Assignment 4

Rajdhani
DATE / /

Title :- Recurrent Neural Network (RNN)

Aim:-

Use the google stock price dataset & design a time series analysis & prediction system using RNN.

Objective:-

Apply the technique of Recurrent Neural Network for prediction.

Pre-requisites :-

Concepts of Deep learning & Recurrent Neural Network.

Requirements :-

Jupyter notebook
Python & its libraries

Theory:-

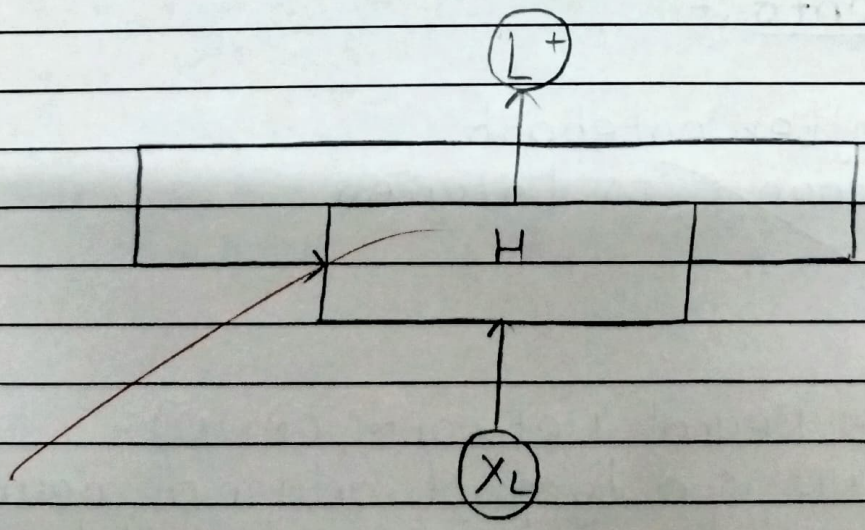
Recurrent Neural Network (RNN):

A RNN is a type of artificial neural network which used sequential data or time series data. RNN members which is useful for predicting stock prices generating text and so on.

Rajdhani
DATE / /

In traditional neural network, the inputs & output are independent of each other but in RNN, the output is dependent on prior elements within the sequence. RNN shares the same weights within each layer of the network & during optimization, the weights and biases are adjusted individually to reduce the loss. RNN use back-propagation through time that sums the error at each time steps it shares parameters across each layers.

The RNN consist of multiple fixed activation function units, on for each time step as it shares parameters across each layer
The RNN consists of multiple fixed activation.



RNN Neuron.

Rajdhani
DATE / /

The input layer XL process the initial input & passes into the middle layer H. This middle layer consist of middle hidden layers, each with its activation function, weights & biases.
In RNN, the information cycle through the loops, so the output is determined by current input and previously received inputs.

Applications include generation of text, machine translation, time series forecasting.

Here's an overview of how RNN works for a classification model.

1. Data Representation :-
Input data is sequential with recurrence connections, updating hidden states at each time step.

2. RNN Architecture :-
RNN process sequential with recurrence connections updating hidden state.

3. Training :-
Input sequence are fed into RNN & parameters are adjusted to minimize loss using backpropagation through time (BPTT).

4. Output layer :-
RNN output layers computes class probabilities using softmax activation.

5. Inference / Prediction :-
Trained RNN predicts class labels for new sequences by selecting the class with the highest probability.

6. Evaluation :-
Model performance is accessed using metrics like accuracy, precision, recall on a separate validation dataset.

Steps :-
1) Import the necessary libraries like Tensorflow, Keras, Numpy, Pandas.
2) Load the google stock price dataset.
3) Preprocess data by normalizing the values & split it into training & testing dataset.
4) Shape the data to fed it into RNN network.
5) Define the RNN architecture with LSTMon GRU neurons in each layer, activation function.
6) Compile the model by specifying loss function, optimizer & evaluation metrics.
7) Train the RNN model on the training set, by specifying epochs & batch size.
8) Evaluate the performance of the model on the testing set.
9) Use the trained model to make predictions on the new unseen time series data.

Rajdhani

DATE / /

Conclusion:-

Hence, we have successfully designed a time series analysis system using RNN, leveraging Google stock price dataset.