WASP Software Engineering and Cloud Computing 2024
Software Engineering Assignment

Sanna Persson
sannape@kth.se

August 19, 2024

1 Introduction

My research focuses on improving the early detection and diagnosis of neurodegenerative diseases by leveraging diffusion MRI and tractography which is a non-invasive technique used to visually represent neural tracts. Diffusion MRI infers the diffusion directions of water in the brain which occurs preferentially along the axons, and modeling the diffusion allows us to identify neural tracts (or streamlines). This technique is instrumental in understanding the complex network of connections in the brain, particularly the changes in structural brain connectivity at different disease stages. However, the field currently lacks reliable methods for structural connectomics and diagnostics from brain tractograms. We aim to address this gap to allow for more precise connectivity analysis and possible clinical applications. The methods developed during the studies will primarily be applied to data from Alzheimer's and Parkinson's patients. The following studies are planned to be completed during the project:

(I) Develop metrics for assessing redundancy in tractography methods working with pseudo-labels.

(II) Improving existing methods for tractography filtering with graph neural networks and introducing a bundle context by modeling a cohort of streamlines as a graph.

(III) Generate virtual magnetic resonance elastography images from diffusion MRI for assessment of mechanical properties of the brain without additional examinations.

(IV) Generate synthetic diffusion MRI (dMRI) images for training tractography and tractography filtering methods with diffusion models.

(V) Multi-modal model with dMRI and magnetic resonance elastography (MRE) images for improved tractography and predictive downstream diagnostic tasks.

2 Lecture takeaways and reflection

2.1 Verification and validation in medical technology

My research involves developing methods to improve the accuracy and precision of the conclusions that other researchers and practitioners are able to draw from their data. A major challenge is that the ground truth is elusive in the field of tractography. We cannot see the scale of axons; therefore, we need to verify our methods on pseudo-labels, comparing them to established methods and on synthetic data to make sure they are following the governing principles we determined. The question of validation may, however, be even more challenging. In my research, we are attempting to anchor our research with collaborators at the Karolinska Hospital. Even so, with the validation that our research is valuable, there

are many steps to implementing a method in practice. For example, in my recent project, I developed a metric to quantify redundancy, yet at this point, several hurdles would need to be overcome to make sure that the method:

- Generalizes to MRI images of different quality and intra-datasets
- Improve processing time per subject
- Validate intra-subject performance and identify edge cases

That is, the primary verification of a method in our field is often on a high-quality open dataset or synthetic dataset. In the clinic, time and money are limited, and it is common with lower-resolution images and artifacts.

2.2 Confirmation bias between technology and medicine

Confirmation bias is a significant concern in the field in light of the absence of ground truth labels. One problem that I often face is the lack of widely accepted benchmarks. Each research group tests its machine learning models on different datasets, possibly with some overlap, resulting in difficulty. Each group in the field will then advocate their own methods, validation, and conclusions, leading to a confirmation bias hindering progress. This bias is accentuated by the reluctance to sharing data and models in the medical field, partly due to privacy reasons. Any researcher who wants to compare models is then faced with the time-consuming task of re-implementing and re-training the models. The solution to this is similar to what has happened in the field of deep learning, where almost any task has rigorous and widely accepted benchmarks on which any researcher needs to report results to be considered state-of-the-art. There is an issue with confirmation bias in benchmarks as well, for example, making models biased towards certain demographics or specific data distribution, but the alternative is hard to quantify results and a field in disarray.

2.3 How do people think AI will change medicine?

During Lenberg's lecture, he addresses the AI adoption process at SAAB and how employee attitudes are a success factor. At SAAB, many people are hesitant or scared of what changes AI will bring to their working life; some are maybe concerned about their job security. In medical technology, the gates are guarded, rightfully so, by medical doctors who are ultimately responsible for the patients' well-being. The tools that they adopt should already be deemed safe and of clear utility to their work. From the patient's perspective, the uncertainty is even greater. When explaining my research, I sometimes get the reaction that AI will ruin healthcare, making it even more impersonal and difficult to receive good care. Both the patient's view and the medical doctor's are important to determine the actual effect of AI in healthcare.

2.4 Using AI tools for testing

In the guest lecture by Parthasarathy at Volvo Group, he discusses using LLMs as tools for testing software and writing test cases. He specifically emphasizes the improved efficiency of engineers when they are able to leverage AI models for mundane code writing. He also presents AI as possibly able to identify and write more test cases than is otherwise possible. In my field, the corresponding way is that many of the medical imaging tools work as an assistance to that of a trained radiologist eye. In my work, we can quantify and visualize things a radiologist may use intuition and experience to assess in an MRI image. One central argument from the proponents of AI in medicine is the potential to streamline processes and speed them up by using humans to verify decisions and assessments made by a model. The concern

is similar to that in the code writing at Volvo, which induces an over-reliance on systems that could be faulty or not work on edge cases. One can also turn the process around and suggest that AI can be used to quality-assure medical doctors' work, identifying mistakes in radiologist assessments or patient journals. We will likely see an adoption of technology from both ends in this case. In structural connectomics, this becomes abundantly clear when we are producing metrics comparing patients to control on data produced with methods (deep learning and traditional) containing significant limitations. There is an inclination for people on the clinical side to trust the generated data representation, overlooking the uncertainty, and similar behavior is likely to present in many fields.

3 CAIN papers

3.1 Investigating the Impact of Solid Design Principles on Machine Learning Code Understanding

In [1], test the hypothesis that using SOLID design principles improves code understanding among machine learning engineers. The SOLID design framework consists of five design principles that describe how to write software that is easy to comprehend, maintain, and re-use. The principles are:

- **Single Responsibility Principle (SRP)**: A class should have only one job or responsibility.
- **Open/Closed Principle (OCP)**: Software entities (classes, modules, functions) should be open for extension but closed for modification.
- **Liskov Substitution Principle (LSP)**: Objects in a program should be replaceable with instances of their subtypes without altering the correctness of that program.
- **Interface Segregation Principle (ISP)**: Many client-specific interfaces are better than one general-purpose interface.
- **Dependency Inversion Principle (DIP)**: High-level modules should not depend on low-level modules. Both should depend on abstractions. Abstractions should not depend on details. Details should depend on abstractions.

We can summarize these principles in simple terms: code should be built carefully, incurred dependencies should be observed, and functionality should be divided into logical building blocks to be combined according to the needs of clients. The authors state that in ML, these principles are sometimes hard to follow due to the experimental nature of the field. In many cases, important code bases are made up of Jupyter notebooks, ultimately creating significant technical debt. The hypothesis was that applying SOLID design principles to the industrial code base would improve engineers' understanding of the ML code. In an experiment where code was refactored according to each design principle, the authors analyzed a group of engineers' understanding of the code.

The findings of the paper were that for the LSP, ISP, and DIP, the study participants found the code significantly more understandable when it had been refactored. However, the two first SOLID principles did not obtain significant differences between the original and refactored code. Their conclusion is that there is value to the SOLID design principles and that the data science community should prioritize the implementation of similar principles for making code easier to understand and maintain.

The practice of code maintainability and understanding is an issue, in my opinion, for every researcher working on ML. A practical example, all too common, is that open-source code is not equivalent to easily usable and understandable code. In many cases, the file structure is illogical; there is no documentation or easy way to use the code because it requires a specific data structure to be used. Especially in the field of biomedical imaging, it is common to see code published as a black box in a NextFlow/Docker container or code that is barely runnable.

In my research, one of the main goals is to make software available and easily usable for other researchers to experiment with and use, thereby improving the open science climate in the field. The same holds true for implementations, such as MRI scanners, which are a common deployment option for software in my field. Following standard design principles presents a significant advantage in promoting maintainability and code understanding. By adhering to principles such as SOLID, we can ensure that the software is robust and reliable and more manageable for other researchers to comprehend, modify, and extend. This approach fosters collaboration and accelerates advancements by allowing researchers to compare their results and benchmark in an easier way.

3.2 A Combinatorial Approach to Hyperparameter Optimization

Khadka et al. [2] introduced a new algorithm using t-way testing for hyperparameter optimization. This approach leverages combinatorial testing to systematically explore the hyperparameter space, focusing on interactions among a subset of hyperparameters to reduce the computational burden. The motivation is that traditional algorithms, such as grid search or random search, are slow for larger datasets.

The introduced approach begins by identifying relevant hyperparameters and their important values for a given model, forming the Input Parameter Model (IPM). For deciding these parameters and their ranges, the authors emphasize the need for domain-specific knowledge. They then use a combinatorial testing tool to generate a set of t-way tests from the IPM, with each test representing a hyperparameter configuration. If $t = 2$, this corresponds to testing all pairs of possible combinations of values and understanding how all pair-wise interactions work. These configurations are evaluated based on performance metrics such as accuracy to select the optimal hyperparameter configuration.

Their experimental results demonstrate that this approach reduces the number of necessary model evaluations and cuts computational expenses compared to Grid Search, Random Search, in classification and regression task on various datasets while maintaining similar performance. In several of their experiments Bayesian Optimization was faster than t-way testing.

The study also explores the impact of varying the value of t in t-way testing, finding that smaller t values can speed up the optimization process without compromising model accuracy significantly. This suggests that only a few hyperparameter interactions are often sufficient to achieve good performance. The implication for AI engineering is that both small and big projects will gain from considering how they perform their hyperparameter optimization so as not to waste time or miss out on performance. It should be noted show that in many experiments, the differences in errors are not major, but for some tasks, they could be very meaningful.

A limitation not mentioned in the paper is that for larger search spaces and datasets the t-way testing may not be feasible and one may need to more selectively choose the best parameters. Overall their findings could be readily applied to my research. It introduces a structured approach to hyperparameter optimization, identifying the relevant parameters beforehand.

In the work with medical images, many data-dependent hyperparameters need fine-tuning and are often decided based on domain knowledge rather than tested empirically. One such example is down-sampling of tractography streamlines to a specific set of points from varying numbers from 20-100. In this case we used the median number points for simplicity but in reality we could have tested this as the hyperparameter that it is.

Many projects involving ML models for MRI images also include down-sampling, patching, or domain-specific hyperparameters in addition to ML hyperparameters, giving a difficult search space to optimize manually. Using a structured search that effectively examines possible interactions is an important tool to save time and resources. Often, grid search is naively implemented because it is the simplest method, even though random search is taught in even the simplest of ML courses.

Practically, the main takeaway that would improve the use of AI engineering is to test hyperparameters in a structured way and be transparent about the hyperparameter testing that has been done. There are

several tools that make this process simpler, such as Weights & Biases or Optuna, that implement both the random search and Bayesian optimization strategies for hyperparameter tuning.

References

[1] Raphael Cabral et al. "Investigating the Impact of SOLID Design Principles on Machine Learning Code Understanding". In: *Proceedings of the IEEE/ACM 3rd International Conference on AI Engineering - Software Engineering for AI*. CAIN '24. New York, NY, USA: Association for Computing Machinery, June 11, 2024, pp. 7–17. ISBN: 9798400705915. DOI: 10.1145/3644815.3644957. URL: https://dl.acm.org/doi/10.1145/3644815.3644957 (visited on 08/13/2024).

[2] Krishna Khadka et al. "A Combinatorial Approach to Hyperparameter Optimization". In: *Proceedings of the IEEE/ACM 3rd International Conference on AI Engineering - Software Engineering for AI*. CAIN '24. New York, NY, USA: Association for Computing Machinery, June 11, 2024, pp. 140–149. ISBN: 9798400705915. DOI: 10.1145/3644815.3644941. URL: https://dl.acm.org/doi/10.1145/3644815.3644941 (visited on 08/13/2024).