RNN Questions

1. A **Recurrent Neural Network (RNN)** is a type of artificial neural network designed for processing sequential data. RNNs have **loops** that allow information to persist, making them well-suited for tasks where context and order matter, such as **speech recognition, language modeling, and time-series forecasting**.

2. **FNN:** Processes inputs independently; no memory of past inputs.
   Data flows in one direction, from input to output, with no loops. Best for tasks where order doesn't matter, like image classification.
   RNN: Maintains a hidden state that carries information from previous time steps, allowing it to handle sequences.
   Has recurrent connections where outputs from previous steps are fed back into the network. Designed for sequential tasks like speech recognition, time-series forecasting, and natural language processing.

3. The **primary advantage** of using **Recurrent Neural Networks (RNNs)** for **sequential data** is their ability to maintain a **memory of past inputs** through a hidden state, allowing them to capture **temporal dependencies** in data.

4. Training **standard Recurrent Neural Networks (RNNs)**, especially for long sequences, comes with several challenges. The two most significant issues are
   Vanishing Gradient Problem:
   When training RNNs gradients become **exponentially smaller** as they propagate backward through many time steps. This makes it **difficult for the network to learn long-term dependencies**, as earlier layers receive little to no updates.
   Exploding Gradient Problem:
   Opposite to the vanishing gradient issue, sometimes gradients grow **exponentially large**, leading to unstable training.

5. In a **Recurrent Neural Network (RNN)**, the **hidden state** is a key component that allows the network to retain information from previous time steps, helping it process sequential data effectively.
   It Carries Context Across Time Steps and Enables Sequence Processing.

6. The **vanishing gradient problem** occurs when training a **Recurrent Neural Network (RNN)** using **Backpropagation Through Time (BPTT)**, where the gradients become **exponentially small** as they propagate backward through many time steps. This makes it difficult for the network to learn long-term dependencies.
7. One popular architecture that addresses the **vanishing gradient problem** in RNNs is the **Long Short-Term Memory (LSTM)** network.
8. A **Long Short-Term Memory (LSTM) network** is an advanced type of **Recurrent Neural Network (RNN)** specifically designed to handle the **vanishing gradient problem** and effectively learn **long-term dependencies** in sequential data. It has Input Gate, Forget Gate, Cell State, etc.
9. The **Forget Gate** is a crucial component of **Long Short-Term Memory (LSTM)** networks that determines **how much past information should be discarded** from the **cell state** at each time step.
10. LSTMs use three main **gates—Input Gate, Forget Gate, and Output Gate**—to regulate the flow of information through the **cell state**, allowing the network to selectively remember or forget information over long sequences.
11. Speech Recognition & Audio Processing, Natural Language Processing (NLP) & Text Generation, Healthcare & Medical Diagnosis.
12. Video Analysis & Motion Tracking, Speech Recognition & Audio Processing
13. **Backpropagation Through Time (BPTT)** is an extension of the **backpropagation algorithm** used to train **Recurrent Neural Networks (RNNs)** by unrolling them over time. Since RNNs process sequential data, their weights are shared across multiple time steps, requiring a modified learning process.