Massimiliano Arca
Gianmarco Aulicino
Alfonso Catelli
22 January 2022

Artificial Neural Networks and Deep Learning

Time-Series Forecasting Challenge

The objective of this challenge was to correctly predict the future of 7 different time-series. The base idea was to start trying some basic network with some LSTM layer and at the end have a complex structure with Attention Mechanism or a Transformer Architecture.

The variants we developed are:

- Classical LSTM with Direct Forecasting
- Classical LSTM with Autoregressive Forecasting
- Transformers Architecture

In the first two structure we also attempted to insert the Attention Mechanism, but unfortunately the result weren't good enough.

Preprocessing Data

This time, being the dataset a sequential signal, we decided to have a first look at the data.

By this analysis emerged the fact that the implementation of an exponential smoothing algorithm could give more relevance on the patterns of the signals, eliminating some values that are incoherent with the currently analysed frame of data. The smoothing helped lowering the score in the initial development phase but the best score was achieved without it.

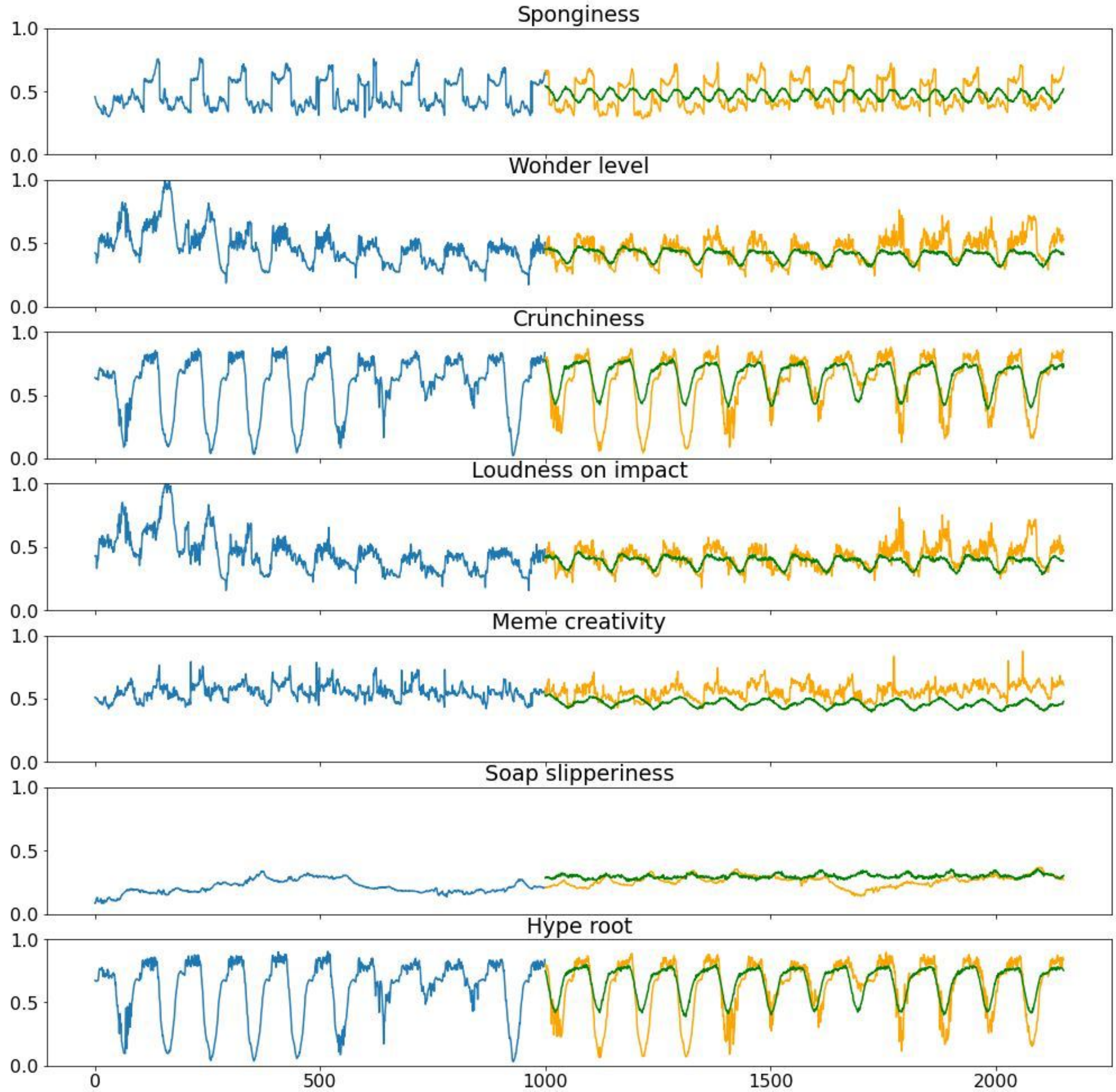
An essential preprocessing was the normalisation of the entire dataset using the min-max formula, it helped to have a really better score and a faster training. We inspect also the time-series to verify if and where there was seasonal or trend part.

Classical LSTM with Direct Forecasting

The first approach was to training the model so the desired prediction time span will be done with a single forecasting. As in the previous challenge we started from the notebook seen in the laboratory sessions with disappointing results so our first choice was to simplify the network down to only one LSTM and we got an acceptable results that was close to 5. We then changed the structure to a Bidirectional and increased the complexity by adding some layers like Convolution, LSTM, Dropout and changing the activation function from ReLU to Leaky ReLU. The results were better so the next step was to tuning the hyper-parameters and the network reached a score below 5 using a window of 4000 and stride of 20. Successively we upgraded the model by making it more complex in order to let it catch more features, but paying attention to not overfit it. What was done was to add different layers one by one, starting from a third bidirectional LSTM using as Dropout regularisation layer. With this configuration the results got worse, so we removed it and we went for the Conv1D layers. We added one at a time, followed by a BatchNorm layer, up to 4 Conv1D layers. This choice was supported by the fact that Conv layers are good at performing feature extraction. Unfortunately, even in this case the results got worse, so we decided to stick with the model with 2 bidirectional LSTMs explained above.

Classical LSTM with Autoregressive Forecasting

We tried to change the approach from direct to autoregressive but maintaining the same model, the first try with a telescope of 1 gave some of the worst result of all, the error was in the magnitude of 10, so change of the telescope was necessary. One important thought was that the evaluation was divided in two parts with different telescope of prediction, so using the greatest common factor between the two, that was 432, could be better. In the end training the best direct forecasting model and leaving the parameters unchanged got us the best result of 3.78. We fix that model and changed window and stride resulting in the best tradeoff of window = 2000 and stride = 20. We also thought that the data may be get different values between the layers of the model so we tried also to add same batch normalization layer and LayerNormalization and resulting in even worse scores (4.0, 4.1) so we went back with a simpler model.



These are the graphs of a prediction made by the best model using Autoregressive.

Transformer

Our transformer attempts were a failure. Even if this is the most powerful architecture seen in laboratory lesson, the best score still was from way simpler network. After researching on how to design our transformer from scratch and various variation of the encoder part, the model was unsuccessful, a sad 4.5 of score. We tried a normalisation layer instead of the min-max normalisation applied directly on the dataset.

Once found what seems a reasonably good structure we tuned some architecture parameters like how many encoder to have, the filter dimension and the increasing rate of the depth of the output, but the score doesn't even reached the 3.9 score. Remaining on a delusive 4.03.

Maybe we have done wrong something or the structure we used was not appropriate for the task of Time Series Forecasting.

Final Submission

In the last few days of the challenge we tried some other model as well as optimised the ones we already developed. We searched for different transformer implementations that used different layers and tried to customise a bit a transformer found on internet.

Trying to modify also the autoregressive model, after researching a few articles online and Keras documentation, we found out that it is suggested also the optimizer called RMSProp instead of Adam because it is recommended for recurrent neural network. However, we could not reach any better results, but we stayed between 3.9 and 4, hence we decided to submit the Classical LSTM with Autoregressive Forecasting model that scored 3.78.