Assignment, WASP SE Course Module 2024

Télio Cropsal

August 29, 2024

1 My research topic

My research topic is about machine learning applied to drug discovery. Drug discovery is the process of finding new promising drugs. It is a long and expensive task, on average 12 years are needed to release a new drug to the market. This long journey can be explained by the very high number of molecules that can be evaluated, the need for a wide range of experiments and the complexity of biological reactions.

In this setting, compagnies and researchers are interested in machine learning to leverage computational resources and previously accumulated data with the goal to find statistically relevant molecules. Hence, limiting the number of experiments to be carried out.

More specifically, my research interest lies into phenotypic drug discovery, where instead of looking for the best molecule to match a known protein, practitioners are evaluating the effects of molecules on cells.

The data representing cells are pictures coming from microscopes, thus, I'm interested in applying the recent advances of multi-modal and text-to-image generative models to phenotypic drug discovery. The goal being to have a trustworthy model able to generate pictures of cells given a molecule or other relevant conditions.

The work done is indented to deliver open-source tools to foster collaboration between experimental and computational researchers, and to lower the barrier to AI-accelerated drug discovery.

2 Relevant ideas for my research from Robert's lecture

The notion of "technical debt" which was introduced during the introduction of the course is highly relevant for any machine learning projects and often neglected by researchers and data scientists. According to Wikipedia, technical debt is the "cost of future reworking required when choosing an easy but limited solution instead of a better solution that could take more time". In the machine learning research community, for the sake of reproducibility, it is common for papers to reference a repository with the code and models to run all the experiments. But most of the time, there is no long-term vision, since the code is mainly there to support a paper, hence all the classical technical debts in machine learning are

amplified. Some of these issues, as specified in the seminal paper from Sculley et al, are data dependencies, anti-patterns such as glue code and configuration debt.

I would like to highlight that these issues are becoming even more problematic with the recent trend of foundational multi-modal models. Indeed, these models usually rely on previously trained models on different modalities to perform new tasks, acting like "undeclared consumers". For instance, Stable Diffusion, one of the most used and powerful text-to-image model relies on CLIP embeddings (a previously trained model over image-text pairs) for the text conditioning. Some papers, such as universal guidance for diffusion models, are advocating for a mix of different signals from different pre-trained models to condition the generative process of diffusion models.

This problem is also present in machine learning for drug discovery. Some foundational models have been trained following LLMs architectures with molecules represented as strings. Nowadays, these models are used effectively as pre-trained baselines for new downstream tasks, at the price of more hidden technical debt. To deal with the issue of "technical debt", the course introduced the practice of "quality assurance" where standards are defined to ensure the compliance of produced software. On top of the standards, tools are developed to verify the different aspects of the project such as the code and the data. Among these tools, I discovered the existence of static analysis and linters specialized for machine learning projects such as "mllint" or "pynblint" for Python notebooks. I'm planning of using these promising tools for my own research.

3 Relevant ideas for my research from guest lectures

The guest lecture from SAAB was mainly about behavioral software engineering and machine learning in a critical environment. Even if the application domains are very different, I realized that they are a lot of similarities between drug discovery and air traffic management. Indeed, both domains are interested in automation with machine learning, but changes need to be thoroughly evaluated since it is critical applications with human lives at stake.

The view of going from software engineering to behavioral software engineering is also relevant since in practice, it is important to convince skeptical chemists and biologists of the usefulness of the new AI-driven tools. Considering end-users, stepping out of your own machine learning bubble is a good way of avoiding some pitfalls.

4 Review of "Prevalence of Code Smells in Reinforcement Learning Projects"

With the rise of reinforcement learning, the authors studied the code quality of reinforcement learning libraries. They considered the repositories with the highest number of stars on GitHub, analyzing only the implementation of the Q-learning algorithm, as well as the whole codebase of ACME, a library maintained by reinforcement learning engineers. All the

libraries are programmed with Python since it is the most used programming language in machine learning.

To evaluate the repositories, they used static analysis tools looking for code smells related to code organization, abstraction and expression.

The authors noticed several patterns from the analysis. First, as expected, the number of code smells in noticeably lower within the ACME codebase, developed by reinforcement learning engineers, than the other libraries. It is also interesting to notice that the ACME implementations are staying close to the mathematical abstraction of reinforcement learning to comply with any user-defined environment (any markovian decision process) as opposed to some libraries with a use-case in mind (e.g. finance or games such as Tetris or Flappy Bird).

Second, without a surprise, it is more likely for projects with a larger codebase to exhibit more smells.

Third, the most common smells among all the projects are showcasing the difficulties to define properly program entities, their behaviors and how they should share information. The design principle violated is the fundamental one about single responsibility. The authors concluded that currently reinforcement learning algorithms cannot be expressed at the appropriate level of abstraction because of the inherent complexity of these algorithms.

Hence, the authors advocate for the development of reinforcement learning specialized software quality tools and more expressive abstractions to increase the maintenance of reinforcement learning systems.

As mentioned in the paper, reinforcement learning is used in chemistry. The main application is to balance the exploitation-exploration tradeoff in the molecular space with desirable properties framed as rewards. It has been applied with success, with for instance REINVENT, an approach leading to the generation of new promising molecules. These methods are popular in the field and rely on reinforcement learning libraries or algorithms developed from scratch, on top of machine learning libraries. Thus, being aware of the likely presence of code smells when using previous methods or developing new ones is crucial.

In this sense, if I would have to release a project using reinforcement learning for drug discovery, following the paper guidelines, I would be cautious at separating the mathematical formalism of reinforcement learning with the practical environment, use static analysis and finally make a motivated decision about the choice of the libraries.

5 Review of "Data Smells in Public Datasets"

Drawing upon the concept of code smells, this paper motivates a similar approach with data. The authors introduced data smells, which correspond to anti-patterns in datasets that indicate early signs of problem or technical debt. To-do so, a catalogue was created with 14 different data smells by analyzing over 25 popular public datasets.

The advantages of looking for established data smells are multiple. First, machine learning systems are not only dependent on the code but also on models and data, but in practice there is a lack for data quality analysis tools. Second, a full training-testing cycle requires time and resources, hence there is a need for catching potential problems as soon as possible.

The list was defined with the objective of having generalizable data smells. The smells are divided into the following groups: redundant value, categorical value, miscellaneous value, missing value, string value. Finally, the authors noted that only checking for the proposed data smells is not enough because they are still subject to interpretation and depending on the applications, domain-specific data smells might be needed.

Hence, the authors pointed out that a good documentation is necessary to share domain knowledge. For this purpose, the documentation should include information regarding the data source and collection, changes that have been made to the dataset along the motivation behind, expected schema of columns and descriptive statistics. Considering my research, I'm working with the JUMP-CP dataset consisting of terabytes of cell images under various perturbations. Metadata files are also available to describe the experiments. The paper is useful to have a principled look at the dataset before starting to work on it. Considering the size of the dataset, finding potential data smells early could help me save a lot of time and computing resources.

The dataset is showcasing some good practices with for instance a schema of the metadata and important information distilled in the dataset paper and GitHub repositories. Unfortunately, there is no proper documentation, it could be convenient to have all the important information in one place. Moreover, the entry cost of using this dataset is quite high considering the size and the domain-knowledge needed, hence a proper documentation would also democratize the dataset to researchers less familiar with biology, or the opposite, researchers less familiar with the tools needed to leverage very large datasets.