MAHAM SHAHID
BYTEWISE ML/DL

Question 01: What are Recurrent Neural Networks, and how do they differ from traditional feedforward neural networks?

Answer:
Recurrent Neural Networks (RNN) are a type of neural network which have been designed to process sequences and differ from common feedforward networks in some important ways:
Architecture: RNNs have loops to maintain history and memory of previous inputs whereas feedforward networks treat each input independent.
States Memory: In these cases, in which the input sequence matters since they represent sequential data such as text or time series, an RNN is also able to store information of states on past steps.
Applications: while feedforward networks are used on tasks like image classification, RNNs are applied to sequential data applications.
Examples: Training Difficulty, RNNs are difficult to train (e. g., slow learning/ in some cases you need deeper networks) due to vanishing gradient problem and this limits its capacity for capturing long term dependencies.

Task: Explain the working of RNN, and how information is passed through the network over time.

RNN Reccurent Neural Network disengages sequential data processing by holding a hidden state to remember previous inputs. At every time step, the hidden state is modified according to both its present input and output from last cell, so that we incorporate context of previous knowledge. At each step, the output is obtained based on an old hidden state. RNNs are trained using Backpropagation Through Time (BPTT), however, long-term dependencies can be difficult for RNN to learn due to the problem of vanishing gradients. Deep learning variants that include LSTMs and GRUs as well attempt to resolve these challenges by providing mechanisms for preserving information over longer sequence lengths.

Question 02: Discuss the advantages and potential drawbacks of stacking RNN layers. What are Bi-directional RNNs, and how do they enhance the performance of sequence models?

Answer:
So, Stack RNNs together will increase the capacity of the model (How much complex patterns it can remember) and also able to getting better representation after hierarchical pattern one over other. But this will also lead to higher computational complexities, training issues while it may enhance overfitting.
Bi-directional RNNs can be trained to process sequences in a forward and backward direction running its outputted results from both start to end as well as end to start, ultimately providing richer context making it useful for many tasks which involve understanding prerequisite steps or determining the future outcome of words. The drawbacks are the more expensive computational process and, for example in real-time applications where future data is unknown, we will not be able to sorry!

Task: Explains when and why you would use stacked RNN layers and bi-directional RNNs in a sequence modeling task.
Stacked RNN layers: This is useful when your data need to capture complex pattern and long-term dependencies in a sequence — by stacking multiple recurrent cells, the model can learn deeper feature representations.
Bi-directional RNNs should be used for tasks which require past and futur context understanding (language processing, speech recognition etc) in order to increase accuracy by using information from the whole sequence.

Question 03: What is a hybrid architecture in the context of sequence modeling? Provide examples of how combining RNNs with other deep learning models can enhance performance.

Answer:
In sequence modeling, hybrid architecture is a mix of RNNs and other deep learning models such as CNN or Transformer to complement each others power. Thus, for instance, the marriage between CNNs and RNNs in a hybrid model like that seen above can leverage spatial features to be learned by ConvNets prior to sequence modeling via Recurrent Neural Networks; or combining natural sequential nature of Recurrent models with long-ranged dependency handling through Transformers is often more fruitful as well — this would benefit tasks such as video understanding or language translation.

Question 04: List down types of RNN model and explain their structures and differences with RNN.

Answer:
Long Short-Term Memory (LSTM)
The structure: LSTMs have the following gates (forget gate, input gate,outputgate) which control how much of information is to be passed through.
Contrast: standard RNNs are not able to learn as well over long-term dependencies because of their vanishing gradient problem, and LSTMs do address this issue.
GRU (Gated Recurrent Unit):
Architecture: GRUs are simpler than LSTMs by folding the forget & input gates into a single update.
Advantage: Are more efficient and faster to train while still maintaining backlog dependencies that are over long periods of time than standard RNNs.
Bidirectional RNN (Bi-RNN):
Pattern: Bi-RNNs operate both forward and backward of sequences using two hidden states
Difference: Unlike normal unidirectional RNNs, Bi-RNNs obtain context from both previous and future inputs.
Deep RNN:
Architecture: Deep RNN means layering multiple of identical RNN layers.
Difference: They capture higher level patterns by adding model depth, as a substitute of one layer RNN.