SuBiTO: Synopsis-based Training Optimization for Continuous Real-Time Neural Learning over Big Streaming Data

Errikos Streviniotis[1,2], **George Klioumis**[1,2], **Nikos Giatrakos**[1]

[1]Technical University of Crete
[2]Athena Research Center
estreviniotis@tuc.gr, gklioumis@tuc.gr, ngiatrakos@tuc.gr

Abstract

In machine learning applications over Big streaming Data, Neural Networks (NNs) are continuously and rapidly trained over voluminous data arriving at high speeds. As soon as a new version of the NN becomes available, it gets deployed for prediction purposes (e.g. classification). The real-time character of such applications greatly depends on the volume and velocity of the data streams, as well as the NN complexity. Training on large volume of ingested streams or using complex NNs, potentially increases accuracy, but may compromise the real-time character of those applications. In this work, we present SuBiTO, a framework that automatically and continuously learns the training time vs accuracy trade-offs as new data stream in and fine tunes: (i) the number, size and type of NN layers; (ii) the size of the ingested data via stream synopses specific parameters; and (iii) the number of training epochs. Finally, SuBiTO suggests optimal sets of such parameters and detects concept drifts, enabling the human operator adapt these parameters on-the-fly, at runtime.

Introduction

In streaming settings, such as real-time social media content moderation/filtering and detection of harmful images on live platforms, AI systems face significant time pressure to make rapid decisions. To accomplish these tasks, it is crucial to use NNs that can be trained swiftly, while maintaining high accuracy. Moreover, to achieve these goals, these applications should update and adapt their models quickly to the highly volatile statistical properties of the ever-evolving input. These characteristics of the training setup necessitates delicate balance between training speed and model accuracy during the continuous training process.

Several sampling-based methods have been proposed for speeding up the training time of NNs, by approximating the matrix products. These techniques fall under two categories: (i) sampling a subset of activated neurons for every hidden layer at every epoch; and (ii) sampling a subset of neurons from the previous layer to approximate the current layer's activations, using the edges of the sampled neurons (Zhong et al. 2023; Wang et al. 2018; Huang et al. 2017). None of these approaches provide a priori known accuracy vs training time trade-offs for any given, continuously trained NN.

To fill this gap, we propose SuBiTO a specialized framework designed for real time, online training over Big streaming Data. SuBiTO automatically inspects the training time vs accuracy trade-offs throughout the training process and fine tunes (i) the size, number and type of NN layers (ii) the size of the ingested data by utilizing stream synopses (e.g. samples, dimensionality reduction, sketches) for input load shedding (Kontaxakis et al. 2023), (iii) the number of training epochs. Finally, optimal synopses parameters, NN architectures and epoch number are suggested providing a priori known balance between training time and accuracy.

Overview of the SuBiTO Approach

To achieve its goals, SuBiTO (Subito 2024) learns the accuracy and training time under various scenarios. Given the low training latency target of the involved applications, increased input size dictates shallower, less complex NNs, potentially trained for fewer epochs. Reducing the stream synopsis size may allow training of a deeper NN for a mediocre number of epochs; but it might be more beneficial to train a less deep NN for an increased number of epochs. To balance accuracy and training time, SuBiTO utilizes Bayesian Optimization (BO) (Snoek, Larochelle, and Adams 2012) performing limited trials to extract relevant statistics.

Figure 1 illustrates the SuBiTO architecture. For now, assume we have determined all the involved parameters; the `Training Pipeline` (blue path) performs the online training process, by employing a type of synopsis (e.g. stratified sampling) on the input streams, using a specific NN architecture, trained for a number of epochs. An identical NN is deployed in the `Prediction Pipeline` of the architecture. For instance, in an image classification scenario, the `Training Pipeline` ingests labeled images from a `Training Topic` of Apache Kafka (Sax 2019), the de facto standard for data stream ingestion (Giatrakos et al. 2023). The `Prediction Pipeline` (red path) ingests unlabeled images from a Kafka `Prediction Topic` and assigns labels based on the most up-to-date NN. As soon as the `Training Pipeline` has an updated model, it directly passes it, via the `Parameters Topic` in Figure 1, to the `Prediction Pipeline`, so that predictions are drawn based on the up-to-date version. Note that, synopses are used to speed up training and make updated models timely available. Parallel predictors can accelerate the

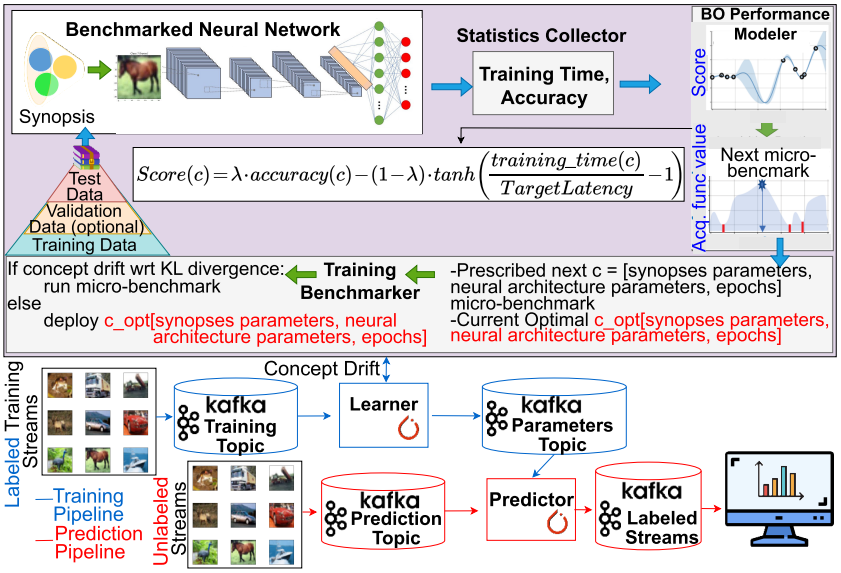

Figure 1: The SuBiTO architecture. SuBiTO open source code available at (Subito 2024).

`Prediction Pipeline`. SuBiTO does not support labeling just a sample of tuples of the `Prediction Topic`, as this would be impractical for most target applications.

When a concept drift is detected, indicating a shift in the data distribution, based on any detection function (currently, we use the KL-divergence), the execution of the `SuBiTO Optimizer` (top of Figure 1) is triggered. The Optimizer collects live input from the `Training Topic` and performs a number of trials using different NN architectures, synopses (e.g. sample size) and epoch parameters. To choose the parameter set of each trial, the `SuBiTO Optimizer` probabilistically selects one among three acquisition functions, namely lower confidence bound, expected improvement and probability of improvement (Snoek, Larochelle, and Adams 2012). After each trial, accuracy and training time statistics are collected and fed to a Gaussian Process Regressor (GPR). In BO, the GPR is trained on a limited set of trials, allowing it to predict the trade-offs for any possible parameter set. Subsequently, the Optimizer queries the GPR that provides predictions on the valid parameter sets. The most preferable such sets are those that maximize a function capturing accuracy and training time trade-off. The choice of this function is orthogonal to SuBiTO. Currently, for a suggested parameter set c, we use $Score(c) = \lambda \cdot accuracy(c) - (1-\lambda) \cdot tanh\left(\frac{training\_time(c)}{TargetLatency} - 1\right)$ (Stavropoulos et al. 2022) for $0 \leq \lambda \leq 1$. Hence, SuBiTO computes a weighted combination of both criteria, penalizing training times above a target latency. Finally, the user selects the desired parameter set which is then deployed on-the-fly in the `Training Pipeline`, while the training, prediction, and concept drift detection mechanisms function as described earlier.

User Experience and Demonstration Scenarios

(Subito 2024) presents the SuBiTO dashboard developed in (Streamlit 2024). To validate in practice that SuBiTO can accommodate any popular neural learning framework, the `SuBiTO Optimizer` is implemented in the latest version of Tensorflow, while the `Training` and `Prediction Pipelines` are implemented in PyTorch. The demonstration will use 2 real-world labeled data streams from image and video moderation analytics, namely the NSFW Detect (HuggingFace 2024) and UCF50 (Reddy and Shah 2013) datasets, for the `Training Pipeline`. The `Prediction Pipeline` will ingest truthful, unlabeled streams generated by either an a priori trained Generative Adversarial Networks or a split on the original dataset for each scenario simulating unbounded, voluminous streams at high speed.

A human operator (e.g., content moderator, community manager, etc.) can use the SuBiTO dashboard (Subito 2024) to set the valid ranges for the system's parameters, such as synopses compression ratio for load shedding, possible ranges of epochs, valid types of NN layers (CNN, LSTM, RNN, GRU, Dense), via a panel. Upon a concept drift and the execution of the `SuBiTO Optimizer`, the human operator can see the Pareto Optimal solutions and the architectures, as well as expected training loss and accuracy of the top-3 options based on the defined $Score$. The human operator can deploy any of the top-3 alternatives by clicking on them. The selected NN is on-the-fly deployed at runtime in the `Training Pipeline` along with continuously updated plots of the actual (instead of expected) accuracy, training loss and epoch duration. The statistics of the `Prediction Pipeline` are presented in a live bar-chart. The human operator can manually execute the SuBiTO at any time, bypassing automatic concept drift detection, for instance, if the `Prediction Pipeline` statistics appear unusual or deviate from the expected distribution.

Acknowledgments

This work was supported by the EU project EVENFLOW under Horizon Europe agreement No. 101070430.

References

Giatrakos, N.; Alevizos, E.; Deligiannakis, A.; Klinkenberg, R.; and Artikis, A. 2023. Proactive Streaming Analytics at Scale: A Journey from the State-of-the-art to a Production Platform. In Frommholz, I.; Hopfgartner, F.; Lee, M.; Oakes, M.; Lalmas, M.; Zhang, M.; and Santos, R. L. T., eds., *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM 2023, Birmingham, United Kingdom, October 21-25, 2023*, 5204–5207. ACM.

Huang, G.; Chen, D.; Li, T.; Wu, F.; van der Maaten, L.; and Weinberger, K. Q. 2017. Multi-Scale Dense Networks for Resource Efficient Image Classification. In *International Conference on Learning Representations*.

HuggingFace. 2024. NSFW-detect Dataset. https://huggingface.co/datasets/deepghs/nsfw_detect. [Accessed: 2024-09-25].

Kontaxakis, A.; Giatrakos, N.; Sacharidis, D.; and Deligiannakis, A. 2023. And synopses for all: A synopses data engine for extreme scale analytics-as-a-service. *Information Systems*, 116: 102221.

Reddy, K.; and Shah, M. 2013. Recognizing 50 human action categories of web videos. *Machine Vision and Applications*, 24.

Sax, M. J. 2019. Apache Kafka. In Sakr, S.; and Zomaya, A. Y., eds., *Encyclopedia of Big Data Technologies*. Springer.

Snoek, J.; Larochelle, H.; and Adams, R. P. 2012. Practical Bayesian optimization of machine learning algorithms. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'12, 2951–2959. Red Hook, NY, USA: Curran Associates Inc.

Stavropoulos, V.; Alevizos, E.; Giatrakos, N.; and Artikis, A. 2022. Optimizing complex event forecasting. In *Proceedings of the 16th ACM International Conference on Distributed and Event-Based Systems*, DEBS '22, 19–30. New York, NY, USA: Association for Computing Machinery. ISBN 9781450393089.

Streamlit. 2024. https://streamlit.io/. Accessed: 2024-09-25.

Subito. 2024. https://subito-ai-for-bigdata.github.io/.

Wang, X.; Yu, F.; Dou, Z.-Y.; Darrell, T.; and Gonzalez, J. E. 2018. SkipNet: Learning Dynamic Routing in Convolutional Networks. In *Computer Vision – ECCV 2018: 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XIII*, 420–436. Berlin, Heidelberg: Springer-Verlag. ISBN 978-3-030-01260-1.

Zhong, Y.; Zhou, J.; Li, P.; and Gong, J. 2023. Dynamically evolving deep neural networks with continuous online learning. *Information Sciences*, 646: 119411.