Name: Yash Gaikwad
Roll No: 381071
PRN: 22420141

Assignment 4

Problem Statement: Time series prediction using RNN – stock market analysis or weather forecasting

Introduction:

Stock market prediction is one of the most challenging tasks in the field of data science due to the dynamic, non-linear, and highly volatile nature of stock prices. Traditional statistical methods often fail to capture the sequential dependencies and temporal patterns present in financial data. Recurrent Neural Networks (RNNs) are well-suited for **time series forecasting** because they are designed to process sequential data, retaining information from previous time steps to make predictions for future values. In this assignment, we implement RNNs for predicting stock price trends using historical stock data.

Objective:

The main objectives of this assignment are:

- To understand how RNNs can be applied to time series prediction.
- To preprocess stock market data for sequential modeling.
- To design and train an RNN for forecasting stock prices.
- To evaluate the model's performance and analyze its ability to capture temporal dependencies.
- To gain insights into the strengths and limitations of RNNs in stock market prediction.

Theory:

1 Time Series Data in Stock Market

- Time series data represents observations recorded sequentially over time.
- Stock prices (open, close, high, low) are classical examples of time series data.
- Predicting future stock prices requires capturing both **short-term fluctuations** and **long-term trends**.

2 Recurrent Neural Networks (RNNs)

- RNNs are a class of neural networks designed to handle sequential data.
- Unlike feedforward neural networks, RNNs have **feedback connections**, allowing them to maintain a memory of previous inputs.
- Each neuron's output is influenced not only by the current input but also by the hidden state from the previous time step.

Key components:

1. **Hidden State** – Stores temporal information from previous steps.
2. **Activation Function (tanh, ReLU)** – Introduces non-linearity.
3. **Output Layer** – Produces prediction (e.g., next stock price).

3 Challenges of Using RNNs

- **Vanishing/Exploding Gradient Problem** when training long sequences.
- Limited ability to capture very long-term dependencies (can be improved using LSTM or GRU).

4 Workflow for Stock Market Prediction

1. **Data Collection & Preprocessing**
    - Obtain historical stock price data (e.g., from Yahoo Finance).
    - Select relevant features (closing price, volume, etc.).
    - Normalize data and create sequences of fixed length for input.
2. **Model Building**
    - Define an RNN model (or variants like LSTM/GRU for better performance).
    - Input layer → RNN/LSTM layers → Dense output layer.
3. **Model Training**
    - Compile with loss function (Mean Squared Error for regression).
    - Train with an optimizer such as Adam/SGD.
4. **Evaluation**
    - Compare predicted stock prices with actual values.
    - Use metrics such as RMSE (Root Mean Square Error) or MAE (Mean Absolute Error).
5. **Prediction & Analysis**
    - Test the model on unseen stock price data.
    - Analyze trends to check how well the model generalizes.

Conclusion:

In this assignment, we applied **Recurrent Neural Networks (RNNs)** for stock market time series prediction. The RNN model was able to learn temporal dependencies from historical stock data and make reasonable predictions about future prices. However, due to challenges like volatility and noise in financial markets, the model's accuracy is limited. This assignment highlights the potential of RNNs in time series forecasting and sets the foundation for using more advanced architectures such as **LSTM** and **GRU**, which can better handle long-term dependencies and improve prediction accuracy in real-world stock market analysis.