Challenge 2: Time Series Forecasting

Dario Del Gaizo
Valeria Detomas
Chiara Fumelli

Group Epoch404

1 Overview

The second competition of A.Y. 2021/2022 is about Time Series Forecasting, which is the process of analyzing time series to make predictions.

2 Data

The input is a 7-feature tensor, containing the following labels: Sponginess, Wonder level, Crunchiness, Loudness on impact, Meme creativity, Soap slipperiness and Hype root. As intended, this is a Multivariate time series forecasting problem, meaning that each variable depends not only on its past values but also has some dependency on other variables. The length of the tensor is 68528 observations, no Null values have been identified. The model is evaluated with the RMSE metric a test set of 864 observations.

3 Model and Hyperparameters

These hyperparameters are a result of error and trial since they were the core of the performance. The hyperparameters are the following:

- Telescope: (how many samples we predicted in the future) at first the best results were achieved with the value equal to 1, meaning the model predicted one sample at the time. This was not ideal so after some trial better results were found through a trained model with the value equal to 24.

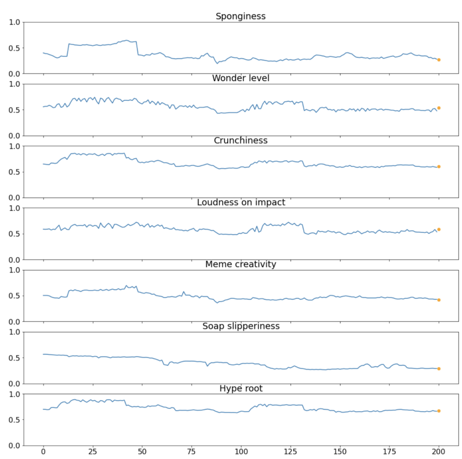

Figure 1: Telescope set to 1

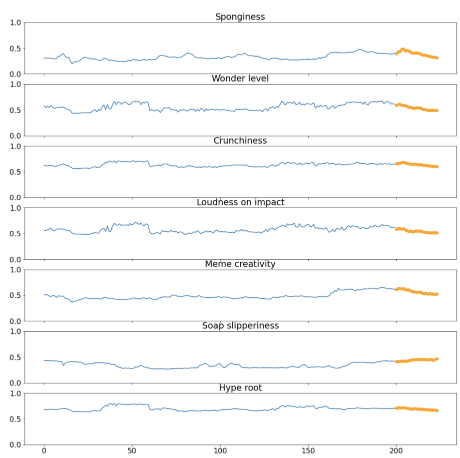

Figure 2: Telescope set to 24

- Window: the most performant value was 200, meaning the prediction will consider the previous 200 samples to make the calculations.

- Stride: the combination window-stride was crucial. The value of 10 was the best.

- Epochs: to train the model it was used 200 epochs, which were enough to reach respectable results.
- Batch size: a batch of 32 samples since it is a pretty standard value that gave good performance (with greater values, the curve was too smooth).

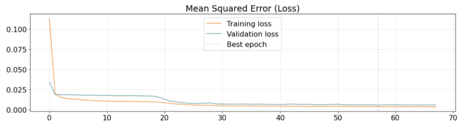

Figure 3: MSE obtained by the best model

Figure 4: MSE obtained by a poorly trained model

4 Structure

The designed model uses a bidirectional LSTMs to predict the first 24 (telescope hyperparameter) values for each class of the input. Since it is bidirectional the LSTM exploits the whole sequence of input. The LSTM is a memory cell that works as a "feature extractor" (like convolutional layers in CNN). To deal with the dynamic dataset it used a memoryless model: the autoregressive model. This means that the model predicts the next input from the previous one using "delay taps".

5 Data transformations

Even after applying the ADF tests, time series seemed to have some kind of trend or seasonality. We were unsure on how to apply such kinds of transformations, because trying these approaches gave poor results.
To try improving predictions we researched some of the properties of time series. A stationary time series is one whose properties do not depend on the time at which the series is observed. If it has trends or seasonalities, it is non-stationary. Performing the Augmented Dickey-Fuller test, the result was that all the time series are stationary. Even though the research on the argument is yet to be expanded and verified, the following definition brought light on the Multivariate time series stationarity. Some of the techniques to remove trends and seasonality are differencing, logged transformation and power transformation.
Co-integration Test
In order to determine the stationarity of a multivariate time series, we need to consider the following two cases:

- *If all the univariate time series in an MTS item are stationary, then the MTS item is stationary*
- *If some or all of the univariate time series in an MTS item are non-stationary, we need to perform the co-integration test to make sure that the MTS item is non-stationary. [1]*

[1] Kiyoung Yang, Cyrus Shahabi. "On the Stationarity of Multivariate Time Series for Correlation-Based Data Analysis".

6 ReduceLROnPlateau and Early Stopping

With a high number of epochs, the main goal is to prevent overfitting. Reduce learning rate when the model stops improving. It reduces learning rate until it reaches the value of $10^{-}6$. Patience us set at 8, so that mean that after 8 epochs with no improvement learning rate will be reduce of 0.5

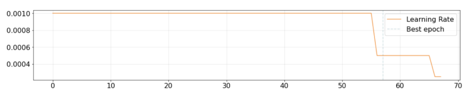

Figure 5: With ReduceLROnPlateau

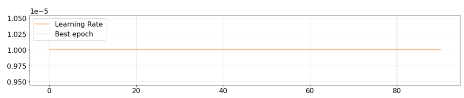

Figure 6: Without ReduceLROnPlateau

7 Testing

We decided to maintain test on our local computers to have an idea of how models were performing. The test size chosen was of 1728, which was large enough to have some graphs on the forecasting made, but small enough to not change predictions on our training dataset.

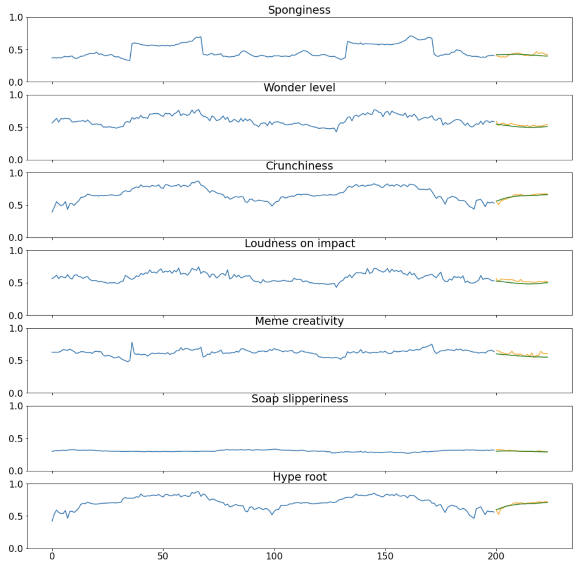

Figure 7: This is an example of the forecasting applied on time series, compared to real values in the test set for the first 24 values

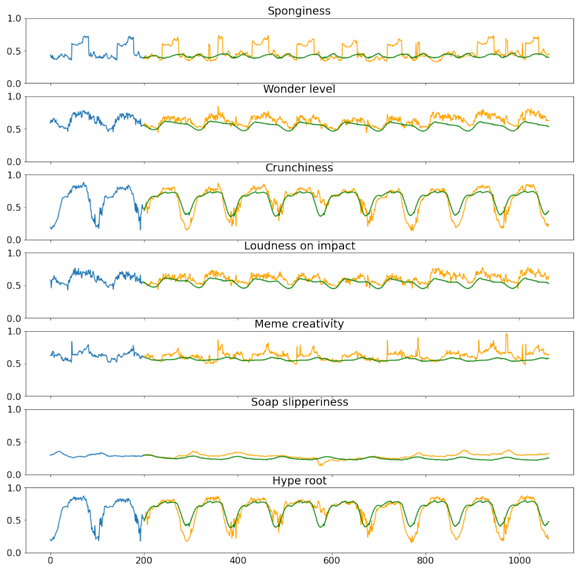

Figure 8: This is an example of locally tested autoregression