Deep Learning

Experiment No.4

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

Theory:-

Steps to follow Google Stock Price Prediction using RNN

1. Preparing the data
2. Visualization of data
3. Data pre-processing and further visualization
4. Analyzing the data
    1. First sets of data elements
    2. Second sets of data elements
5. Building the deep learning model
    1. Stacked model
    2. Model summary and plot
    3. Compiling and fitting the model
    4. Analyzing results
6. Making Predictions
7. Conclusion

Preparing the data

The first step to complete this project on stock price prediction using deep learning

```
import pandas as pd

df = pd.read_csv('HistoricalQuotes.csv')
df.head()
```

Visualize Your Data

For any machine learning and deep learning problem, one of the most crucial steps is the visualization of your data. Once you visualize and pre-process the data, you can have a brief understanding of the type of model you are dealing with and the necessary steps and measures required to solve the task. One of the best visualization libraries in the Python programming language is the Matplotlib library. It will allow you to visualize the dataset accordingly. Let us plot the model with the data and their respective indexes. The data consists of the values of the stocks at their respective intervals.

```
# Using the Matplotlib Library for visualizing our time-series data

import matplotlib.pyplot as plt

plt.title("Data Plot")
plt.xlabel("Index")
plt.ylabel("Data")
plt.plot(df1)
```

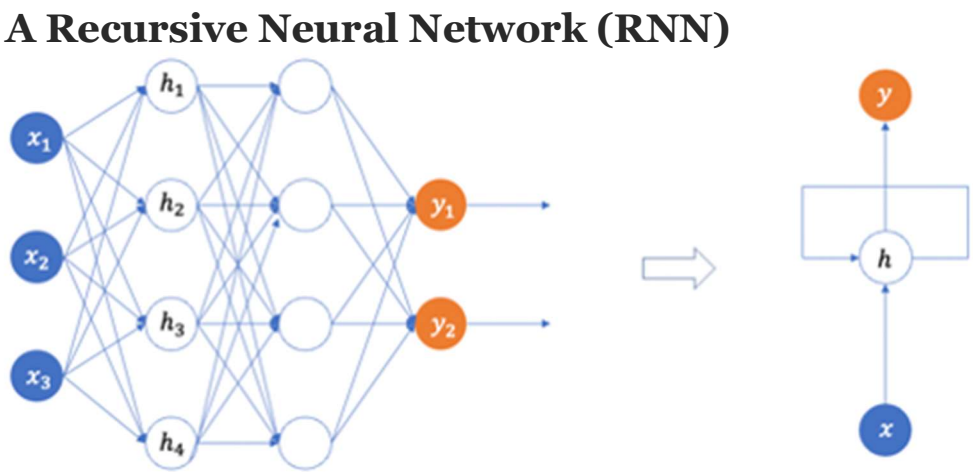

Figure (1): A Recursive Neural Network (RNN)

In order to retain the memory of previous inputs, the Recursive Neural Network (RNN) should be specially designed. The outputs of the previous periods should somewhat become the inputs of the current periods. And the hidden layers will recursively take the inputs of previous periods. In the right-hand side of Figure (1) is a simple graph representation of an RNN. The hidden layer receives the inputs from the input layer, and there is a line to connect a hidden layer back to itself to represent the recursive nature.

What Is a Neural Network?

A Neural Network consists of different layers connected to each other, working on the structure and function of a human brain. It learns from huge volumes of data and uses complex algorithms to train a neural net.

Here is an example of how neural networks can identify a dog's breed based on their features.

The image pixels of two different breeds of dogs are fed to the input layer of the neural network.

The image pixels are then processed in the hidden layers for feature extraction.

The output layer produces the result to identify

Such networks do not require memorizing the past output.

Several neural networks can help solve different business problems.

Feed-Forward Neural Network: Used for general Regression and Classification problems.

Convolutional Neural Network: Used for object detection and image classification.

Deep Belief Network: Used in healthcare sectors for cancer detection.

RNN: Used for speech recognition, voice recognition, time series prediction, and natural language processing.

What Is a Recurrent Neural Network (RNN)?

RNN works on the principle of saving the output of a particular layer and feeding this back to the input in order to predict the output of the layer.

Below is how you can convert a Feed-Forward Neural Network into a Recurrent Neural Network:

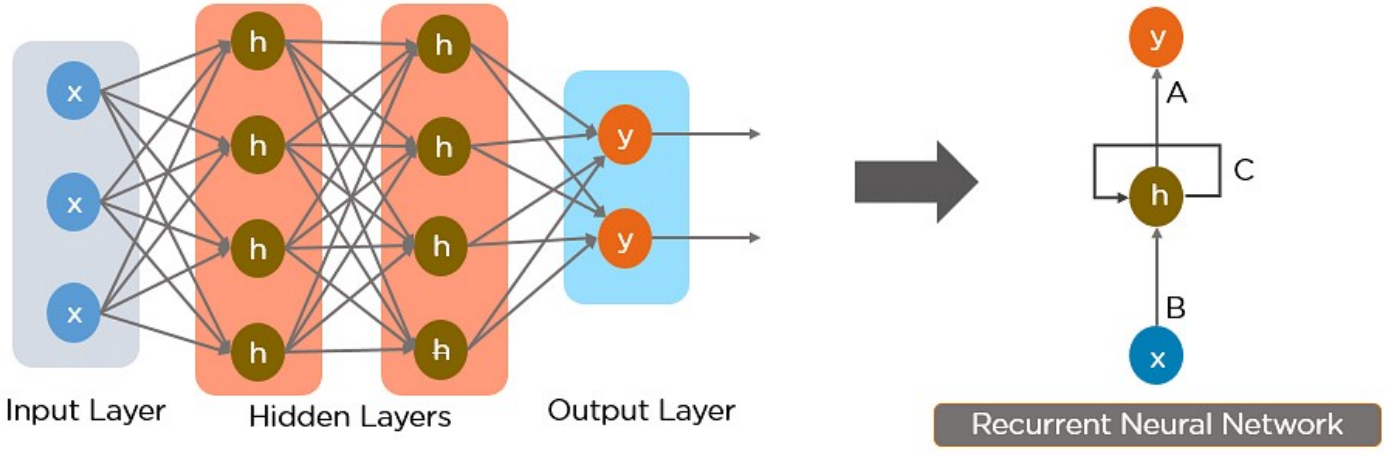

Fig:2 Simple Recurrent Neural Network

The nodes in different layers of the neural network are compressed to form a single layer of recurrent neural networks. A, B, and C are the parameters of the network.

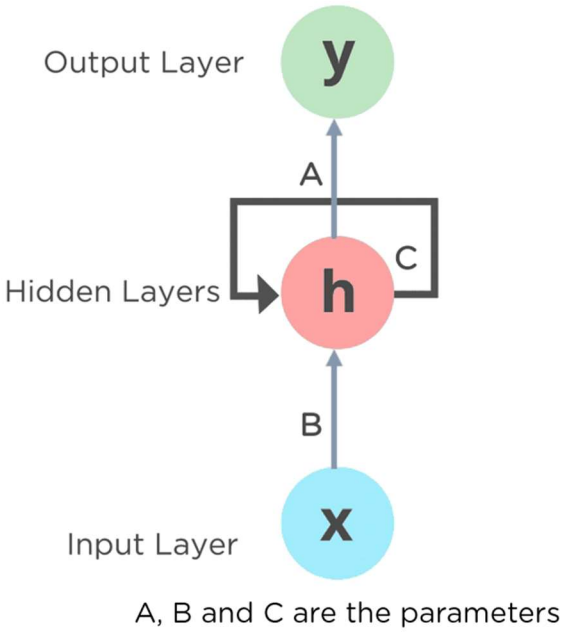

Fig:3 Fully connected Recurrent Neural Network

Here, “x” is the input layer, “h” is the hidden layer, and “y” is the output layer. A, B, and C are the network parameters used to improve the output of the model. At any given time t, the current input is a combination of input at x(t) and x(t-1). The output at any given time is fetched back to the network to improve on the output.

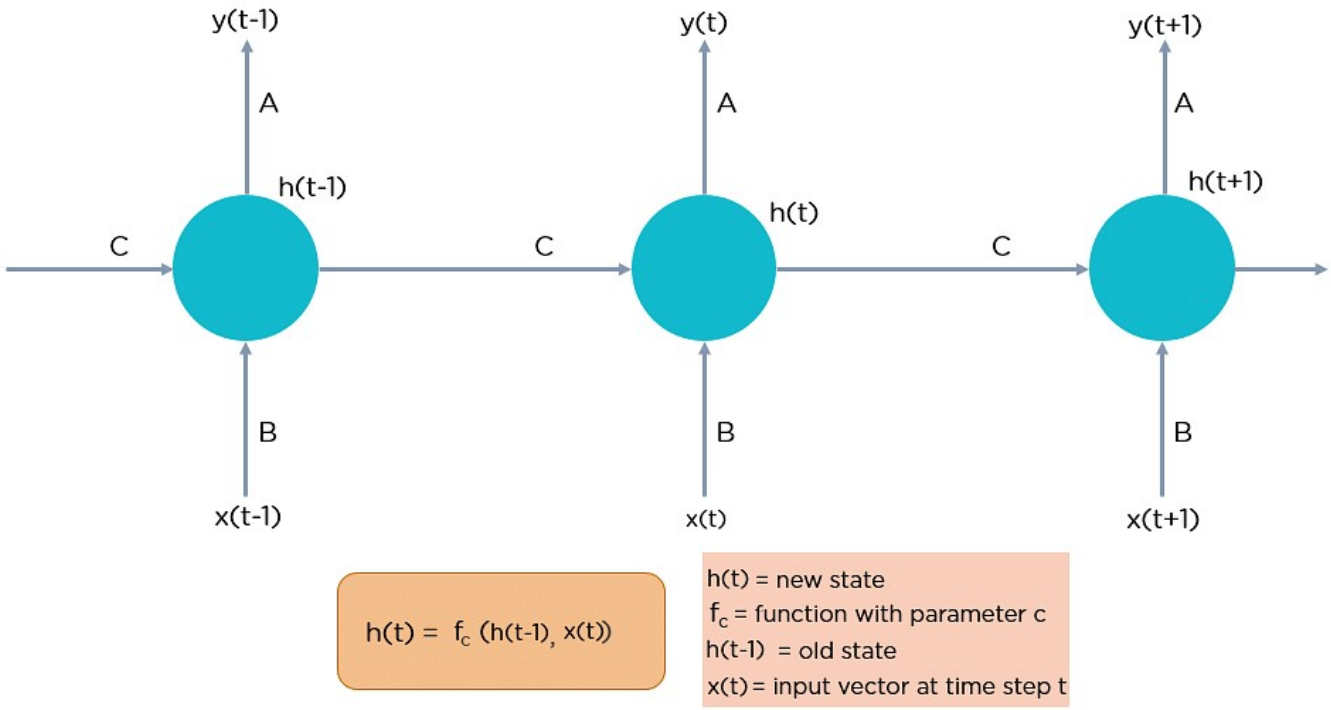

Fig: 4 fully connected Recurrent Neural Network

RNN were created because there were a few issues in the feed-forward neural network:

1. Cannot handle sequential data
2. Considers only the current input
3. Cannot memorize previous inputs

The solution to these issues is the RNN. An RNN can handle sequential data, accepting the current input data, and previously received inputs. RNNs can memorize previous inputs due to their internal memory.

Feed-Forward Neural Networks vs Recurrent Neural Networks

A feed-forward neural network allows information to flow only in the forward direction, from the input nodes, through the hidden layers, and to the output nodes. There are no cycles or loops in the network.

Below is how a simplified presentation of a feed-forward neural network looks like:

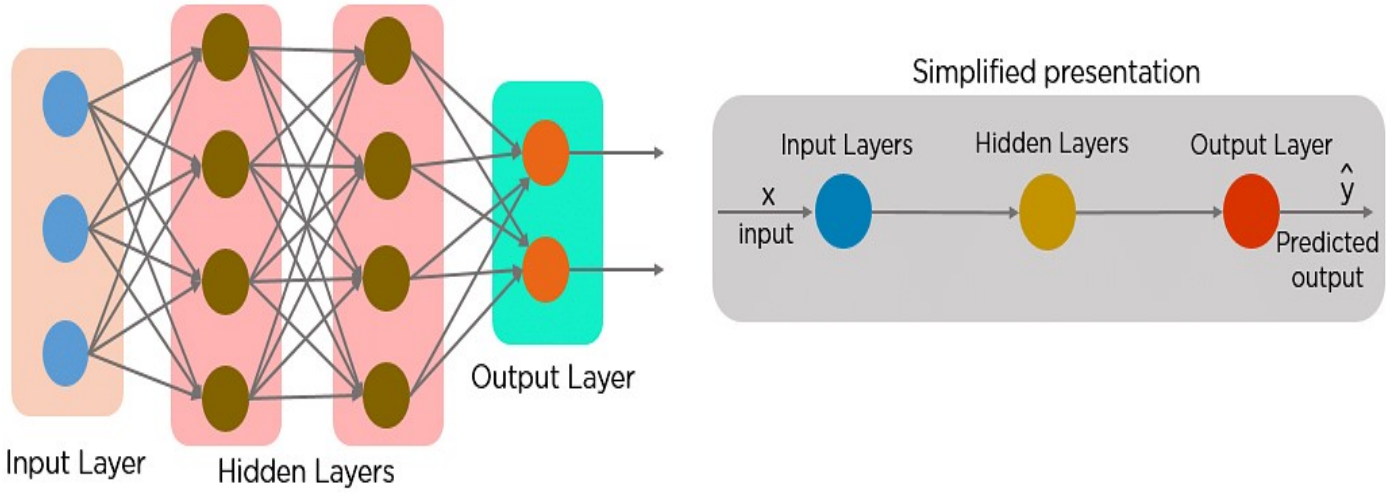

Fig:5 Feed-forward Neural Network

In a feed-forward neural network, the decisions are based on the current input. It doesn't memorize the past data, and there's no future scope. Feed-forward neural networks are used in general regression and classification problems.

Types of Recurrent Neural Networks

There are four types of Recurrent Neural Networks:

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

Applications of Recurrent Neural Networks

RNNs have been shown to achieve state-of-the-art performance on a variety of sequence modeling tasks, including language modeling, speech recognition, and machine translation. These advantages make RNNs a powerful tool for sequence modeling and analysis, and have led to their widespread use in a variety of applications, including natural language processing, speech recognition, and time series analysis.

Limitations of Recurrent Neural Networks

Vanishing And Exploding Gradients

RNNs can suffer from the problem of vanishing or exploding gradients, which can make it difficult to train the network effectively. This occurs when the gradients of the loss function with respect to the parameters become very small or very large as they propagate through time.

Conclusion: - Hence, We have studied and performed recurrent neural network (RNN)**,** Use the Google stock prices dataset, and design a time series analysis and prediction system using RNN.