by propagating the error backward through the network, layer by layer, from the output back to the input. This process adjusts the weights and biases of the neurons to minimize the error of the prediction. Backpropagation is crucial for training because it enables the network to learn from mistakes by adjusting how it processes information to better fit the expected output.

**5. What are Layers in Neural Networks?**

* **Input Layer**: This layer receives the input features directly. It does not perform any computation; it just passes the data to the next layer.
* **Hidden Layers**: These intermediate layers process the data through transformations via weights, biases, and activation functions. They are where the network's learning occurs, capturing complex patterns through their depth and connectivity.
* **Output Layer**: This layer produces the final output of the network after all processing. Its structure depends on the task (e.g., one neuron for binary classification, multiple for multi-class).

**6. What is the Role of Weights and Biases in Neural Networks?** Weights control the strength of the connection between neurons. During training, weights are adjusted to emphasize or de-emphasize certain inputs. Biases add flexibility by allowing the network to fit the best line (or hyperplane in higher dimensions) through the data points, shifting the activation function to the left or right. Together, they determine how input data is transformed into output, affecting the network's ability to model data.

**7. What is Overfitting in Neural Networks?** Overfitting occurs when a neural network learns the training data too well, including its noise and outliers, to the detriment of its performance on new data. The model becomes too specialized to the training set, losing generalization capability. Prevention methods include:

* **Regularization** (like L1/L2 regularization) to penalize large weights.
* **Dropout** to prevent neurons from co-adapting too much.
* **Early Stopping** during training when validation error begins to increase.
* **Data Augmentation** to increase the diversity of the training data.
* Using a **simpler model** or reducing network complexity.

**Part 2: Activation Functions**

**Chosen Activation Function: Swish**

**1. Mathematical Formula:** The Swish function is defined as:

f(x) = x \cdot \sigma(x)

where

\sigma(x)

is the sigmoid function:

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

**2. Behavior of the Activation Function:**

* **Non-linearity**: Swish is continuously differentiable and non-linear, providing a smooth transition between linear and non-linear behavior.
* **Unboundedness**: Unlike ReLU, which has an upper bound, Swish can take on positive values without bound, but it's bounded below by zero due to the sigmoid multiplication.
* **Smoothness**: It's smooth, which can lead to better gradient flow compared to functions with sharp changes.

**3. Where and Why It's Used:** Swish can be used in any layer where ReLU or its variants are typically applied. It's particularly useful in deep networks because:

* It consistently outperforms ReLU in several benchmarks, especially in deeper models.
* It mitigates the "dying ReLU" problem where neurons can become inactive and never reactivate.
* Its smooth curve can lead to better optimization dynamics, especially in scenarios where the data distribution might benefit from a less abrupt transition at zero.

**4. Advantages and Disadvantages:**

* **Advantages:**
  + Outperforms ReLU in many scenarios due to its adaptive nature, allowing for better gradient flow and expressiveness.
  + It's self-gating, meaning the function itself controls how much of the linear part is passed through, which can be beneficial for learning.
* **Disadvantages:**
  + Computationally more expensive than ReLU since it involves both multiplication and the sigmoid function.
  + Less intuitive than ReLU; the benefits might come at the cost of interpretability.

**5. Real-World Application:** Swish has been applied in various deep learning tasks:

* **Image Classification**: In architectures like MobileNetV3, Swish activation helps in achieving better accuracy with fewer parameters.
* **Natural Language Processing**: Swish can improve performance in models like BERT by allowing better gradient flow through deep layers, enhancing the model's ability to understand complex linguistic structures.
* **Object Detection**: In tasks where the network depth is crucial, Swish's smooth gradient might help in maintaining performance across layers.

This function's versatility makes it suitable for many modern neural network architectures where performance, rather than computational simplicity, is prioritized.