Long Short Term Memory Networks for Anomaly Detection in Time Series

Pankaj Malhotra, Lovekesh Vig, Gautam Shroff, Puneet Agarwal

Experiments and Code Extensions
Course: Advanced Machine Learning for Anomaly Detection
Student: Fabian Hildebrandt
Matriculation No.: 23392857
E-Mail: fabian.hildebrandt@fau.de

I. Introduction

The approach Long short term memory networks for anomaly detection in time series (LSTM-AD) [1] has a high relevance in the domain of time-series anomaly detection (TSAD). The original paper suggests to use the Resilient Backpropagation (Rprop) method to optimize the model weights. Rprop is an algorithm, that adjusts each parameter independently based on the sign of the gradient. The step size for each individual weight increases, if the sign of the gradient remains constant and it decreases if the gradient sign changes. However, many implementations of LSTM-ADs use the Adam optimizer to update the model weights during training. This report demonstrates a comparison between both methods as well as a hyperparameter tuning evaluation. The Rprop algorithm demonstrates improved performance on the *TimeSeriesBench* [2] benchmark compared to the implementation using the Adam optimizer.

II. Used Code and Datasets

The original paper, “Long Short Term Memory Networks for Anomaly Detection in Time Series,” presents an innovative approach with significant relevance to the field of time series anomaly detection. I didn’t use the code or the datasets of the original paper, due to the lack of implementation details and low quality datasets (old and unlabeled data coming from a different paper). Instead, I used *TimeSeriesBench* [2], a comprehensive benchmark platform, which offers well-documented code, an evaluation infrastructure and access to high-quality datasets. The code and datasets are publicly available on a GitHub repository [3] with a GNU General Public License (GPL) Version 3. The GPL v3 license allows to use, study, and modify the contents. When redistributing the contents, it is required to provide it with the same license and give attribution to the original authors. I use the `PyTorch` implementation of the LSTM-AD and the *AIOPS* dataset [4], which is a labeled dataset containing training, validation and test sequences for anomaly detection.

III. Experiments

The code is adapted to train the LSTM-AD with different hyperparameters. In the experiments, I implement and compare various LSTMs trained using the Adam optimizer and Resilient Backpropagation optimizer, a different number of hidden dimensions of the cell states and hidden states (10-50), shallow versus deeper stacked LSTMs (1-5 layers) and different numbers of predicted timesteps (1-5). I provide a metric that divides the obtained F1 score in percent by the training time in seconds to get a good estimate of performance per compute.

IV. Results

The networks trained with the Resilient Backpropagation algorithm demonstrate consistently better performance and shorter training times than the networks trained with Adam optimizer. The smaller networks trained with Rprop with only one LSTM layer (1st place) or only 10/ 20 hidden dimensions (2nd/ 3rd place) show the best performance per compute. This suggests that the specific task could be solved with a relatively simple network and does not require a deep network with a large latent space. The full results are shown in table I.

TABLE I
Comparison of different Hyperparameter settings of LSTM-ADs using Adam optimizer (left) and Rprop (right)

F1	Training [s]	Score [% / s]	# hidden	# layers	Prediction length
0.89 0.92	19.27 7.79	4.62 11.81	10	2	3
0.85 0.87	36.14 9.54	2.35 9.12	20	2	3
0.79 0.83	34.76 13.44	2.27 6.18	30	2	3
0.85 0.94	35.6 13.76	2.39 6.83	40	2	3
0.79 0.95	39.4 11.99	2.01 7.92	50	2	3
0.84 0.91	18.24 6.5	4.61 14.0	30	1	3
0.65 0.93	35.33 12.58	1.84 7.39	30	2	3
0.72 0.93	48.46 14.82	1.49 6.28	30	3	3
0.85 0.9	69.33 24.54	1.23 3.67	30	4	3
0.67 0.93	55.66 30.72	1.2 3.03	30	5	3
0.87 0.94	42.14 12.37	2.06 7.6	30	2	1
0.87 0.93	40.23 12.56	2.16 7.4	30	2	2
0.67 0.88	35.73 15.21	1.88 5.79	30	2	3
0.71 0.87	38.79 12.88	1.83 6.75	30	2	4
0.7 0.89	39.21 13.0	1.79 6.85	30	2	5

Further investigations could focus on a empirical comparison of the two optimizers using more datasets and a larger number of hyperparameter settings.

A. Reproduce the results

The adapted code and the results are available on the following GitHub repository [5]. Feel free to carry out your own experiments.

References

[1] P. Malhotra, L. Vig, G. Shroff, P. Agarwal *et al.*, "Long short term memory networks for anomaly detection in time series." in *Esann*, vol. 2015, 2015, p. 89.

[2] H. Si, C. Pei, H. Cui, J. Yang, Y. Sun, S. Zhang, J. Li, H. Zhang, J. Han, D. Pei, J. Li, and G. Xie, "Timeseriesbench: An industrial-grade benchmark for time series anomaly detection models," 2024.

[3] ——, "Easytsad: A framework for easy running and evaluating your tsad algorithm," https://github.com/CSTCloudOps/EasyTSAD, 2023.

[4] A. Competition, "KPI Dataset," https://github.com/NetManAIOps/KPI-Anomaly-Detection, 2018.

[5] F. Hildebrandt, "Lstm-ads: Comparing different implementations of long short term memory networks for anomaly detection in time series," https://github.com/FabianHildebrandt/LSTM-ADs, 2024.