| Answers | Coding Efficiency | Viva | Timely Completion | Total | Dated Sign of Subject Teacher |
|---|---|---|---|---|---|
| 5 | 5 | 5 | 5 | 20 | |
| | | | | | |

Start Date :..... Date of Completion:.....

## Group B Deep Learning

## Assignment No: 4A

**Title of the Assignment:**

Recurrent neural network (RNN) Use the Google stock prices dataset and design a time series analysis and prediction system using RNN

**Objective of the Assignment:**

Students should be able to perform time series analysis and prediction system using RNN.

**Prerequisite:**

1. Basic of programming language
2. Concept of Classification
3. Concept of Deep Neural Network

---

**Contents for Theory:**

1. What is RNN?
2. What is LSTM?

---

**1. What is RNN**

A recurrent neural network (RNN) is a type of artificial neural network which uses sequential data or time series data. These deep learning algorithms are commonly used for ordinal or temporal problems, such as language translation, natural language processing (nlp), speech recognition, and image captioning; they are incorporated into popular applications such as Siri, voice search, and Google Translate. Like feedforward and convolutional neural networks (CNNs), recurrent neural networks utilize training data to learn. They are distinguished by their "memory" as they take information from prior inputs to influence the current input and output. While traditional deep neural networks assume that inputs and outputs are independent of each other, the output of recurrent neural networks depends on the prior elements within the sequence. While future events would also be helpful in determining the output of a given sequence, unidirectional recurrent neural networks cannot account for these events in their predictions.

Let's take an idiom, such as "feeling under the weather", which is commonly used when someone is ill, to aid us in the explanation of RNNs. In order for the idiom to make sense, it needs to be expressed in that specific order. As a result, recurrent networks need to account for the position of each word in the idiom and they use that information to predict the next word in the sequence.

Another distinguishing characteristic of recurrent networks is that they share parameters across each layer of the network. While feedforward networks have different weights across each node, recurrent neural networks share the same weight parameter within each layer of the network. That said, these weights are still adjusted in the through the processes of backpropagation and gradient descent to facilitate reinforcement learning.

Recurrent neural networks leverage backpropagation through time (BPTT) algorithm to determine the gradients, which is slightly different from traditional backpropagation as it is specific to sequence data. The principles of BPTT are the same as traditional backpropagation, where the model trains itself by calculating errors from its output layer to its input layer. These calculations allow us to adjust and fit the parameters of the model appropriately. BPTT differs from the traditional approach in that BPTT sums errors at each time step whereas feedforward networks do not need to sum errors as they do not share parameters across each layer.

Through this process, RNNs tend to run into two problems, known as exploding gradients and vanishing gradients. These issues are defined by the size of the gradient, which is the slope of the loss function along the error curve. When the gradient is too small, it continues to become smaller, updating the weight parameters until they become insignificant—i.e. 0. When that

occurs, the algorithm is no longer learning. Exploding gradients occur when the gradient is too large, creating an unstable model. In this case, the model weights will grow too large, and they will eventually be represented as NaN. One solution to these issues is to reduce the number of hidden layers within the neural network, eliminating some of the complexity in the RNN model.

**Long short-term memory (LSTM)**:

This is a popular RNN architecture, which was introduced by Sepp Hochreiter and Juergen Schmidhuber If the previous state that is influencing the current prediction is not in the recent past, the RNN model may not be able to accurately predict the current state. As an example, let's say we wanted to predict the italicized words in following, "Alice is allergic to nuts. She can't eat *peanut butter*." The context of a nut allergy can help us anticipate that the food that cannot be eaten contains nuts. However, if that context was a few sentences prior, then it would make it difficult, or even impossible, for the RNN to connect the information. To remedy this, LSTMs have "cells" in the hidden layers of the neural network, which have three gates–an input gate, an output gate, and a forget gate. These gates control the flow of information which is needed to predict the output in the network. For example, if gender pronouns, such as "she", was repeated multiple times in prior sentences, you may exclude that from the cell state.

**Assignment Question No. 4**

1. What is the basic concept of Recurrent Neural Network?
2. For what RNN is used and achieve the best results?
3. What is 'gradient' when we are talking about RNN?
4. One of the RNN´s issue is 'Exploding Gradients'. What is that?
5. The other RNN´s issue is called 'Vanishing Gradients'. What is that?
6. LSTM? What is that?