1. What's the difference between Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) and in which cases would use each one?

Ans)

	CNN	RNN
i)	CNN stands for Convolutional Neural Networks	RNN stands for Recurrent Neural Network
ii)	CNN is considered to be more potent than RNN	RNN includes less feature compatibility when compared to CNN.
iii)	CNN is ideal for image procesing and video procesing	RNN is ideal for text and speech Analysis.
iv)	It is suitable for spatial data likes images	RNN is used for temporal data, also called sequential data.
v)	The network takes fixed size inputs and generates fixed size outputs.	RNN can handle arbitrary input/output lengths.

2. How many dimension must the inputs of an RNN layer have? What does each dimension represent? What about its outputs?

Ans) RNN input needs to have 3 dimensions. It would be batch size, number of steps and number of features. RNN layer can return the

entire sequence of output for each sample (one vector per timestep per sample), if you set return_sequences = True. The shape of this output is (batch_size, timestamps, units)

3. What are main difficulties when training RNN How can you handle them?

Ans.) There are two widely known issues with properly training Recurrent Neural Network, the ~~vani~~ vanishing and exploding gradient problems. The problems can be solved by using LSTM (Long short-term memory). LSTMs solve the problem using a unique additive gradient structure that includes direct access to forget gate's activation

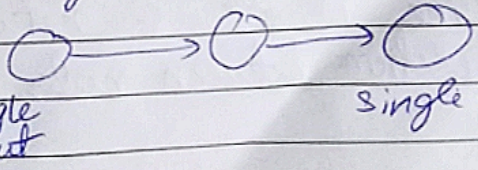
4. What are the uses of using RNN in NLP.

Ans.) RNNs and LSTMs have received the most success when working with sequences of words and paragraph, generally in field of NLP. They are also used as generative models that produce a sequence output not only with text but on application such as generating handwriting

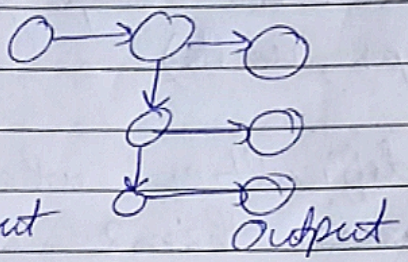
5. What type of Recurrent Neural Networks (RNN) do you know? Explain each of them.

Ans.) Types of RNN

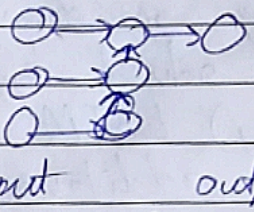
→ One to one RNN - In this type of RNN, there is only one input layer and one output layer



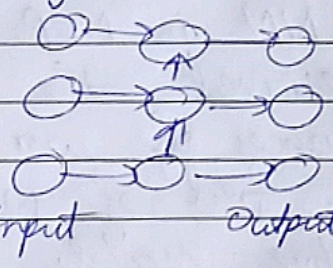
→ One to many RNN - A single input and several output describes one to many RNN



→ Many to one RNN - Several inputs and single output layers.



→ Many to many RNN - Several inputs and several output layers.



6. What is difference between Traditional Feed Forward Networks and Recurrent Neural Network?

Ans:) Feed Forward Neural Network is an artificial neural network where there is no feedback from output to input. One can also treat it as a network with no cyclic connection between nodes.

RNN which is again a class of ANN where there is feedback from output to input. One can also define it as a network where network connection between nodes form directed graph.

7. Why are RNNs better than MLPs at predicting Time Series Data?

Ans) RNNs have the ability to hold a state, that means the model can learn which information it wants to save and what to delete based on ordering and how you designed the creation and passing of state, while this works would be a alot more difficult in MLP.

8. Explain the intuition behind RNN having a Vanishing gradient problem?

Ans) We use sigmoid function in RNN to predict the output so we can get probability output for a certain class. The issues occurs when we are taking derivative and derivative of the sigmoid is always below 0.25 and hence when we multiply a lot of derivative together according to chain rule, we end up with vanishing value such that we can't use them for error calculation.

9. What is backpropagation through time?

Ans) Backpropagation through time is the application of Backpropagation training algorithm to recurrent neural network applied to sequence data like a time series. A recurrent neural network is shown one input each timestep and predict one output. BPTT works by unrolling all input timesteps.

10.

10. Why do we need to use memory cells when using RNNs?

Ans> Due to their internal memory, RNNs can remember important things about the input they received, which allows them to be very precise in predicting what's coming next. This is why they're the preferred algorithm for sequential data like time series, speech, text, financial data, audio, video, weather and much more.