**Recurrent Neural Network (RNN)**

**Instructions:**

Please share your answers filled in-line in the word document. Submit code separately wherever applicable.

Please ensure you update all the details:

**Name: K Sai Manoj Batch ID:** 29072024

**Topic: Recurrent Neural Network.**

**Guidelines:**

**1. An assignment submission is considered complete only when the correct and executable code(s) and documentation explaining the method and results are submitted. Failing to submit either of those will be considered an invalid submission and not a correct submission.**

**2. Ensure that you submit your assignments correctly and in full. Resubmission is not allowed.**

**3. Post the submission you can evaluate your work by referring to the keys provided. (will be available only post the submission).**

**Hints:**

1. **Business Problem**
   1. **What is the business objective?**
   2. **Are there any constraints?**
2. **Work on each feature of the dataset to create a data dictionary as displayed in the image below:**

**Make a table as shown above and provide information about the features such as its data type and its relevance to the model building. And if not relevant, provide reasons and a description of the feature.**

**Using Python to perform the following:**

1. **Data Pre-processing**

**3.1 Data Cleaning, Feature Engineering, etc.**

**3.2 Outlier treatment.**

1. **Model Building**

**4.1 Build a Recurrent Neural Network.**

**4.2 Train and test the model.**

**4.3 Briefly explain the model output in the documentation.**

1. **Write about the benefits/impact of the solution - in what way does the business (client) benefit from the solution provided?**
2. **Use Tensorflow for this assignment. Depending on your system configuration, use either Tensorflow GPU or Tensorflow CPU versions.**

**Problem Statement: -**

1. Here is the time series data [110,125,133,146,158,172,187,196,210]. Build an RNN/LSTM model to predict the next 10 digits. similarly show the next 10 Forecasting in streamlit Framework
2. **Write down the applications of RNN.**

**ANS:**

Recurrent Neural Networks (RNNs) have numerous applications in various fields:

Natural Language Processing (NLP)

1. Language Translation: Google Translate, Microsoft Translator

2. Text Summarization: Summarize long documents, articles

3. Sentiment Analysis: Determine sentiment (positive/negative) of text

4. Chatbots: Customer service, virtual assistants (e.g., Siri, Alexa)

5. Speech Recognition: Voice-to-text systems (e.g., Google Voice)

Speech Processing

1. Speech-to-Text Systems

2. Voice Recognition: Biometric authentication

3. Audio Classification: Music genre classification

Time Series Prediction

1. Stock Market Prediction: Forecast stock prices

2. Weather Forecasting: Predict weather patterns

3. Traffic Prediction: Optimize traffic flow

Image and Video Processing

1. Image Captioning: Generate captions for images

2. Video Analysis: Object detection, tracking

3. Action Recognition: Identify actions in videos

1. **Write about how the inputs are selected for LSTM/RNN models. Explain in terms of timesteps, samples, and features**.

LSTM (Long Short-Term Memory) and RNN (Recurrent Neural Network) models process sequential data, where each input is a sequence of values. To prepare inputs for these models, you need to understand three key concepts:

1. Timesteps (Time Steps or Sequence Length): The number of time intervals or steps in a sequence.

2. Samples (Batch Size): The number of independent sequences or observations.

3. Features (Input Dimensions): The number of variables or characteristics measured at each timestep.

Input Shape

The input shape for LSTM/RNN models is typically:

(samples, timesteps, features)

- samples: Number of independent sequences (batch size)

- timesteps: Length of each sequence (number of time steps)

- features: Number of variables measured at each timestep

Example

Suppose you're predicting stock prices using historical data. Your dataset consists of:

- 1000 stocks (samples)

- 30 days of historical prices (timesteps)

- 5 features: Open, High, Low, Close, Volume

Input shape: (1000, 30, 5)

Data Preparation

To prepare inputs for LSTM/RNN models:

1. Split data into sequences: Divide your data into sequences of equal length (timesteps).

2. Normalize/Scale data: Normalize or scale your data to ensure consistent ranges.

3. Pad sequences: Pad shorter sequences to match the longest sequence (timesteps).

4. Reshape data: Reshape data into the (samples, timesteps, features) format.

Example

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import LSTM, Dense

# Define input shape

input\_shape = (30, 5) # timesteps, features

# Create LSTM model

model = Sequential()

model.add(LSTM(50, input\_shape=input\_shape))

model.add(Dense(1))

model.compile(loss='mean\_squared\_error', optimizer='adam')

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

Disadvantages of MLP (Multilayer Perceptron) for Sequence Data

MLPs are powerful neural networks, but they have limitations when dealing with sequence data:

1. No Temporal Dependencies

MLPs don't account for temporal relationships between sequential data points.

2. Fixed Input Length

MLPs require fixed-length input vectors, making them unsuitable for variable-length sequences.

3. No Spatial Hierarchy

MLPs don't capture hierarchical relationships between sequence elements.

4. Lack of Contextual Understanding

MLPs process individual elements independently, ignoring contextual information.

5. Insensitivity to Order

MLPs don't consider the order of sequence elements.

6. Overfitting

MLPs can overfit complex sequence data due to their rigid architecture.

7. Difficulty with Long-Range Dependencies

MLPs struggle to capture long-range dependencies in sequences.