Final Exam

ECE 2795 - AI and ML Application in Power Systems

December 2024

Time Series Prediction Using Neural Networks

Problem Description:

You are provided with a time series dataset containing two signals: Ot and Rt. The data has a resolution of 15-minute intervals and spans a period of 11 months. Your task is to develop and compare predictive models for both signals (Ot and Rt) using four different neural network architectures: **RNN**, **LSTM**, **GRU**, and **Transformers**.

Tasks:

Data Preparation:

1. Preprocess the time series data as necessary to prepare it for input into neural networks.
2. The dataset file name is `timeseries_data_2024.csv`.
3. Split the data into training (80%) and testing (20%) sets.

Model Development and Training:

1. Implement and train the following neural network architectures:
   - Recurrent Neural Networks (RNN)
   - Long Short-Term Memory Networks (LSTM)
   - Gated Recurrent Units (GRU)
   - Transformer models
2. Use the Ot and Rt signals as targets for prediction.

Model Evaluation:

1. Evaluate the performance of each model on both the training and testing datasets.

2. Compute evaluation metrics for the predictions, such as Mean Squared Error (MSE), Mean Absolute Error (MAE), or any other appropriate metric.

3. Plot the training and testing loss curves for each model.

Hyperparameter Tuning:

1. Select appropriate hyperparameters (e.g., learning rate, batch size, number of epochs, sequence length) to achieve the best possible prediction accuracy for each model.

2. Document your chosen hyperparameters and provide justification for your choices.

Comparison and Discussion:

1. Compare the performance of all four models and identify which model achieves the best accuracy for predicting the Ot and Rt signals.

2. Discuss the strengths and weaknesses of each model based on your observations.

Deliverables:

1. **Code Implementation:**
   - Provide a Python implementation for preprocessing, training, and evaluation of all four models.
   - Clearly document your code and include comments to explain your methodology.
2. **Results and Plots:**
   - Include the training and testing loss curves for each model.
   - Provide a table summarizing the evaluation metrics (e.g., MSE, MAE, and R^2) for each model on the testing data.
3. **Report:**
   - Write a brief report (1–2 pages) summarizing your approach, hyperparameter choices, results, and comparison of the models.
   - Include a discussion on how the choice of model architecture impacted the prediction accuracy for the Ot and Rt signals.

Notes:

1. Students are free to experiment with the number of epochs, architecture layers, and other hyperparameters to achieve the best results.

2. Use any Python-based deep learning library (e.g., TensorFlow, PyTorch) to implement the models.

Grading Rubric:

1. Data Preparation (10 points): Proper splitting and preprocessing of the data.
2. Model Implementation (40 points): Correct implementation of RNN, LSTM, GRU, and Transformers.
3. Model Evaluation (20 points): Accurate computation of metrics and meaningful plots.
4. Hyperparameter Tuning (10 points): Justification and application of hyperparameter tuning.
5. Comparison and Discussion (20 points): Insightful analysis of model performance. Create a table and compare all methods with their accuracy, error, convergence and any other appropriate indices.

Good luck!