Recurrent Neural Network (RNN)

Advance Models

- Recurrent Neural Networks (RNN's)

Introduction to RNN

Think of RNNs as neural networks that have memory. They are designed to work with sequence data, where each piece of data is connected to previous ones. This makes them particularly useful for tasks like time series prediction, language modeling, and sequence generation.

Text → Vector

ChatBot → Question and Answer

RNN → Sequence of words

Language Translation → Hindi → English] RNN

Text Generation →

Gmail → Auto suggestion

↓

RNN

→RNN are type of Neural Network that can be used to model sequence data.

Memory → In RNN, this internal memory allows them to analyze sequential data, where the order of information is crucial.

RNN → 1980

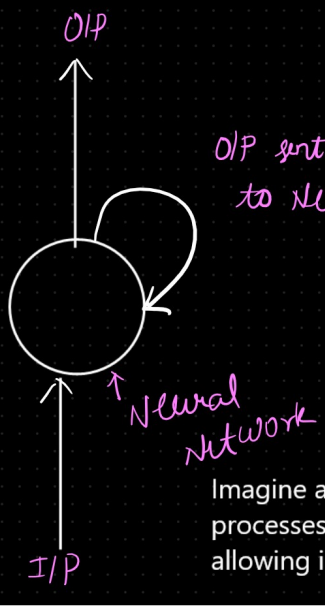
↓

Advanced → Big Data

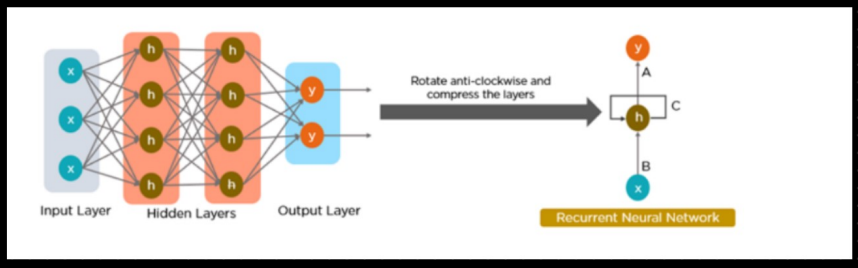
↓

LSTM (Long-Short-Term memory)

Basic Architecture of RNN



Imagine a traditional neural network as a single chain, where each layer processes information independently. In contrast, an RNN has loops within it, allowing information to persist.



All of the inputs and outputs in standard neural networks are independent of one another, however in some circumstances, such as when predicting the next word of a phrase, the prior words are necessary, and so the previous words must be remembered. As a result, RNN was created, which used a Hidden Layer to overcome the problem.

What makes RNN Special?

→ Internal Memory

This is the key feature of RNNs. It allows them to remember past inputs and use that context when processing new information.

→ Sequential Data Processing

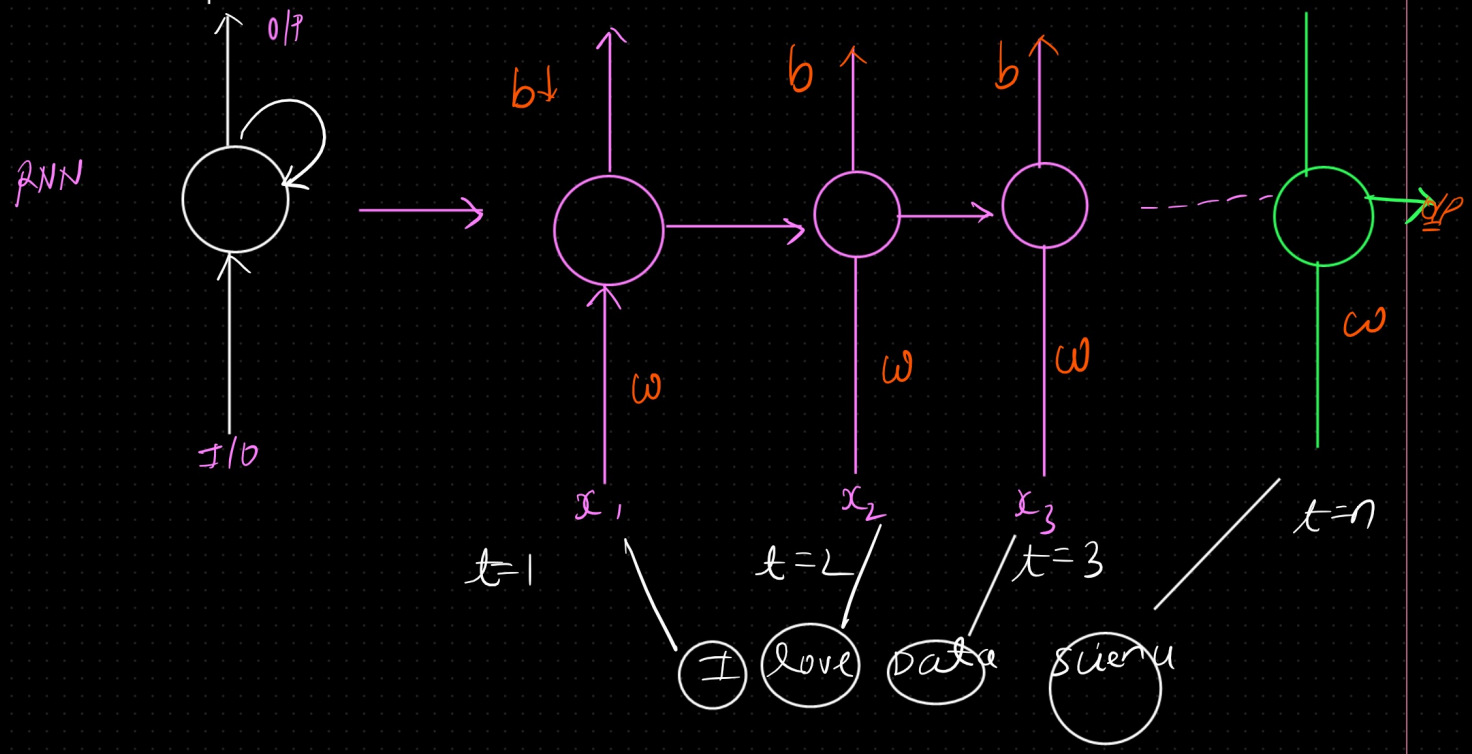
Because of their memory, RNNs are exceptional at handling sequential data where the order of elements matters. This makes them ideal for tasks like speech recognition, machine translation, natural language processing(nlp) and text generation.

→ Contextual Understanding

RNNs can analyze the current input in relation to what they've "seen" before. This contextual understanding is crucial for tasks where meaning depends on prior information.

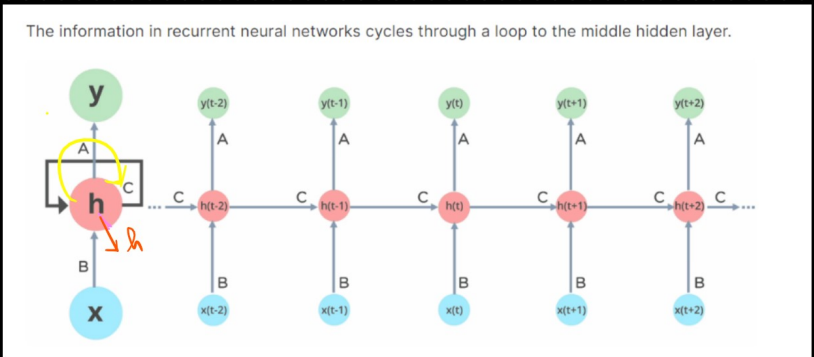
→ Dynamic Processing

RNNs can continuously update their internal memory as they process new data. This allows them to adapt to changing patterns within a sequence.



text → vector

How does RNN works?



The input layer x receives and processes the neural network's input before passing it on to the middle layer.

Multiple hidden layers can be found in the middle layer h, each with its own activation functions, weights, and biases. You can utilize a recurrent neural network if the various parameters of different hidden layers are not impacted by the preceding layer, i.e. There is no memory in the neural network.

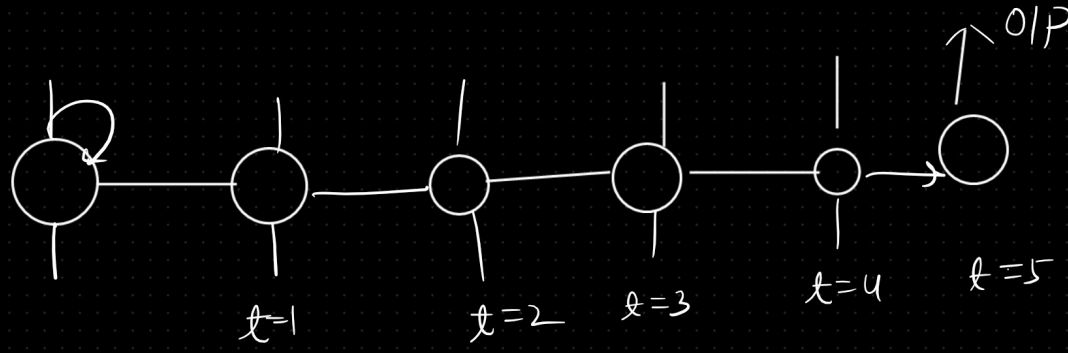
The different activation functions, weights, and biases will be standardized by the Recurrent Neural Network, ensuring that each hidden layer has the same characteristics. Rather than constructing numerous hidden layers, it will create only one and loop over it as many times as necessary.

Types of RNN

1. One-to-one
2. One-to-Many
3. Many to One
4. Many to Many

One-To-One

There is only one pair here. A one-to-one architecture is used in traditional neural networks.



Eg → Text Generation, Movie Recommendation

② One - To - Many

A single input in a one-to-many network might result in numerous outputs. One too many networks are used in the production of music, for example

③ Many - To - One

In this scenario, a single output is produced by combining many inputs from distinct time steps. Sentiment analysis and emotion identification use such networks, in which the class label is determined by a sequence of words.

④ Many - To - Many

For many to many, there are numerous options. Two inputs yield three outputs. Machine translation systems, such as English to French or vice versa translation systems, use many to many networks.

Activation Function

① Sigmoid

→ Squishes any real number between 0 and 1.

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

② tanh

→ Squeezes any real number between -1 and 1.

③ RELU

$\max(0, x)$

→ Outputs the input value if positive, otherwise outputs 0.

④ Leaky ReLu

Similar to ReLU, but for negative inputs, it outputs a small fraction of the input instead of 0.

⑤ Softmax

Converts a vector of real numbers into a probability distribution where all elements sum to 1.

Advantages and disadvantages of RNN
Advantages of RNNs:

Handle sequential data effectively, including text, speech, and time series.
Process inputs of any length, unlike feedforward neural networks.
Share weights across time steps, enhancing training efficiency.
Disadvantages of RNNs:

Prone to vanishing and exploding gradient problems, hindering learning.
Training can be challenging, especially for long sequences.
Computationally slower than other neural network architectures.