**Topic: Recurrent Neural Network (RNN)**

**Instructions:**

**1. Business Problem**

* 1. **Objective**
  2. **Constraints (if any)**

**Using Python perform:**

**2. Data Pre-processing (if applicable)**

**2.1 Data cleaning, Feature Engineering etc.**

**3. Exploratory Data Analysis (EDA): (if applicable)**

**3.1. Summary**

**3.2. Univariate analysis**

**3.3. Bivariate analysis**

**4. Model Building**

**4.3 Using Python libraries perform the below tasks**

**5. Result Share the benefits/impact of the solution - how or in what way the business (client) gets benefit from the solution provided. (If applicable)**

**Note:**

The assignment should be submitted in the following format:

* Python code
* Code Modularization should be maintained
* Documentation of the modules (elaborating on steps mentioned above).

1. **Here is the time series data [110, 125, 133, 146, 158, 172, 187, 196, 210].**

**Build RNN/LSTM model to predict the next 10 digits.**

**Ans:**

* As this is a univariate time series, LSTMs can be used to model univariate time series forecasting problems.
* These are problems comprised of a single series of observations and a model is required to learn from the series of past observations to predict the next value in the sequence.

**Data Preparation:**

* Before a univariate series can be modelled, it must be prepared.
* The LSTM model will learn a function that maps a sequence of past observations as input to an output observation. As such, the sequence of observations must be transformed into multiple examples from which the LSTM can learn.
* We can divide the sequence into multiple input/output patterns called samples, where three-time steps are used as input and one-time step is used as output for the one-step prediction that is being learned.

|  |  |  |  |
| --- | --- | --- | --- |
| X | | | Y |
| 110 | 125 | 133 | 146 |
| 125 | 133 | 146 | 158 |
| 133 | 146 | 158 | 172 |

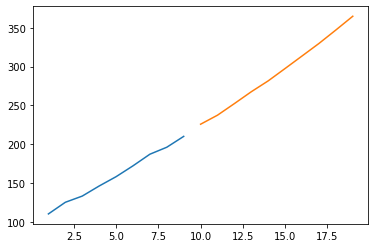
* The *split sequence ()* function below implements this behavior and will split a given univariate sequence into multiple samples where each sample has a specified number of time steps and the output is a single time step.
* The shape of the independent and dependent variables is (6,3)

**Model Building:**

* An LSTM layer requires a three-dimensional input and LSTMs by default will produce a two-dimensional output as an interpretation from the end of the sequence.
* We can address this by having the LSTM output a value for each time step in the input data by setting the *return\_sequences=True* argument on the layer. This allows us to have 3D output from hidden LSTM layer as input to the next.
* The model is now define using the relu activation function with dense “1” and no of epochs given are 300
* Now the loop is created to demonstrate the prediction for next 10 days.
* Here are the below predicted values for the next 10 days

**[225.69,237.35,252.32,267.62,281.64,297.60,313.70,329.86,347.22,364.96]**

**Graphical Representation of output:**

****

1. Write down the multiple applications of RNN.

**Ans:**

RNNs are widely used in the following domains/ applications:

* Prediction problems
* Language Modelling and Generating Text
* Machine Translation
* Speech Recognition
* Generating Image Descriptions
* Video Tagging
* Text Summarization
* Call Centre Analysis
* Face detection, OCR Applications as Image Recognition
* Other applications like Music composition

1. How to do select the inputs for a LSTM/RNN models. Explain in the terms of timesteps, samples and feature.

**Ans:**

The LSTM input layer is specified by the “*input\_shape*” argument on the first hidden layer of the network is used to select the inputs.

* **Samples:** One sequence is one sample. A batch is comprised of one or more samples.
* **Time Steps:** One-time step is one point of observation in the sample.
* **Features:** One feature is one observation at a time step

1. What are the disadvantages of MLP when dealing with sequence data.

**Ans:**

* MLPs do not share features learned across different positions of the data sample.
* MLP always will be restricted to the size which it was trained while the RNN model will be able to add two numbers with even more bits.
* For MLPs, all inputs which are connected to the hidden layers or maybe output layers, have weight. Consequently, increasing the number of inputs will lead to more weights which are not trained yet.