Assignment No 4

Name : Samiksha More

PRN : 22420127

Roll : 381070

Problem Statement: Time Series Prediction using Recurrent Neural Networks (RNN) – Stock Market Analysis or Weather Forecasting

Objective

The objective is to implement a Recurrent Neural Network (RNN) to predict future values in a time series dataset, such as stock market prices or weather conditions. The goal is to utilize the ability of RNNs to capture temporal dependencies and sequential patterns in data for accurate forecasting.

Theory

Time series data is sequential, where current values depend on past observations. Traditional models like ARIMA have limitations in capturing complex non-linear patterns.

Recurrent Neural Networks (RNNs) are designed for sequential data processing. Unlike feedforward networks, RNNs have a feedback loop that allows them to store memory of previous inputs.

Key theoretical points:

* Sequential Processing: Inputs are processed step by step, with outputs depending on current and past inputs.
* Hidden State: Maintains context from previous time steps.
* Variants of RNN:
  + LSTM (Long Short-Term Memory): Handles long-term dependencies using gates.
  + GRU (Gated Recurrent Unit): Simplified version of LSTM with fewer parameters.
* Loss Function: Mean Squared Error (MSE) for regression tasks like forecasting.

Methodology

1. Data Collection:
   * Stock market dataset (closing prices, volume, etc.) or weather dataset (temperature, humidity, rainfall).
2. Data Preprocessing:
   * Normalize values for stable training.
   * Convert data into sequences (sliding window approach).
   * Split into training and testing sets.
3. Model Design (RNN Architecture):
   * Input layer: sequential data points.
   * RNN/LSTM/GRU layers: capture temporal dependencies.
   * Dense output layer: predict next value(s).
4. Training:
   * Optimize using Adam/SGD.
   * Use MSE as loss function.
   * Train over multiple epochs until convergence.
5. Evaluation:
   * Compare predicted vs actual values.
   * Metrics: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), R² Score.
6. Prediction:
   * Feed the last sequence of data into the trained model.
   * Predict the next time step(s).

Advantages

* Captures sequential and temporal dependencies in data.
* More accurate than traditional statistical methods for non-linear time series.
* Flexible – works for single-step and multi-step forecasting.
* Applicable to a wide range of real-world problems.

Limitations

* Training is computationally intensive.
* Prone to vanishing or exploding gradient problem (improved by LSTM/GRU).
* Requires large datasets for better generalization.
* Sensitive to noise in data (common in stock markets).

Applications

* Finance: Stock price prediction, trading signal generation.
* Weather Forecasting: Predicting temperature, rainfall, or wind speed.
* Energy: Electricity demand forecasting.
* Healthcare: Patient monitoring (heartbeat, glucose levels).
* Transportation: Traffic flow prediction.

Working / Algorithm

1. Input time series data is fed into the RNN model.
2. At each time step, the RNN processes the input along with its hidden state (previous memory).
3. The hidden state is updated and passed to the next time step.
4. After processing the sequence, the model outputs a prediction for the next value(s).
5. Loss is calculated between prediction and actual value.
6. Weights are updated using backpropagation through time (BPTT).
7. The process is repeated for multiple epochs until the model learns temporal patterns.

Conclusion

Recurrent Neural Networks, particularly LSTMs and GRUs, are powerful tools for time series forecasting. They capture temporal dependencies and outperform traditional models in handling complex patterns. While computationally demanding, they are widely applied in stock market analysis, weather forecasting, and other domains requiring sequence prediction.