Second Challenge ANNDL

The ReLUtants

Academic Year 2023/2024

Jacopo Piazzalunga, Gabriele Puglisi, Davide Salonico, Denis Sanduleanu

1 Introduction

The goal of this challenge is to develop forecasting models capable of predicting future samples in various uncorrelated time series. These models must efficiently utilize past observations to forecast accurately while demonstrating broad generalization capabilities across different time domains. This report emphasizes not just the design and implementation of such models, but also their adaptability and versatility, ensuring they are not constrained by single or predefined temporal contexts. Additionally, the focus will be on the evaluation methodology, specifically how the Mean Squared Error (MSE) metric is employed to assess model performance. This approach combines deep learning and time series analysis to achieve robust and adaptable forecasting solutions.

2 Dataset

2.1 Analysis

Initially, we conducted a thorough examination of the dataset. This extensive dataset comprises a total of 48,000 time series, each annotated with its associated valid period and category. The presence of zero-padding within the sequences gave us the flexibility to reconstruct the data in various ways. Time series within the same category are indicative of samples originating from the same underlying dataset. However, it's essential to note that no additional contextual information beyond these categories was available. Upon a thorough examination of the dataset, we determined that there were no discernible patterns or characteristics unique to each category. Consequently, we decided to discard the idea of developing an ensemble of specialized models, with each model focusing exclusively on a single category. This dataset characteristic implies that we must rely on the inherent patterns and relationships within the time series data itself for our analysis and modeling efforts.

2.2 Data sampling

Creating our training and test sets involved a carefully designed data sampling strategy. We systematically sampled the various time series in sequences, trying different sizes for the window, the telescope, and the stride while traversing the valid period of each sequence. In instances where the dimensions didn't align precisely, we applied padding to ensure compatibility. Our padding strategies encompassed options such as zero padding, symmetric padding, and reflect padding. During these attempts we noticed that zero padding consistently outperformed the other strategies across a wide range of models, leading us to adopt it as the preferred choice. An interesting aspect of our dataset was the initially unbalanced distribution of categories. We found that there was no need to employ upsampling or downsampling techniques because category balance or imbalance did not significantly impact the performance of our forecasting models.

3 Models Architecture

In our exploration of time series forecasting, we drew inspiration from recent papers in the field. There are lots of different implementations to address this task including Transformers, LSTM (Long

Short-Term Memory), MLP (Multi-Layer Perceptron), and CNN (Convolutional Neural Network). After conducting numerous experiments and iterations, we arrived at a strategic decision regarding the core architecture of our forecasting model. We opted for a model built with the combination of 1D Convolutions and LSTM layers. This choice was driven by careful consideration of our specific dataset. In the following sections, we will delve into the model's design, hyperparameters, and training strategies.

3.1 Selection of Backbone Network

Among the models we experimented with the best one was, as said in the previous section, a combination of 1D Convolution and LSTM. Current literature widely supports the adoption of such architectures for time series forecasting. Their inherent capability to refeed themselves with their own output aligns seamlessly with the requirements of sequence prediction tasks. The high-level architecture of our chosen model features two layers of bidirectional LSTM separated by dropout layers to enhance generalization. This is followed by two dense layers, completing the model's design.

The model is used in an autoregressive fashion: it doesn't predict immediatly the whole sequence but it predicts only a few values, refeed itself with a sequence that includes the newly predicted one and keeps going until the required number of values is predicted

3.2 Solutions implemented in our model

In our model development, we introduced several strategic enhancements to optimize performance and robustness:

- **Data Split Ratio**: We carefully considered the data split ratio, ultimately settling on a split of 0.68 for training, 0.12 for validation, and 0.2 for testing. This choice struck the ideal balance between result accuracy
- **Early Stopping**: The implementation of Early Stopping had an enormous impact on preventing overfitting and saving time during training phase
- **Dropout Layers**: The inclusion of dropout helped prevent the model from relying too heavily on specific neurons, promoting a more robust and generalized network.
- **Optimization with AdamW**: Across different model architectures, our experiments consistently demonstrated that AdamW consistently outperformed other widely used optimizers such as standard Adam, SGD, and RMSprop.
- **Fine-Tuning with Dynamic Learning Rate**: For the fine tuning phase we decided to adopt a low learning rate with a callback that reduces learning rate when a metric has stopped shrinking, in our case the validation loss.

4 Experiments

During this challenge various techniques were experimented:

- Tranformer based approaches didn't bring us evident advantages with respect to RNN or even simpler models
- Based on [ULM] we tried to implement a SLP: a simple feed forward neural network with a sine activation function which takes as inputs the embedding of the sequences using Time2Vector. Differently from the model cited in the paper we didn't use Additive Time2Vector but the performances are meant to be similar. An another attempt was made using a simple Multilayer Perceptron but even though local scores on the test set were encouraging, on the private test set they were not matched. Our hypothesis is that we made some mistake in the sampling procedure so that it doesn't have the same characteristics of the private one. A particular fact we noticed is that trough hyperparameter tuning we discovered that a lower latent dimension in T2V embedding led to better results

- Changing window size have improved by far local loss during validation and test but it generated a problem in the submission phase since the test is performed on sequences of length 200 and every attempt to adapt our models with different windows turned out to be a failure
- Normalizing data using a RobustScaler and identifying those time series that were scaled to different intervals as outliers led to their removal from the training set. This approach significantly enhanced the performance of many of our models, showing over a 30% improvement on our local test set. Notably, the local test set did not have any time series removed to maintain the fairness of the test. However, once submitted, these models performed poorly on the private test set, even worse than the same models trained on the standard training set. This underperformance could be attributed to the fact that no time series in the private test set were actual outliers.
- Data augmentation randomly performed with jittering, scaling, shifting and time warping haven't helped in generalizing the model since it was way less efficient.
- Considering the substantial training time required for our model, automatic hyperparameter optimization on this model wasn't pursued, while for lighter modes this practice was conducted using the Optuna framework, a powerful tool for automating the search for optimal hyperparameters.

5 Results

1 shows the values of MSE and MAE during the training phase. In 2 we can see the results obtained on our notebook making inference with the test set.

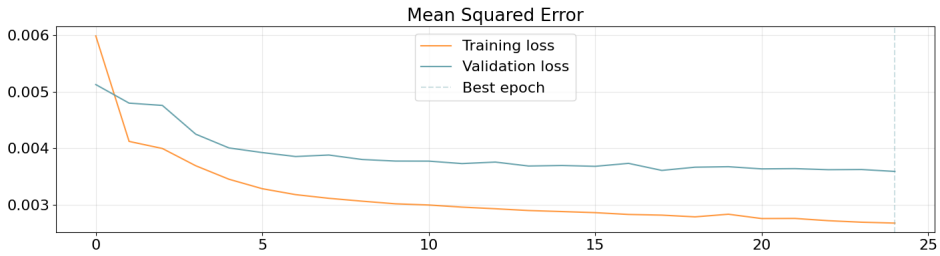

Figure 1: Model loss in the first phase during the fine tuning operation

MSE	MAE
0.00902192	0.06223073

Table 1: Results on our test set with a telescope of 18

MSE	MAE
0.01008289	0.07199894

Table 2: Results on CodaLab private test set with a telescope of 18

6 Contributions

In our collaborative project, each team member played an integral role, contributing their expertise to create a well-rounded and successful endeavor. Gabriele dedicated substantial effort to conducting a thorough literature review. His deep dive into various research papers provided us with a comprehensive understanding of state-of-the-art models, which significantly guided our model selection process. Jacopo developed the main pipeline of the whole project with a first inspection on the data. He also experimented with architectures such as Transformers and Seq2Seq. Davide invested his time in experimenting with simpler models such as SLP, Time2Vec and different kind of normalization over the

data. When feasible he tried to hypertune the parameters. Denis emerged as a pivotal contributor in the domain of data sampling. His valuable insights and support were instrumental in enhancing the diversity of our dataset, a critical aspect of achieving robust and adaptable forecasting models.

References

[KGE+] Seyed Mehran Kazemi, Rishab Goel, Sepehr Eghbali, Janahan Ramanan, Jaspreet Sahota, Sanjay Thakur, Stella Wu, Cathal Smyth, Pascal Poupart, and Marcus Brubaker. Time2vec: Learning a vector representation of time.

[Pat] Puja P. Pathak. https://medium.com/analytics-vidhya/time-series-forecasting-a-complete-guide-d963142da33fr.

[Pei] Marco Peixeiro. https://towardsdatascience.com/the-complete-guide-to-time-series-analysis-and-forecasting-70d476bfe775.

[SRR] Eli Simhayev, Kahif Rasul, and Niels Rogge. https://huggingface.co/blog/autoformer.

[TPU] https://www.kaggle.com/docs/tpu.

[ULM] Riccardo Ughi, Eugenio Lomurno, and Matteo Matteucci. Two steps forward and one behind: Rethinking time series forecasting with deep learning.

[ZCZX] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting?