Tracking Signal Usage for Time Series Forecast Monitoring in an AWS MLOps Environment

Gerrit Derk Scheppat

Darmstadt University Of Applied Sciences - Faculty of Mathematics and Natural Sciences & Informatics

Supervision by Prof. Dr. Christoph Busch

Gerrit.Scheppat@Gmail.com

h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES

Abstract

Machine Learning Operations (MLOps), the task of coordinating machine learning projects with multiple models and team members, is growing in importance and interest. Cloud computing resources are a popular option in this scenario due to many reasons like easily accessible computing resources, billing by usage time and further available services like a fully managed environment. Two approaches to monitor models in an MLOps environment are compared by using two popular statistical time series forecasting models and two datasets from a widely known forecasting competition. One approach is the default Amazon Web Services (AWS) model drift monitoring and the other is a tracking signal monitoring. The goal is to reduce economical and ecological costs generated by retraining deployed models with more recent data in a cloud environment. The tracking signal monitoring is shown to serve as a more generic approach which can reduce costs when a decreased model performance is accepted for lower training costs. The AWS monitoring with in-sample error metrics as monitoring threshold used as retraining trigger shows a better performance at a comparable level of retraining counts.

Introduction

With progress in big data and deep learning, the usage of Artificial Intelligence (AI) and Machine Learning (ML) is present in various fields [3]. The increasing usage of related models often requires life cycle management to support the organization between the different roles involved in producing, hosting and maintaining models which includes data scientists, data engineers, software architects and Development & Operations (DevOps) specialists. Besides the amount of different roles involved, the typically required large datasets for model training bring additional challenges with data storage and management [1]. These challenges lead to the interest in Machine Learning Operations (MLOps) architectures that provide functions to organize the workflow connected to the model life cycle. There are several solutions available providing MLOps services and as one of the major actors in cloud computing, Amazon Web Services (AWS) provides MLOps tools as well. Especially time series forecast models need to be monitored after deployment and retrained with new data since prediction accuracy can quickly degrade when the underlying patterns that the original model was trained on, change [4]. Since cloud computing is highly demanded, the way of how MLOps is integrated in cloud computing environments need to be understood and optimized. This research has the goal to evaluate if a time series forecasting model drift monitoring performs in an MLOps scenario performs better with a tracking signal based monitoring then with the default AWS monitoring. Economical and ecological cost are evaluated with two models, two datasets and multiple monitoring approaches.

Main Objectives

1. Evaluate the cost-benefit relation of the tracking signal monitoring and the resulting retrainings.
2. Compare the results of the tracking signal monitoring with a model retraining at every new observation, no retraining and the AWS default model drift monitoring.
3. Find an adapted tracking signal threshold that results in the same retraining count as the AWS monitoring for a direct performance comparison.
4. Evaluate economical and ecological costs for all approaches.
5. Assess if a given tracking signal threshold results in comparable retaining count across multiple models and datasets to qualify it as a more generic approach in an MLOps environment.

Materials and Methods

As time series forecasting models, the established models *ARIMA* [2] and *ETS* [5] are used. Two datasets from the M4 forecasting competition [6] are used: daily and hourly. 10 time series are sampled each and a framework is build to trigger retrainings of the models when the performance drops below a given threshold. Costs and expected CO_2 emissions are evaluated on the basis of m5.xlarge AWS EC2 instances. The costs of degraded model performance is calculated with a given cost factor of 0.01€ per error value per period.

Error and Monitoring Metrics

To calculate the forecasting error and compare the model performance multiple metrics are used:

Mean Absolute Error (MAE)

$$\mathsf{MAE} = \frac{1}{n}\sum_{t=1}^{n} |e_t(1)|, \tag{1}$$

Mean Squared Error (MSE)

$$\mathsf{MSE} = \frac{1}{n}\sum_{t=1}^{n} [e_t(1)]^2 \tag{2}$$

and a tracking signal (TS) [7]

$$TS = |\frac{\alpha_1 e_t + (1-\alpha_1)E_{t-1}}{\alpha_2|e_t| + (1-\alpha_2)MAE_{t-1}}| \tag{3}$$

where $e_t(1)$ is the one-step-ahead forecast error, α_1 and α_2 are smoothing parameters, E_t is a smoothed error sum and MAE_t is a smoothed variant of the MAE. The *Root Mean Squared Error (RMSE)* is used as well.

Results

With the given parameters, the always retrain method results in the lowest total costs but due to the high retraining count, the training costs as well as the environmental costs are relatively high. The time required for a retraining with the given models is shorter then the required startup time for the EC2 instances before the actual computing can begin. The tracking signal monitoring with the set threshold value of 1.2 can reduce the retraining count but higher costs due to a reduced model forecasting performance result as well. If this benefit of a reduced retraining count and the resulting reduced training costs is more important then the degraded model performance highly depends on the usecase. With the given cost factor, the tracking signal usage can not be recommended. The AWS monitoring results is less retrainings then expected. The same tracking signal threshold results in a comparable level of retrainings over both models and datasets. An adapted tracking signal threshold of 0.4 is found to generate the same retraining count as with the AWS monitoring but in a direct comparison, the tracking signal performs worse. The ETS model shows much higher errors then the ARIMA model with the daily dataset. The environmental impact and costs stay on a low level.

	N Retrain	MAE	MSE	RMSE	Cost Training	Cost Error	Total Costs	Co2 [g]	Co2 [€]
In-Sample ARIMA		63.29	17679.67	109.31					
In-Sample ETS		63.74	17413.10	107.88					
Never Retrain ARIMA	0	551.98	979591.19	619.62	0.00	551.98	551.98	0.00	0.0000
Never Retrain ETS	0	539.35	949778.68	603.80	0.00	539.35	539.35	0.00	0.0000
Always Retrain ARIMA	100	76.56	13129.23	99.42	1.95	76.56	78.51	**118.9**	0.0066
Always Retrain ETS	100	75.84	13074.55	99.28	1.82	75.84	**77.66**	111.0	0.0061
AWS Retrain ARIMA	59.4	81.82	14717.50	106.24	1.16	81.82	82.98	70.7	0.0039
AWS Retrain ETS	58.1	80.71	14482.44	105.24	1.06	80.71	81.77	64.5	0.0036
TS ARIMA	23.9	122.89	34424.48	157.53	0.47	122.89	123.36	28.4	0.0016
TS ETS	24.4	117.17	31575.43	151.07	0.44	117.17	117.61	27.1	0.0015
TS Adapted ARIMA	62.5	92.22	19169.75	120.57	1.22	92.22	93.44	74.3	0.0041
TS Adapted ETS	60.7	93.05	21122.28	124.24	1.11	93.05	94.16	67.4	0.0037

Figure 1: Results of 10 Daily Time Series, all Costs in €, TS = Tracking Signal, Total Cost = Cost Training + Cost Error

	N Retrain	MAE	MSE	RMSE	Cost Training	Cost Error	Total Costs	Co2 [g]	Co2 [€]
In-Sample ARIMA		156.21	335764.60	211.45					
In-Sample ETS		148.66	359972.28	222.66					
Never Retrain ARIMA	0	1060.16	14947306.75	1321.18	0.00	1060.16	1060.16	0.00	0.0000
Never Retrain ETS	0	1944.71	42413385.75	2247.71	0.00	1944.71	1944.71	0.00	0.0000
Always Retrain ARIMA	100	186.52	432435.96	235.94	1.90	186.52	188.42	**115.6**	0.0064
Always Retrain ETS	100	163.73	331660.73	214.54	1.82	163.73	**165.55**	111.0	0.0061
AWS Retrain ARIMA	60.3	210.15	544797.63	265.39	1.15	210.15	211.30	69.6	0.0038
AWS Retrain ETS	54.7	218.43	670828.37	295.75	1.00	218.43	219.43	60.7	0.0033
TS ARIMA	29.8	357.74	2123711.33	507.48	0.57	357.74	358.30	34.4	0.0019
TS ETS	29.4	805.20	11815265.78	1161.47	0.54	805.20	805.74	32.6	0.0018
TS Adapted ARIMA	65.7	209.66	562254.38	274.73	1.25	209.66	210.91	76.0	0.0042
TS Adapted ETS	60.0	333.53	2698491.76	564.64	1.09	333.53	334.62	66.6	0.0037

Figure 2: Results of 10 Hourly Time Series, all Costs in €, TS = Tracking Signal, Total Cost = Cost Training + Cost Error

Conclusions

- The cost-benefit relation of the tracking signal can not surpass the AWS monitoring method with the given parameters. In a scenario with higher training costs or reduced costs for a degraded model performance, the tracking signal might become the desired solution.
- The tracking signal monitoring has a generic character that fits well in an MLOps scenario.
- More complex ML models like neural networks might favour the tracking signal monitoring due to the expected increased training costs.
- To further reduce training costs, an optimization of the EC2 startup time and instance usage can be recommended.

Forthcoming Research

The impact of different drift monitoring approaches without required ground truth data in combination with the described monitoring is to be explored. Different tracking signals or parameters can be evaluated with different data and threshold values to further assess the described findings.

References

[1] Pulkit Agrawal, Rajat Arya, Aanchal Bindal, Sandeep Bhatia, Anupriya Gagneja, Joseph Godlewski, Yucheng Low, Timothy Muss, Mudit Manu Paliwal, Sethu Raman, Vishrut Shah, Bochao Shen, Laura Sugden, Kaiyu Zhao, and Ming-Chuan Wu. Data platform for machine learning. In *Proceedings of the 2019 International Conference on Management of Data*, SIGMOD '19, pages 1803–1816, New York, NY, USA, 2019. Association for Computing Machinery.

[2] G. E. P. Box and G. M. Jenkins. *Time series analysis: Forecasting and control*. Holden Day, San Francisco, USA, 1970.

[3] Min Chen, Shiwen Mao, and Yunhao Liu. Big data: A survey. *Mobile Networks and Applications*, 19(2):171–209, 2014.

[4] Yan Deng, Majid Jaraiedi, and Wafik H Iskander. Tracking signal test to monitor an intelligent time series forecasting model. In *Intelligent Manufacturing*, volume 5263, pages 149–160. SPIE, 2004.

[5] Garima Jain and Bhawna Mallick. A study of time series models arima and ets, 2017.

[6] Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. The m4 competition: Results, findings, conclusion and way forward. *International Journal of Forecasting*, 34(4):802–808, 2018.

[7] John O. McClain. Dominant tracking signals. *International Journal of Forecasting*, 4(4):563–572, 1988.