Neural Networks and Deep Learning (2025)

Project 2:

Time Series Forecasting with NAR and NARX

Due date: 26th Ordibehesht 1404 [23:59]

Introduction

In this project, you will explore nonlinear time series forecasting using neural network models. Specifically, you will investigate two types of models:

- **NAR (Nonlinear AutoRegressive) models**, which predict future values based solely on past values of the same variable.
- **NARX (Nonlinear AutoRegressive with eXogenous inputs) models**, which incorporate additional input variables to improve forecasting accuracy.

The objective is to forecast climate-related features including **temperature**, **wind speed**, and **humidity** using the Delhi Climate dataset.

The work is divided into three main tasks:

Task 1: Forecasting with NAR Model

Use NAR models to predict the next time step for the following variables:

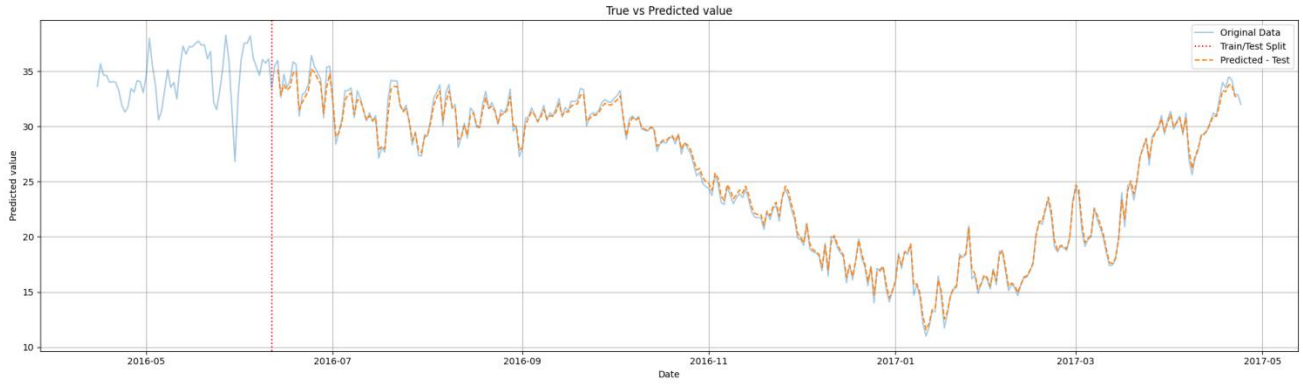
- Temperature
- Wind Speed
- Humidity

Implement the model with and without normalization.

Expected Deliverables:

- Loss curve vs. epochs
- Provide visual comparison of performance.
- Error metrics (e.g., MSE) for each variable.
- Comparative analysis of normalized vs. non-normalized results.

Here's an example of how your final results plot should look:



Task 2: Forecasting with NARX Model

Use a NARX model to predict future temperature values. You will use humidity as an external input (exogenous variable) in your model.

Expected Deliverables:

- Loss curve vs. epochs
- Provide visual comparison of performance.
- Error metrics (e.g., MSE) for each variable.
- Comparative analysis of normalized vs. non-normalized results.
- Comparison of NAR and NARX
    - Use the same dataset split.
    - Discuss which model performed better and why.

Task 3: Effect of Input Size (Lag Length)

Investigate how the number of lag steps affects model performance. Run experiments by varying lag sizes (e.g., 2, 4, 6, 8, 10). Use the NAR model structure from Task1. Analyze how increasing history length improves or harms accuracy.

Expected Deliverables:

- Plot: Input Size vs. MSE
- Discussion on optimal input size and overfitting risks

General Notes

- Use TensorFlow/Keras or pytorch for implementation if you are using python.
- You are free to choose the number of hidden layers and the activation functions for your models, as long as the architecture is clearly explained and justified.
- Provide clean plots with proper labels and legends.

Notes:
▪ Allowed programming languages: **Python**, **MATLAB**
▪ Any sign of cheating would result in a **zero** grade for this assignment.
▪ You should upload your submissions at: Link
All of the files should be in a ZIP file named in this format:
"FirstNameFamilyName-SudentNumber.zip"
Ex: "AmirZamani-4033040.zip"
▪ Your reports should be in a PDF file including: key points of your implementation, explanation of your chosen approach, reports of your final results and answers of assignment questions (if given)