אוניברסיטת בן-גוריון בנגב
Ben-Gurion University of the Negev

Sequential Modeling
O. Azencot
Winter 2023

Assignment 2: **Benchmark Forecasting Methods**
deadline: December 27 at 5pm

In this home assignment, you will benchmark several forecasting methods on classification and regression tasks. Submission is in pairs. Your work should include two separate files: 1) a zip file named `code.zip` containing ALL the code you used (even if you think it is not an important piece of code), excluding library code such as Numpy/pyTorch. 2) a single pdf file named `hw2.pdf` that contains your report. Your report should be a **detailed** description of your approach to solve each of the tasks, as well as several examples of the results you obtain as detailed below. You should aim for a report with a total of 5 pages. Please submit your work via Moodle. For questions, use the Slack channel, or contact me via email.

1 Data

You will work with a music dataset for classification and an exchange rate dataset for regression. The datasets are provided in Moodle next to the HW description document. See files named `JSB_Chorales.mat` and `exchange_rate.csv`, respectively. JSB Chorales [1] is a polyphonic music dataset consisting of the entire corpus of 382 four-part harmonized chorales by J. S. Bach. Each input is a sequence of elements. Each element is an 88-bit binary code that corresponds to the 88 keys on a piano, with 1 indicating a key that is pressed at a given time. Exchange [2] records the daily exchange rates of eight different countries ranging from 1990 to 2016. Some pre-processing for the datasets may be required such as normalizing the data. See examples in https://github.com/locuslab/TCN for JSB Chorales, and https://github.com/thuml/Autoformer for Exchange.

2 Sequential Forecasting

In the last home assignment, we considered the sequential forecasting task in the context of a regression problem. Now, we will explore both classification and regression settings. For this home assignment, the main difference between the two settings is in the domain of the inputs and outputs. Specifically, a sequence in the JSB Chorales classification dataset $x_{1:t} = \{x_1, \ldots, x_t\}$, where $x_j \in \{0,1\}^{88}$ for every $j = 1, \ldots, t$. Namely, every input vector is binary. In contrast, a sequence in the Exchange regression dataset $x_{1:t}$ for the univariate case is real-valued, i.e., $x_j \in \mathbb{R}$. The goal in both cases is to faithfully predict the next item in the sequence using a function f. That is, we require $x_{t+1} \approx f(x_{1:t})$ for any t. We denote by $\tilde{x}_{t+1} = f(x_{1:t})$ the forecast of some forecasting method. We aim for a low forecasting error as measured by the mean squared error

(MSE) measure, i.e.,

$$\text{MSE}(\tilde{x}_t, x_t) = |\tilde{x}_t - x_t|_2^2 = \sum_j (\tilde{x}_t[j] - x_t[j])^2 \ , \tag{1}$$

where $x_t[j]$ is the j-th entry in the vector x_t.

The function f can be modeled in several ways. In HW1 you used `RNN` modules. Here, we will extend your work and consider the models: autoregressive integrated moving average (`ARIMA`), recurrent neural networks (`RNN`), temporal convolutional networks (`TCN`), and the `TRANSFORMER`.

1. `ARIMA`, see tutorial and documentation.
2. `RNN`, see tutorial and documentation.
3. `TCN`, see tutorial and documentation.
4. `TRANSFORMER`, see tutorial and documentation.

3 Tasks

1. Download the datasets and pre-process them, following the examples in the links above.
2. Implement or use online code for the above forecasting models. Use each of the models to solve the forecasting task separately for the JSB Choarles and Exchange datasets. Your model should predict the next state x_{t+1} from its previous observations $x_{1:t}$ for every t. Compute the prediction error on the train and test sets per epoch, and produce their plot. Specifically, train the RNN model you have implemented for 100 epochs, at the end of each epoch record the prediction error on the train and test sets. Produce a single graph containing both prediction errors as a function of the epoch.
3. Use the trained models to produce two tables comparing the results of the models on JSB Chorales and Exchange.

4 Additional Comments

1. You are allowed to use neural models that are built-in in pyTorch/Tensorflow.
2. You should not attach code to the report. Instead, please attach ALL the code you used (even if you think it is not an important piece of code) to a compressed zip package.
3. A significant portion of the grade is dedicated to the neatness and descriptiveness of the report. You should make all the figures and discussions to be as clear as possible. In particular, the axes ticks and labels, as well as the legend, and etc. should be clear without zooming in. Finally, you should describe the architecture you designed and implemented for every task.
4. At the same time, the report should not be too long. Please aim for a 5 page document.

References

[1] Moray Allan and Christopher Williams. Harmonising chorales by probabilistic inference. *Advances in neural information processing systems*, 17, 2004.

[2] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 95–104, 2018.