Aleksander Lempinen

MLOps approach for application specific performance tuning for machine learning systems

Master's Thesis in Information Technology

May 2, 2023

University of Jyväskylä

Faculty of Information Technology

Author: Aleksander Lempinen

Contact information: aleksander.lempinen@gmail.com

Supervisor: Tommi Mikkonen

Title: MLOps approach for application specific performance tuning for machine learning systems

Työn nimi: TODO samma på finska

Project: Master's Thesis

Study line: Educational Technology

Page count: 24+0

Abstract: TODO abstract

Keywords: LaTeX, gradu3, Master's Theses, Bachelor's Theses, user's guide

Suomenkielinen tiivistelmä: TODO tiivistelmä suomeksi

Avainsanat: MLOPS, TODO

Glossary

ML	Machine Learning
MLOps	Machine Learning Operations
TODO	TODO

List of Figures

Contents

1 Introduction

Machine learning (ML) systems are widely adopted and many organizations successfully have ML models running in production.

- Problem with MLOps/ML tools
  - traditional ML performance metrics such as accuracy
  - fancy features such as neural architecture search, AutoML, performance tuning
  - fancy techniques such as early stopping, grid search, bayesian optimization search etc.
  - little support for non-ML metrics: CPU util, memory used, latency, throughput (images/s etc.), hardware required (CPU, GPU, TPU etc.)
  - ML in production has many objectives besides accuracy for example: satellite image processing model A took 4-5h and model B took 5min. Model A is infeasible in production despite being more accurate.
  - "Better' depends on the specific application
- Hypothesis: Early stopping will speed up computing non-ML metrics (CPU util, Memory use, GPU/TPU requirement, latency, throughput)

TODO smaller and smaller devices, limited resources

Real-world ML systems in addition to ML performance metrics will have similar performance metrics as traditional software systems.

The aim of the study is to tune performance metrics particularly relevant to real-world ML systems. TODO Summary of thesis The main contribution of this master's thesis is using ML performance tuning techniques such as early stopping for tuning a wider range of real-world ML system performance metrics.

TODO Structure of the thesis

2 Machine Learning Operations

2.1 Machine Learning

2.1.1 Overview

Writing programs and developing algorithms to complete specific tasks is a labor intensive task requiring professional programming expertise. A different approach is to develop generic algorithms that can change behavior by learning. The field studying these types of algorithms is called machine learning. Machine learning algorithms learn by optimizing a set of parameters called a model and this optimization process is called training the model. A simplified machine learning workflow consists of splitting the data into training and test datasets, performing preprocessing separately for each dataset, training the model on the training dataset and finally evaluating the trained model on the test dataset. Machine learning workflows are explained in more detail in section 2.1.2.

Machine learning is widely used in applications like search, drug design or ad placement and can be also known as data mining or predictive analytics (Domingos October 2012). Developing machine learning systems, which are systems that are based on machine learning, can be a difficult task. Unlike traditional software development, experiments with both code and data as inputs are central to machine learning development (Zaharia et al. 2018) and reproducibility of the experiments is often problematic. While plenty of research focuses on machine learning methods or even datasets and data quality, the biggest bottleneck is human cycles (Domingos October 2012). Faster iterations improve the machine learning developer or researcher experience. An important metric to pay attention to and optimize is the mean iteration cycle for machine learning developers.

Machine learning can be practiced with two different goals in mind. First is explanatory modeling with the purpose of scientific theory building and testing and the second is predictive modeling mostly used outside of scientific research (Shmueli August 2010). One practical difference is that unlike predictive modeling, explanatory modeling rarely uses holdout test sets or cross validation for evaluation (Shmueli August 2010). Lack or presence of evalua-

tion on a test set can be used as a heuristic to quickly determine whether a machine learning project is explanatory or predictive in nature. However, even explanatory modeling benefits from evaluating the predictive power (Shmueli August 2010). Domingos (October 2012) in their paper assume all machine learning is predictive in nature and state the following:

> The fundamental goal of machine learning is to generalize beyond the examples in the training set.

It is important to keep in mind the end goals of a machine learning project, because common practices in a research setting might not be applicable when creating machine learning systems.

Machine learning algorithms can be categorized as supervised learning, unsupervised learning or reinforcement learning. The main differences are related to whether the model learns by using "right answers" provided by labeled data in supervised learning, by finding structure in the dataset in unsupervised learning or by interacting with the world in reinforcement learning. Model evaluation is also different with supervised learning mostly relying on universal cross-validation on previously unseen data and unsupervised or reinforcement learning relying on internal evaluation metrics tied to the specific algorithm Unsupervised learning has the advantage of not requiring labeled data which is an advantage for problems where labels are uncommon (Le et al. July 2012).

The amount of different ML algorithms is very large. This thesis will use commonly used supervised learning algorithms with widely available implementations to simplify the experiments in section 3.2 and focus more on the research questions.

2.1.2 Machine learning development workflow

Projects involving machine learning often make common methodological mistakes that threaten the validity of the models, but are easily avoided. Several workflows for developing machine learning systems have emerged attempting to standardize the overall development process including CRISP-DM and KDD. These workflows will overall have similar steps. Using CRISP-DM as an example we have the following steps in the workflow:

1. Business and Data Understanding
2. Data Preparation
3. Modeling
4. Evaluation
5. Deployment

Business and Data understanding refers to understanding the problem and the datasets.

Preprocessing, feature engineering

2.1.3 Hyperparameter optimization

Parameters given as part of a configuration to the machine learning model are called hyperparameters (Yang and Shami November 2020). For example learning rate, batch size or a classification threshold are hyperparameters set before the model is trained. Most common hyperparameter selection technique in practice is manually adjusting the hyperparameters and is humorously called graduate student descent.

Hyperparameter selection is a difficult task and automatic tuning of hyperparameters can help achieve state-of-the-art performance (Maclaurin, Duvenaud, and Adams April 2015). Automatic hyperparameter optimization techniques include grid search, random search, gradient based optimization and Bayesian optimization and they have different benefits and limitations (Yang and Shami November 2020).

Similar to hyperparameter optimization we have neural architecture optimization and meta modeling where model structure or modeling algorithm is treated as a tunable parameter (Baker et al. November 2017). Traditional hyperparameter tuning methods such as Bayesian optimization are unfeasible for more than 10-20 hyperparameters (Maclaurin, Duvenaud, and Adams April 2015). Performance prediction is an important step to reduce the amount of computation required for neural architecture search and hyperparameter optimization (Baker et al. November 2017).

Early stopping is a technique in which model training is halted before completion. This is done to save on computational resources and the main reason for early stopping is predicting

that the model will have poor performance Performance prediction usually relies on predicting a machine learning performance metric such as RMSE or F1 score given the current model is performing during training compared to other models

Surprisingly there is little research on using early stopping to optimize alternative performance metrics such as CPU usage, memory usage or latency. This is one of the main research questions this thesis will attempt to answer.

2.2 DevOps

2.2.1 Overview

Most commonly DevOps is said to have a focus on software quality, collaboration between development and operations, process speed and rapid feedback (Mishra and Otaiwi November 2020; Waller, Ehmke, and Hasselbring April 2015; Perera, Silva, and Perera September 2017). DevOps can be viewed from different points of view such as culture, collaboration, automation, measurements and monitoring (Mishra and Otaiwi November 2020; Waller, Ehmke, and Hasselbring April 2015). There is little consensus on the exact definition of DevOps (Smeds, Nybom, and Porres 2015). This section will describe measurement, monitoring and automation parts of DevOps in more detail.

Continuous integration, continuous deployment and continuous monitoring are well known practices in DevOps (Waller, Ehmke, and Hasselbring April 2015) describing the automatic nature of integrating, deploying and monitoring code changes. Performance profiling and monitoring are similar activities and the main difference is whether it's done during the development process or during operations respectively (Waller, Ehmke, and Hasselbring April 2015). DevOps bridges the gap between evaluating performance during the development process and during operations (Brunnert et al. August 2015).

TODO Devops lifecycle steps

TODO resource allocation/resource consumption, small memory software, benchmarking

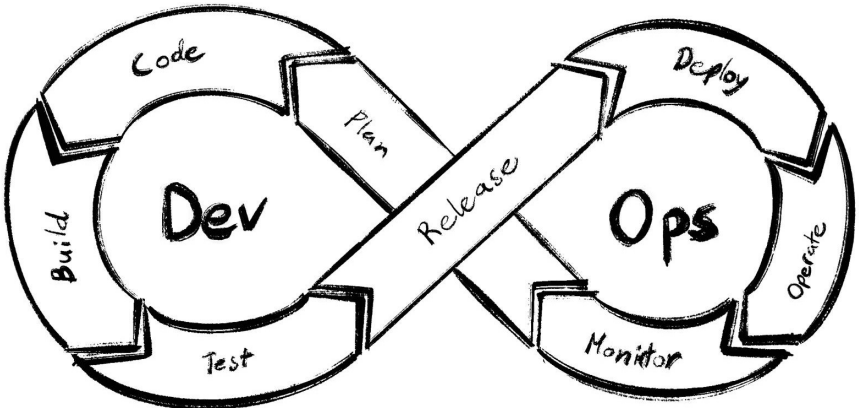

Figure 1. Example of a typical DevOps lifecycle

2.2.2 Measurement and monitoring

Performance metrics are fundamental to all activities involving performance evaluation such as profiling or monitoring (Brunnert et al. August 2015). Common metrics involve measuring the CPU, but other metrics such as memory usage, network traffic or I/O usage are not as well defined as a CPU metric (Brunnert et al. August 2015).

- Task Completion time
- Throughput
- Latency
- CPU usage
- GPU usage
- RAM usage
- VRAM usage
- I/O usage
- Network traffic

2.2.3 Continuous optimization?

Preprocessing

Training

Serving Latency

Resource demands might change depending on the inputs (Brunnert et al. August 2015) making it important to systematically measure performance not only based on code changes but also on configuration changes or even data changes.

2.3 MLOps

2.3.1 Overview

TODO (Kreuzberger, Kühl, and Hirschl May 2022) data scientists doing manual work issue from conclusions, definition of MLOps, limit to technical stuff and tooling

Requirements for a machine learning system are different depending on the task. For example speech and object recognition might have no particular performance requirements during training but strict latency and computational resource restrictions when deployed to serve large amounts users (Hinton, Vinyals, and Dean March 2015). One of the key areas of MLOps is using machine learning in production systems in addition to data processing and machine learning model training.

Performance measuring software is not new, but ML brings additional challenges in the form of models and data which requires a modified approach (Breck et al. 2017). It is also important to note, that not every data scientist or machine learning engineer working on machine learning systems has a software engineering background (Finzer 2013) and might lack the necessary knowledge to apply software engineering best practices to machine learning systems.

TODO MLOps lifecycle steps

TODO what DevOps brings to ML

TODO Continuous Training

2.3.2 AutoML

Machine learning systems in addition to machine learning performance metrics and system performance metrics will have their performance metrics tied to product or organization met-

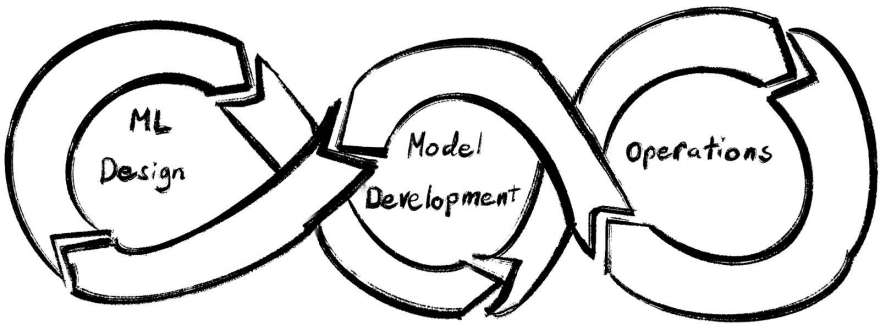

Figure 2. Example of a typical MLOps lifecycle

rics such as user churn rate or click-through rate (Shankar et al. September 2022). Choosing the right metrics to evaluate a machine learning system is important and the metrics will be different for different machine learning systems (Shankar et al. September 2022).

Automated Machine Learning (AutoML) aims to minimize human intervention in completing data analytics tasks using machine learning algorithms (Yang and Shami November 2022).

2.3.3 Performance prediction and early stopping

3 Methods

3.1 Research setup

3.1.1 Scope

The scope of the study is limited to 5 performance metrics of 3 different ML models trained and tested on 3 different datasets using a distributed computing framework Ray Tune (Liaw et al. July 2018).

TODO different models, different datasets

TODO Vertailukriteeristö: tapana ohjelmistopuolella + tapana koneoppimispuolella

TODO different resources (memory, time, accuracy)

3.1.2 Research Questions

This master's thesis asks the following research questions:

- *RQ1*: How early stopping affects performance metrics during hyperparameter optimization?
- *RQ2*: How early stopping affects performance metrics during neural architecture optimization?

3.1.3 Methodology

Methodology used is expanded from an existing methodology for machine learning experiment design (Fernandez-Lozano et al. December 2016) to include AutoML and

3.2 Experiments

3.2.1 Datasets

MNIST (Deng November 2012)

Penn Machine Learning Benchmarks (Olson et al. December 2017)

3.2.2 Machine Learning algorithms

3.2.3 Metrics

3.2.4 Training and validation

3.2.5 Inference

4 Results

TODO This is a results chapter

5 Discussion

5.1 Research Questions revisited

5.1.1 Research question RQ1

5.1.2 Research question RQ2

5.1.3 Research question RQ3

5.2 Interpretation

5.2.1 Implications for research

5.2.2 Implications for practice

5.3 Limitations

5.3.1 Datasets

5.3.2 Machine Learning algorithms

5.3.3 Metrics

5.3.4 Training and validation

5.3.5 Inference

5.4 Related Work

To find relevant related work both reverse snowballing and forward snowballing is used on a set of MLOps papers previously known to the author.

Benchmarking ML systems (Cardoso Silva et al. December 2020)

5.5 Future Work

TODO This is a discussion chapter

6 Conclusions

Summary

TODO This is a conclusions chapter

Bibliography

Baker, Bowen, Otkrist Gupta, Ramesh Raskar, and Nikhil Naik. November 2017. *Accelerating Neural Architecture Search Using Performance Prediction,* arXiv:1705.10823. Visited on January 25, 2023. https://doi.org/10.48550/arXiv.1705.10823. arXiv: arXiv:1705.10823.

Breck, Eric, Shanqing Cai, Eric Nielsen, Michael Salib, and D. Sculley. 2017. "The ML Test Score: A Rubric for ML Production Readiness and Technical Debt Reduction". In *Proceedings of IEEE Big Data.*

Brunnert, Andreas, Andre van Hoorn, Felix Willnecker, Alexandru Danciu, Wilhelm Hasselbring, Christoph Heger, Nikolas Herbst, et al. August 2015. *Performance-Oriented DevOps: A Research Agenda,* arXiv:1508.04752. Visited on January 17, 2023. https://doi.org/10.48550/arXiv.1508.04752. arXiv: arXiv:1508.04752.

Cardoso Silva, Lucas, Fernando Rezende Zagatti, Bruno Silva Sette, Lucas Nildaimon dos Santos Silva, Daniel Lucrédio, Diego Furtado Silva, and Helena de Medeiros Caseli. December 2020. "Benchmarking Machine Learning Solutions in Production". In *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA),* 626–633. https://doi.org/10.1109/ICMLA51294.2020.00104.

Deng, Li. November 2012. "The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web]". *IEEE Signal Processing Magazine* 29, number 6 (): 141–142. ISSN: 1558-0792. https://doi.org/10.1109/MSP.2012.2211477.

Domingos, Pedro. October 2012. "A Few Useful Things to Know about Machine Learning". *Communications of the ACM* 55, number 10 (): 78–87. ISSN: 0001-0782, visited on March 14, 2023. https://doi.org/10.1145/2347736.2347755.

Fernandez-Lozano, Carlos, Marcos Gestal, Cristian R. Munteanu, Julian Dorado, and Alejandro Pazos. December 2016. "A Methodology for the Design of Experiments in Computational Intelligence with Multiple Regression Models". *PeerJ* 4 (): e2721. ISSN: 2167-8359, visited on February 15, 2023. https://doi.org/10.7717/peerj.2721.

Finzer, William. 2013. “The Data Science Education Dilemma”. *Technology Innovations in Statistics Education* 7 (2). Visited on January 17, 2023. https://doi.org/10.5070/T572013891.

Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean. March 2015. *Distilling the Knowledge in a Neural Network,* arXiv:1503.02531. Visited on February 3, 2023. https://doi.org/10.48550/arXiv.1503.02531. arXiv: arXiv:1503.02531.

Kreuzberger, Dominik, Niklas Kühl, and Sebastian Hirschl. May 2022. *Machine Learning Operations (MLOps): Overview, Definition, and Architecture,* arXiv:2205.02302. Visited on December 7, 2022. https://doi.org/10.48550/arXiv.2205.02302. arXiv: arXiv:2205.02302.

Le, Quoc V., Marc'Aurelio Ranzato, Rajat Monga, Matthieu Devin, Kai Chen, Greg S. Corrado, Jeff Dean, and Andrew Y. Ng. July 2012. *Building High-Level Features Using Large Scale Unsupervised Learning,* arXiv:1112.6209. Visited on February 3, 2023. https://doi.org/10.48550/arXiv.1112.6209. arXiv: arXiv:1112.6209.

Liaw, Richard, Eric Liang, Robert Nishihara, Philipp Moritz, Joseph E. Gonzalez, and Ion Stoica. July 2018. *Tune: A Research Platform for Distributed Model Selection and Training,* arXiv:1807.05118. Visited on February 22, 2023. https://doi.org/10.48550/arXiv.1807.05118. arXiv: arXiv:1807.05118.

Maclaurin, Dougal, David Duvenaud, and Ryan P. Adams. April 2015. *Gradient-Based Hyperparameter Optimization through Reversible Learning,* arXiv:1502.03492. Visited on February 3, 2023. https://doi.org/10.48550/arXiv.1502.03492. arXiv: arXiv:1502.03492.

Mishra, Alok, and Ziadoon Otaiwi. November 2020. “DevOps and Software Quality: A Systematic Mapping”. *Computer Science Review* 38 (): 100308. ISSN: 1574-0137, visited on January 17, 2023. https://doi.org/10.1016/j.cosrev.2020.100308.

Olson, Randal S., William La Cava, Patryk Orzechowski, Ryan J. Urbanowicz, and Jason H. Moore. December 2017. “PMLB: A Large Benchmark Suite for Machine Learning Evaluation and Comparison”. *BioData Mining* 10, number 1 (): 36. ISSN: 1756-0381, visited on February 22, 2023. https://doi.org/10.1186/s13040-017-0154-4.

Perera, Pulasthi, Roshali Silva, and Indika Perera. September 2017. “Improve Software Quality through Practicing DevOps”. In *2017 Seventeenth International Conference on Advances in ICT for Emerging Regions (ICTer),* 1–6. https://doi.org/10.1109/ICTER.2017.8257807.

Shankar, Shreya, Rolando Garcia, Joseph M. Hellerstein, and Aditya G. Parameswaran. September 2022. *Operationalizing Machine Learning: An Interview Study,* arXiv:2209.09125. Visited on December 7, 2022. https://doi.org/10.48550/arXiv.2209.09125. arXiv: arXiv:2209.09125.

Shmueli, Galit. August 2010. “To Explain or to Predict?” *Statistical Science* 25, number 3 (). ISSN: 0883-4237, visited on March 14, 2023. https://doi.org/10.1214/10-STS330. arXiv: 1101.0891 [stat].

Smeds, Jens, Kristian Nybom, and Ivan Porres. 2015. “DevOps: A Definition and Perceived Adoption Impediments”. In *Agile Processes in Software Engineering and Extreme Programming,* edited by Casper Lassenius, Torgeir Dingsøyr, and Maria Paasivaara, 166–177. Lecture Notes in Business Information Processing. Cham: Springer International Publishing. ISBN: 978-3-319-18612-2. https://doi.org/10.1007/978-3-319-18612-2_14.

Waller, Jan, Nils C. Ehmke, and Wilhelm Hasselbring. April 2015. “Including Performance Benchmarks into Continuous Integration to Enable DevOps”. *ACM SIGSOFT Software Engineering Notes* 40, number 2 (): 1–4. ISSN: 0163-5948, visited on January 17, 2023. https://doi.org/10.1145/2735399.2735416.

Yang, Li, and Abdallah Shami. November 2020. “On Hyperparameter Optimization of Machine Learning Algorithms: Theory and Practice”. *Neurocomputing* 415 (): 295–316. ISSN: 09252312, visited on January 25, 2023. https://doi.org/10.1016/j.neucom.2020.07.061. arXiv: 2007.15745 [cs, stat].

———. November 2022. “IoT Data Analytics in Dynamic Environments: From An Automated Machine Learning Perspective”. *Engineering Applications of Artificial Intelligence* 116 (): 105366. ISSN: 09521976, visited on January 25, 2023. https://doi.org/10.1016/j.engappai.2022.105366. arXiv: 2209.08018 [cs, eess].

Zaharia, M., A. Chen, A. Davidson, A. Ghodsi, S. Hong, A. Konwinski, Siddharth Murching, et al. 2018. “Accelerating the Machine Learning Lifecycle with MLflow”. *IEEE Data Eng. Bull.,* visited on March 14, 2023.