Unit III : Recurrent Neural Networks

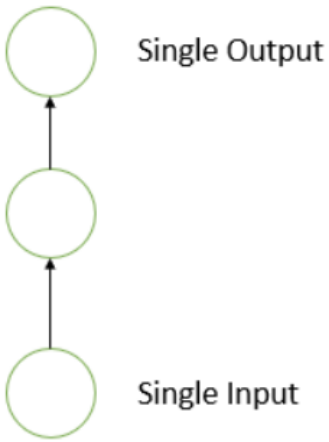
▼ Recurrent Neural Networks

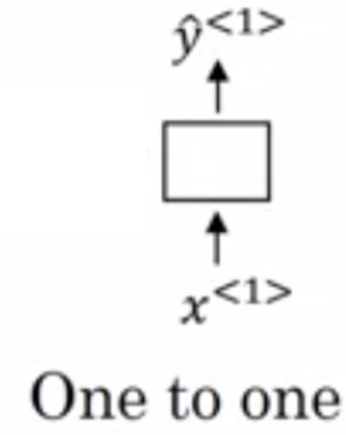
- Recurrent Neural Networks (RNNs) are a type of Neural Network where output from previous step is fed as input to current step.
- RNNs have an infinite impulse response and it is a directed cyclic graph.
- Most important feature of a RNN is the Hidden State, which remembers some information about previous steps or sequence.
- RNNs have a ‘memory’, which remembers all information about what has been calculated.
- RNNs are theoretically Turning complete and run arbitrary programs to produce an arbitrary sequence of outputs.
- Some applications of RNNs include : Handwriting Recognition, Speech Recognition, etc.

▼ Types of Recurrent Neural Networks

1. **One to One RNN**

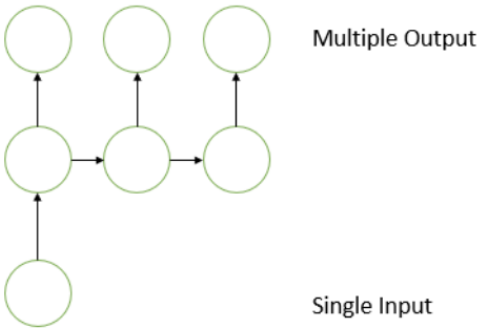
   Used to solve general ML problems that have only one input and one output.

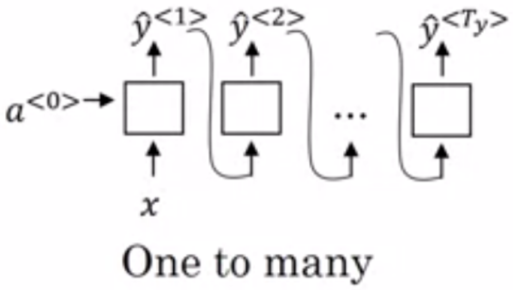




2. **One to Many RNN**

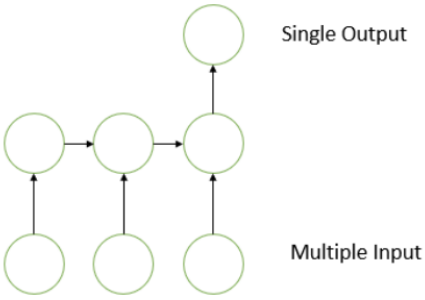
   Single input and Multiple Outputs. Used in Music Generation models.

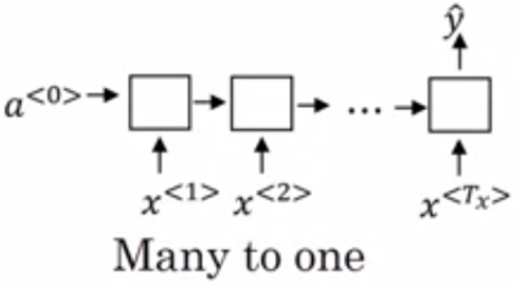




3. **Many to One RNN**

   Multiple inputs and a single output. Used in Sentiment Analysis Models.





4. **Many to Many RNN**

   Multiple inputs give multiple outputs. They are of two types :

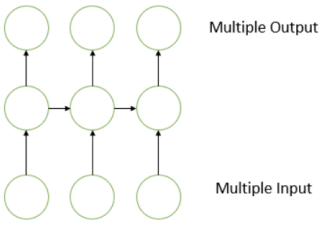
   a. Tx = Ty

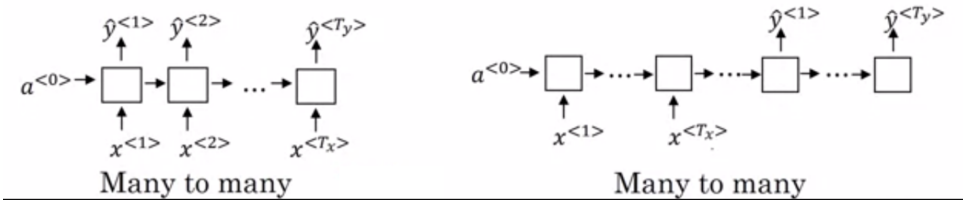
      Input and Output layers are of the same size. Used in Named Entity Recognition.

   b. Tx ≠ Ty

      Input and Output layers are of different sizes. Used in Translation.

      For example, "I Love you", the 3 magical words of the English language translates to only 2 in Spanish, "te amo".

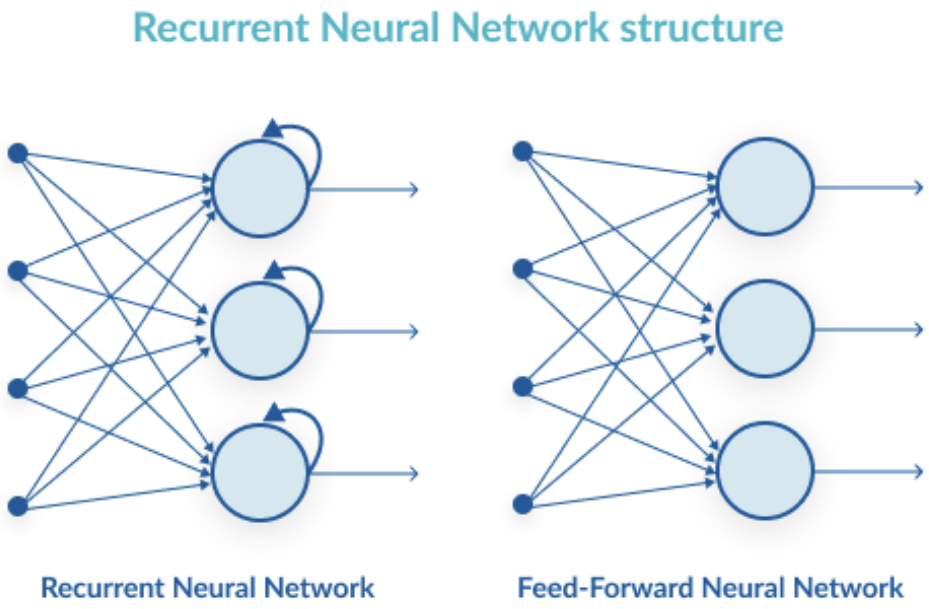




▼ Feedforward Neural Networks vs Recurrent Neural Networks

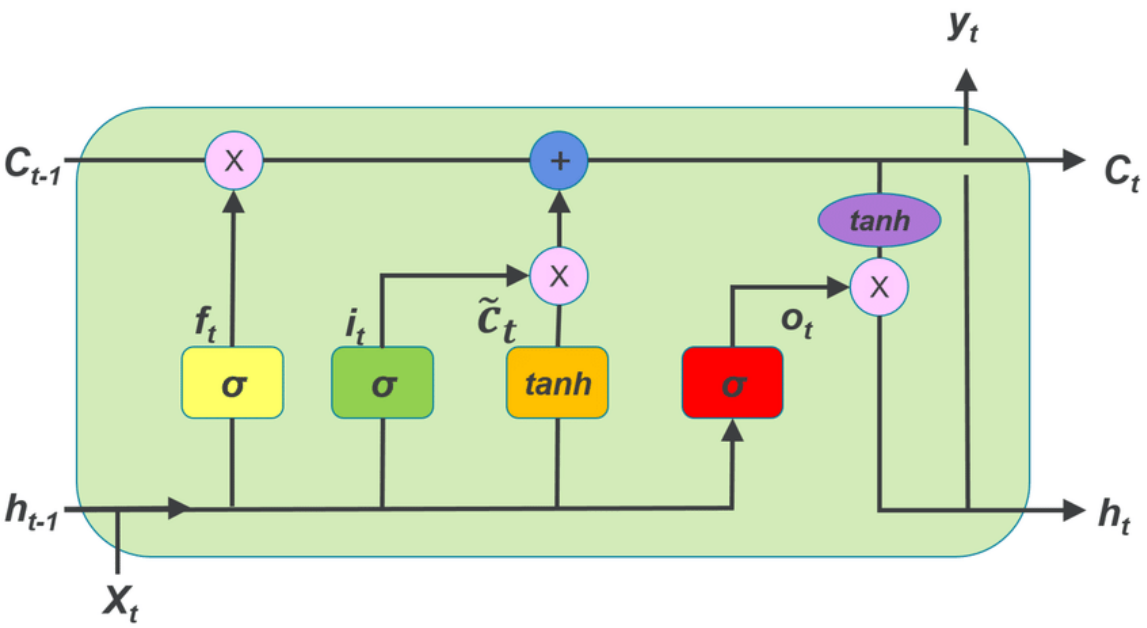
Feedforward NN	**Recurrent NN**
1. Connections between Nodes do not form a cycle.	1. Connections between Nodes forms a cyclical directed graph.
2. Information travels in one direction i.e. it never goes backwards.	2. RNNs have a feedback loop, where data can be fed back into input at same point before it is fed forward again.

Feedforward NN	Recurrent NN
3. Data from prior levels cannot be saved. There is no internal state of memory.	3. RNNs have a memory which remembers some information about previous steps or sequence.



▼ Long Short Term Memory Networks (LSTM)

- LSTM networks are a type of RNN, capable of learning long term dependancies.
- It addressed the issue of RNN long term dependancy, where RNNs are unable to predict words stored in long term memory, but can make more accurate predictions based on current data. RNNs fail to provide efficient performance as gap length rises.
- LSTM may keep information for a long time by default.
- It is used for time series data processing, prediction and classification.



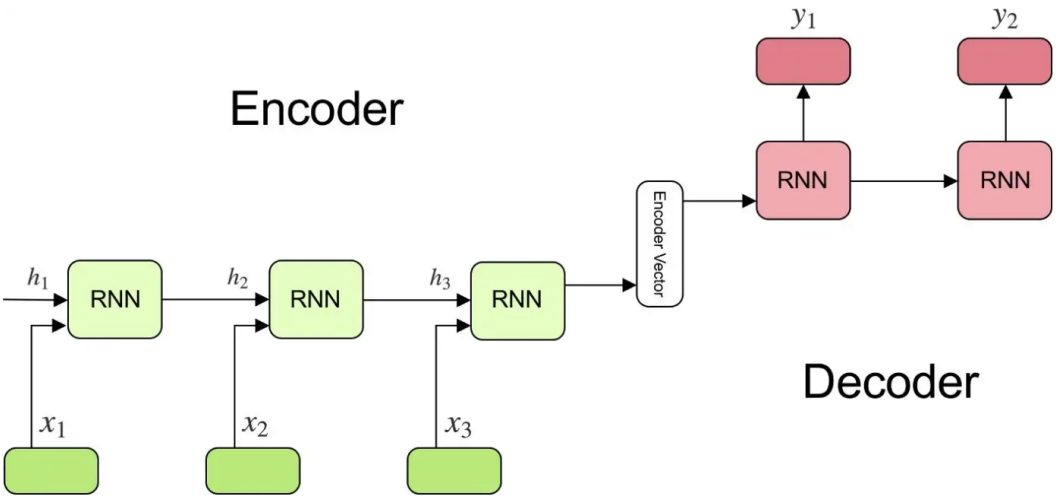
- LSTM Units consist of a cell, an input gate, a forget gate and an output gate.
- LSTM Steps :

1. Using forget gate, information to be forgotten is identified from the prior time step.
2. Using input gate and tanh, new information is sought for updating cell state
3. Information from the two gates is used to update Cell state
4. Dense layers receive output from LSTM Cell. After dense layer, output stage is given a softmax activation function.

▼ Encoder Decoder Architectures

- An encoder takes variable length sequences as input (eg. text) and turns it into numeric representation.

- Decoders unfold a vector representing the sequence state and return something meaningful like - text, tags or labels



Encoder-decoder sequence to sequence model

- Length of input and output may differ. This major advantage opens door to interesting applications like Q/A and video captioning.
- One limitation is that all information needs to be summarised in a 1D Vector, which can be extremely difficult to achieve for long input sequences.
- Architectures -
    1. Many to One
    2. One to Many
    3. Many to Many

▼ Recursive Neural Networks

- A Recursive Neural Network is a kind of Deep Neural Network created by applying same set of weights over a structured input to produce structured prediction over variable size input.
- Recursive Neural Networks can learn Hierarchical Models as opposed to RNNs, that can handle only sequential data

- They have a tree structure - child nodes combine to produce parents, and each child-parent connection has a weight matrix.
- Similar children share the same weights.