MADS – “Statec Workshop” (Vasja S.)

Homework IV – NN model

In this homework, you will forecast the time series using a neural network (NN) model. You may use alternative packages, commands, or software. As before, submit the homework in HTML, Word, PDF, or Markdown format. Please also attach the script and data used for the analysis.

The deadline for Homework IV is January 15th.

Name submitted files as follows:

HWIII_analysis_[yourLastName].ipnyb/doc/pdf – In this file you describe the analysis

HWIII_data_[yourLastName].* - this file should contain the data

HWIII_script_[yourLastName].* - this file should contain the code which produces results

If you are using Markdown or Jupyter notebook, the file HWII_script_[yourLastName].* is not required.

While writing the analysis file, please use the heading tags used below.

Please perform the following steps:

1. [NN forecast] Forecast the series you selected in Homework I (HWI) using a Neural Network. Apply the 80-20 sample split as done in Homework II (HWII) and Homework III (HWIII). Include as well the series from HWIII (VAR) along with their lags. The Neural Network model does not need to be perfect. A small network with one hidden layer and a small number of nodes (e.g., 2-5) will suffice. I recommend using Python packages such as Keras, TensorFlow, or PyTorch, as the code we used in class was highly fine-tuned to work for a specific case. Most Python packages handle initialization and fine-tuning automatically, which will simplify your task.
2. [Forecast comparison] Compare the forecasting performance of the ARIMA model (from HWII), the VAR model (from HWIII), and the Neural Network (NN) model (from this homework). For each of the three models, calculate the Root Mean Squared Forecast Error (RMSFE; see below) on the last 20% of the sample. Use 1-step-ahead expanding-window forecasts; *do not use multi-step-ahead forecasts*. Describe which model performs the best and comment on potential reasons why that model outperforms the others.

BONUS: Estimate a more advanced NN model. In practice, using a linear layer with a skip connection from inputs to output works well, as part of the neural network effectively "becomes" a linear model, similar to an AR model (see hewamalage2021recurrent.pdf). You can also consider RNN or LSTM, GRU, CNN...

FURTHER LITERATURE:

Time-Series Forecasting Competition:

Makridakis Competitions

https://en.wikipedia.org/wiki/Makridakis_Competitions

- hewamalage2021recurrent.pdf: A survey on the use of RNNs in time series forecasting. This survey is rich in practical advice that can help improve your forecasts.

- makridakis2023m6.pdf: Discusses the findings of the 6th Time-Series Forecasting Competition (M6, 2023), which included participants from both industry and academia. The M6 competition focused on financial forecasting. The section titled "Major Findings and Insights" provides an excellent overview of the challenges in outperforming the market. Previous competitions primarily dealt with socio-economic data.

- makridakis2020m4.pdf: Covers the M4 (2020) competition, which was the first to demonstrate that a neural network could outperform statistical methods. However, the final forecasts were a combination of statistical and machine learning methods.

RMSFE is a measure of the spread of the forecast error distribution (=y_true – y_forecasted). It is calculated as the square root of the average (squared) forecast error:

$$RMSFE = \sqrt{\frac{\sum_i \left(y_i^{true} - \hat{y}_i^{forecast}\right)^2}{N}}$$

A perfect model would produce RMSFE = 0. The larger the RMSFE the worse the model performs. RMSFE is “scale dependent”. Its value depends on the scale of y. This implies that you can only compare models (ARIMA/VAR/NN) in which variable y is on the same scale (for example, for all the 3 models y needs to be in growth rates; it cannot be that for NN y is in levels and for VAR in growth rates…).