SNS COLLEGE OF TECHNOLOGY, COIMBATORE –35
(An Autonomous Institution)
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Recursive Neural Network (RNN)

Recursive Neural Networks are a type of neural network architecture that is specially designed to process hierarchical structures and capture dependencies within recursively structured data. Unlike traditional feedforward neural networks (RNNs), Recursive Neural Networks or RvNN can efficiently handle tree-structured inputs which makes them suitable for tasks involving nested and hierarchical relationships. In this article, we will discuss RvNN, its work principles, and some frequently asked questions.
Recursion is very preliminary concept of DSA where a function or a structure is defined in terms of itself in such way that during execution it calls itself still the used defined criteria or condition is satisfied. When it is used in neural networks, it is just an ability to process data in a hierarchical or nested manner. Here, information from lower-level structures is used to form representations at higher levels. By using this concept, models can capture more complex patterns present in the nested data.
What is RvNN?
From the family of Neural networks, RvNN is a special Deep Learning which has the ability to operate on structured input data like parse trees in natural language processing or molecular structures in chemistry. The network processes the input recursively, combining information from child nodes to form representations for parent nodes. RvNN mainly used in some NLP tasks like sentiment analysis etc. by processing data which is in the format of parse tree. RvNN processes parse trees by assigning vectors to each word or subphrase based on the information from its children which allows the network to capture hierarchical relationships and dependencies within the sentence.

Working Principles of RvNN
Some of the key-working principals of RvNN is discussed below:

1. Recursive Structure Handling: RvNN is designed to handle recursive structures which means it can naturally process hierarchical relationships in data by combining information from child nodes to form representations for parent nodes.
2. Parameter Sharing: RvNN often uses shared parameters across different levels of the hierarchy which enables the model to generalize well and learn from various parts of the input structure.
3. Tree Traversal: RvNN traverses the tree structure in a bottom-up or top-down manner by simultaneously updating node representations based on the information gathered from their children.

4. Composition Function: The composition function in RvNN combines information from child nodes to create a