Redefining ML for Open Source Science

COM 314 – Advanced Presentational Speaking

J. Setpal

April 5, 2024

Background

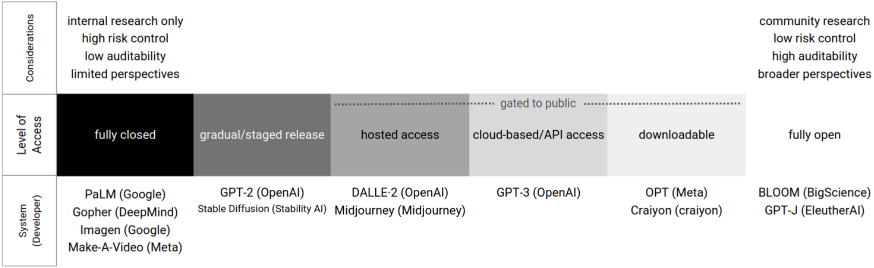
Free and Open Source Software (FOSS) in a *scientific* setting allows researchers to 'stand on the shoulders of giants'.

[1] Solaiman [2023]

Background

Free and Open Source Software (FOSS) in a *scientific* setting allows researchers to 'stand on the shoulders of giants'.

The following figure[1] presents a proposed gradient of open-source in ML:

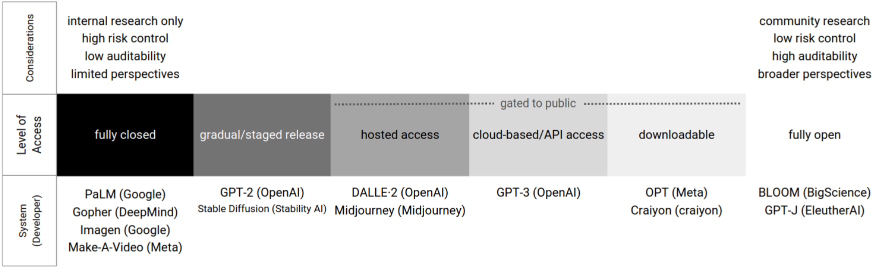


[1] Solaiman [2023]

Background

Free and Open Source Software (FOSS) in a *scientific* setting allows researchers to 'stand on the shoulders of giants'.

The following figure[1] presents a proposed gradient of open-source in ML:



This definition is <u>impractical</u> for machine learning projects.

[1] Solaiman [2023]

Death to Jupyter Notebooks

Jupyter Notebooks are *fantastic* for experimentation, but unusable in a production context.

Death to Jupyter Notebooks

Jupyter Notebooks are *fantastic* for experimentation, but unusable in a production context.

What **not** to do: `https://github.com/jinensetpal/archimede.git`

Death to Jupyter Notebooks

Jupyter Notebooks are *fantastic* for experimentation, but unusable in a production context.

What **<u>not</u>** to do: `https://github.com/jinensetpal/archimede.git`
Because:

a. There's no real entrypoint.

Death to Jupyter Notebooks

Jupyter Notebooks are *fantastic* for experimentation, but unusable in a production context.

What **<u>not</u>** to do: `https://github.com/jinensetpal/archimede.git`
Because:

a. There's no real entrypoint.
b. Random pickled objects.

Death to Jupyter Notebooks

Jupyter Notebooks are *fantastic* for experimentation, but unusable in a production context.

What **not** to do: `https://github.com/jinensetpal/archimede.git`
Because:

a. There's no real entrypoint.
b. Random pickled objects.
c. No version control.

Death to Jupyter Notebooks

Jupyter Notebooks are *fantastic* for experimentation, but unusable in a production context.

What **not** to do: `https://github.com/jinensetpal/archimede.git`
Because:

a. There's no real entrypoint.
b. Random pickled objects.
c. No version control.

Despite being computationally inexpensive, and having open source {code, data, hyperparameters}, it's not *actually* helpful.

Death to Jupyter Notebooks

Jupyter Notebooks are *fantastic* for experimentation, but unusable in a production context.

What **not** to do: `https://github.com/jinensetpal/archimede.git`
Because:

a. There's no real entrypoint.
b. Random pickled objects.
c. No version control.

Despite being computationally inexpensive, and having open source $\{$code, data, hyperparameters$\}$, it's not *actually* helpful.

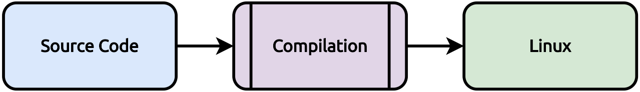
A clear solution for this requires us to recontextualize how we approach Machine Learning.

Let's Recontextualize ML Development

Idea: training $\approx$ compilation

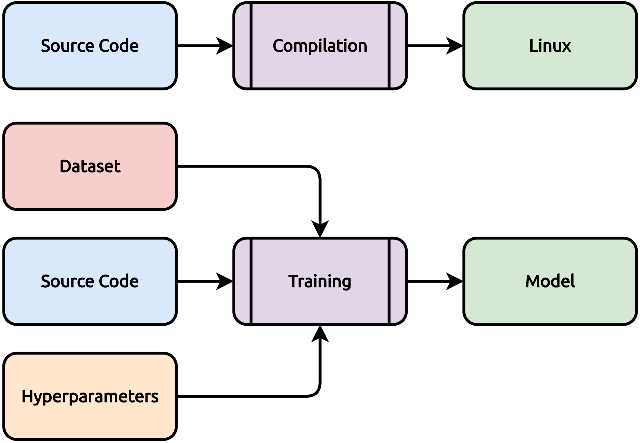
Let's Recontextualize ML Development

Idea: training $\approx$ compilation



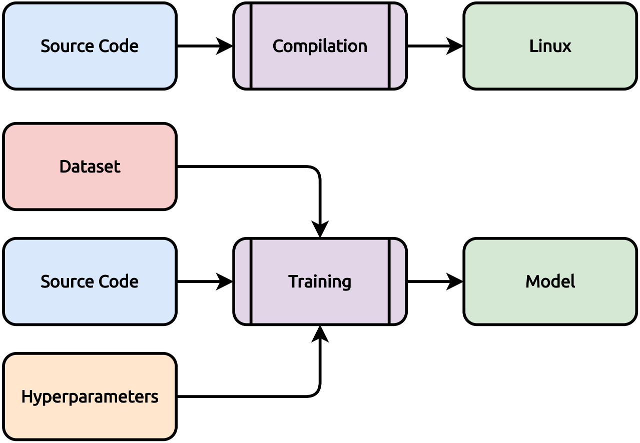
Let's Recontextualize ML Development

Idea: training $\approx$ compilation



Let's Recontextualize ML Development

Idea: training $\approx$ compilation



Key Difference: time(training) $>>$ time(compilation)

Reproducibility for Open Source Science

Machine Learning is a **science**.

Reproducibility for Open Source Science

Machine Learning is a **science**. Sometimes, the results of the experiments are production-ready. Then, it's also **software**.

Reproducibility for Open Source Science

Machine Learning is a **science**. Sometimes, the results of the experiments are production-ready. Then, it's also **software**.

Consequence: Traditional 'open source' *is not enough*.

Reproducibility for Open Source Science

Machine Learning is a **science**. Sometimes, the results of the experiments are production-ready. Then, it's also **software**.

Consequence: Traditional 'open source' *is not enough*.
Idea: Free and Open Source Science = Open Source + Reproducibility.

Reproducibility for Open Source Science

Machine Learning is a **science**. Sometimes, the results of the experiments are production-ready. Then, it's also **software**.

Consequence: Traditional 'open source' *is not enough*.
Idea: Free and Open Source Science = Open Source + Reproducibility.
Bonus: We can reuse the 'FOSS' acronym!

Reproducibility for Open Source Science

Machine Learning is a **science**. Sometimes, the results of the experiments are production-ready. Then, it's also **software**.

Consequence: Traditional 'open source' *is not enough*.
Idea: Free and Open Source Science = Open Source + Reproducibility.
Bonus: We can reuse the 'FOSS' acronym!

Important Note

This still is a partial answer. The democratization of accelerated hardware is still a **significant challenge** we fail to address.

How can we achieve this?

Step 0: Accept[2] that **not everything can be open**.

[2] begrudgingly

How can we achieve this?

Step 0: Accept[2] that **not everything can be open**. The maximal approach won't work.

[2] begrudgingly

How can we achieve this?

Step 0: Accept[2] that **not everything can be open**. The maximal approach won't work.

This is primarily owing to data privacy, and extends to model parameters.[3]

[2] begrudgingly

[3] `https://unlearning-challenge.github.io/`

How can we achieve this?

Step 0: Accept[2] that **not everything can be open**. The maximal approach won't work.

This is primarily owing to data privacy, and extends to model parameters.[3]

However, we *should* expect:

a. Documentation.

[2] begrudgingly

[3] `https://unlearning-challenge.github.io/`

How can we achieve this?

Step 0: Accept[2] that **not everything can be open**. The maximal approach won't work.

This is primarily owing to data privacy, and extends to model parameters.[3]

However, we *should* expect:

a. Documentation.

b. Synthetic Dataset Samples.

[2] begrudgingly

[3] `https://unlearning-challenge.github.io/`

How can we achieve this?

Step 0: Accept[2] that **not everything can be open**. The maximal approach won't work.

This is primarily owing to data privacy, and extends to model parameters.[3]

However, we *should* expect:

a. Documentation.
b. Synthetic Dataset Samples.
c. Training and Inference Code.

[2] begrudgingly
[3] `https://unlearning-challenge.github.io/`

How can we achieve this?

Step 0: Accept[2] that **not everything can be open**. The maximal approach won't work.

This is primarily owing to data privacy, and extends to model parameters.[3]

However, we *should* expect:

a. Documentation.
b. Synthetic Dataset Samples.
c. Training and Inference Code.
d. Descriptive whitepaper.

[2] begrudgingly
[3] `https://unlearning-challenge.github.io/`

How can we achieve this?

Step 0: Accept[2] that **not everything can be open**. The maximal approach won't work.

This is primarily owing to data privacy, and extends to model parameters.[3]

However, we *should* expect:

a. Documentation.
b. Synthetic Dataset Samples.
c. Training and Inference Code.
d. Descriptive whitepaper.
e. **Permissive Licensing**[4].

[2] begrudgingly
[3] `https://unlearning-challenge.github.io/`
[4] Widder et al. [2023]

How can we achieve this?

Step 0: Accept[2] that **not everything can be open**. The maximal approach won't work.

This is primarily owing to data privacy, and extends to model parameters.[3]

However, we *should* expect:

a. Documentation.
b. Synthetic Dataset Samples.
c. Training and Inference Code.
d. Descriptive whitepaper.
e. **Permissive Licensing**[4].

So; where do we go from here?

[2] begrudgingly
[3] `https://unlearning-challenge.github.io/`
[4] Widder et al. [2023]

The Reproducibility Checklist

Finally, we need to ensure that research can be replicated **by third parties**.

The Reproducibility Checklist

Finally, we need to ensure that research can be replicated **by third parties**.

For this, we use Dr. Pineau's **Reproducibility Checklist**[5].

[5] Pineau et al. [2021]

The Reproducibility Checklist

Finally, we need to ensure that research can be replicated **by third parties**.

For this, we use Dr. Pineau's **Reproducibility Checklist**[5]. Critical ideas:

a. **Models and algorithms** require clear descriptions of assumptions, settings and time-complexity analyses.

[5] Pineau et al. [2021]

The Reproducibility Checklist

Finally, we need to ensure that research can be replicated **by third parties**.

For this, we use Dr. Pineau's **Reproducibility Checklist**[5]. Critical ideas:

a. **Models and algorithms** require clear descriptions of assumptions, settings and time-complexity analyses.
b. **Datasets** must include statistics, splits, and pre-processing procedure.

[5] Pineau et al. [2021]

The Reproducibility Checklist

Finally, we need to ensure that research can be replicated **by third parties**.

For this, we use Dr. Pineau's **Reproducibility Checklist**[5]. Critical ideas:

a. **Models and algorithms** require clear descriptions of assumptions, settings and time-complexity analyses.
b. **Datasets** must include statistics, splits, and pre-processing procedure.
c. **Code** must specify requirements and code for training, inference as well as any pre-trained models.

[5] Pineau et al. [2021]

The Reproducibility Checklist

Finally, we need to ensure that research can be replicated **by third parties**.

For this, we use Dr. Pineau's **Reproducibility Checklist**[5]. Critical ideas:

a. **Models and algorithms** require clear descriptions of assumptions, settings and time-complexity analyses.

b. **Datasets** must include statistics, splits, and pre-processing procedure.

c. **Code** must specify requirements and code for training, inference as well as any pre-trained models.

d. **Experiments** must include the range of hyperparameters, the best configuration, the evaluation statistic and training hardware.

[5] Pineau et al. [2021]

The Reproducibility Checklist

Finally, we need to ensure that research can be replicated **by third parties**.

For this, we use Dr. Pineau's **Reproducibility Checklist**[5]. Critical ideas:

a. **Models and algorithms** require clear descriptions of assumptions, settings and time-complexity analyses.
b. **Datasets** must include statistics, splits, and pre-processing procedure.
c. **Code** must specify requirements and code for training, inference as well as any pre-trained models.
d. **Experiments** must include the range of hyperparameters, the best configuration, the evaluation statistic and training hardware.

As a consequence, we can realistically evaluate the claims made by the paper's authors.

[5] Pineau et al. [2021]

Thank you!

Have an awesome rest of your day!

Slides: `https://www.cs.purdue.edu/homes/jsetpal/slides/fossml.pdf`

References

Ali Koc and Abdullah Uz Tansel. A survey of version control systems. *ICEME 2011*, 2011.

Joelle Pineau, Philippe Vincent-Lamarre, Koustuv Sinha, Vincent Larivière, Alina Beygelzimer, Florence d'Alché Buc, Emily Fox, and Hugo Larochelle. Improving reproducibility in machine learning research (a report from the neurips 2019 reproducibility program). *The Journal of Machine Learning Research*, 22(1): 7459–7478, 2021.

Irene Solaiman. The gradient of generative ai release: Methods and considerations. In *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*, pages 111–122, 2023.

David Gray Widder, Sarah West, and Meredith Whittaker. Open (for business): Big tech, concentrated power, and the political economy of open ai. *Concentrated Power, and the Political Economy of Open AI (August 17, 2023)*, 2023.

Matei Zaharia, Andrew Chen, Aaron Davidson, Ali Ghodsi, Sue Ann Hong, Andy Konwinski, Siddharth Murching, Tomas Nykodym, Paul Ogilvie, Mani Parkhe, et al. Accelerating the machine learning lifecycle with mlflow. *IEEE Data Eng. Bull.*, 41(4): 39–45, 2018.

Appendix: Version Control it All

git is a brilliant tool that allows us to version control code; but what about data?

Enter **DVC**[6] (Data Version Control). It enables us to add, track, push, pull and checkout data.

Consequence: Data is now tracked. It's associated with a specific commit, and can be diffed.

[6] Koc and Tansel [2011]

Appendix: Systematic Experiment & Model Tracking

Next, we target the unpredictability of training.

We are not guaranteed a minima. Therefore, we track **metrics** and **hyperparameters**, to find the best set for a given run.

MLFlow[7] helps track and compare various experiments and parameters.

In addition, it allows tagging runs, registering models, and deploying a target model-as-a-service using Docker.

This tool manages the experiment-model **lifecycle**.

[7] Zaharia et al. [2018]

Appendix: Extensibility + the Overarching Principle

This is a sample framework intended to establish a baseline approach.

The goal is to extend this on a case-by-case basis; these concepts apply generally.

To adapt the **approach** to your use-case:

a. Find differences from the established standard.
b. Identify the parameters required to recreate the experimental setup.
c. Set hard / soft requirements based on criticality to replication & user privacy.