Machine Learning Engineering
(and Sustainability)

Prof. Dr. Michael Granitzer
Chair of Data Science
University of Passau

contributions from Thomas Weißgerber, Christofer Fellicious, Mehdi Ben Amor, Lorenz Wendlinger

Virtual Summer School
25.06.2020, Online
What constitutes a good Machine Learning solution?

1. Machine Learning Algorithm
2. Dataset
3. Hyperparameter Tuning
4. Evaluation Metric

Often overlooked: The engineering process (or how we bring 1-4 together)

- Software Engineering
- Experimental Setup
- Data Management
Some Examples of Engineering Impact

Proper engineering, fair comparison „Engineering > Algorithm“

Engineering and testing is hard, requires lots of resources, but is nevertheless essential.

A Metric Learning Reality Check

Kevin Musgrave¹, Serge Belongie¹, Ser-Nam Lin²
¹Cornell Tech ²Facebook AI

Abstract. Deep metric learning papers from the past four years consistently claimed great advances in accuracy, often more than doubling the performance of decade-old methods. In this paper, we take a closer look at the field to see if this is actually true. We find flaws in experimental setup of these papers, and propose a new way to evaluate metric learning algorithms. Finally, we present experimental results that show that the improvements over time have been marginal at best.

(a) The trend according to papers  (b) The trend according to reality

Influence of non-deterministic GPU execution

Optimisation Parameters + Random Seeds gives ~ ±1%

Fellicious, Weißgerber, Granitzer, Effects of random seeds on the accuracy of Convolutional Neural Networks, LOD 2020
Why is ML Engineering complex?

Some Factors

- Dataset Creation (train/test/validation/development sets)
- (Big) Data Management
- Potentially infinite model choices and hyperparameter tuning
- Moving Targets / Changing Requirements / Learning on the go
- Complex Technical Infrastructure
- Demanding Skillset of Practitioners / Interdisciplinary Teams (Coding, Math, Communication, Domain Knowledge etc.)
- Reporting Results (e.g. in Papers)
Our Wish List

What we want to have…

• Logging strategies
• Robust and standardized testing
• DevOPs – Deployment, Data Management
• Don’t Repeat Yourself - Reuse existing experimental results
• Collaboration and Communication Support (beyond ipyn’s)

What we don’t want to have…

• Additional effort in
  • Coding
  • Deployment
• Setting up additional environments or tools
• Change underlying software development and business processes

Comparable to general Software Engineering?
Our Contribution - PyPads and Padre
https://github.com/padre-lab-eu/

Python Library plus Wikidata Infrastructure for non-intrusive logging and semantic analysis.

- Monkey-patching of existing frameworks for automatic (detailed) logging
- Post-hoc semantic analysis of logs for inter-library comparison and archiving
- Descriptive metadata format for community driven mappings
Conclusion

• ML Engineering is highly undervalued
  „We present a new Algorithm“ vs. „We properly engineered …“

• Comparable to Software Engineering, we need a better ML Engineering

• Tools for automating „as much as possible“

• PyPads and Padre with contributions to automated logging and semantic harmonization
Questions?