MLOps Homework #1 Summary

Luke Schwenke & Aaron Chan

October 23, 2023

GitHub & Git LFS

Ease of Installation

- Easy to install
- Good documentation
- Functions similarly to base git, and integrates well. Learning curve not very high if you are familiar with git.
- Integrates well with github (you won't have to create your own lfs server)

Ease of Data Versioning

- Essentially the same versioning system as git
- Tracks changes with each commit. A file tracked with lfs is stored on an lfs server instead of your git repo.
- No gui, so may be difficult to identify which version a set of data is in a large repo.

Ease of Switching Versions for Same Model

- Easy, provided you know how to use git
- Works essentially the same as git, but with a few specific lfs commands. Otherwise branching, committing, changing versions, functions basically the same.

---

lakeFS

Ease of Installation

- Easy to install and import
- Decent documentation on main website but not much discussion elsewhere on the internet
- Once you have the code setup for Python it is smooth and readable

Ease of Data Versioning

- lakeFS has multiple ways to do versioning but the most straightforward for our understanding was to switch between object-level (csv file) commits

- Unlike GitHub which stores the copy in a local folder that has Git enabled, lakeFS will connect to the remote cloud repo (AWS S3 bucket backend) and pull down the object via that commit ID
- In our implementation switching the data versioning requires amending a single index in the API call to the lakeFS server

Ease of Switching Versions for Same Model

- Not difficult, requires switching one value before querying the lakeFS remote database
- There are multiple ways to do this (references different branches, Commit ID's, etc.)

Model Results

Epochs = 5, Batch Size = 32, Loss = MSE

Model	MAE	MSE	RMSE
v1 (baseline)	1,159	23,581,957,829	153,564
v2	161	42,058	205
v2 - DP	165	45,212	213

DP was accomplished used the DPKerasAdamOptimizer

Differential Privacy Epsilon Results:

DP-SGD performed over 24,023 examples with 256 examples per iteration, noise multiplier 0.5 for 10 epochs with microbatching, and no bound on number of examples per user.

This privacy guarantee protects the release of all model checkpoints in addition to the final model.

Example-level DP with add-or-remove-one adjacency at delta = 1e-05 computed with RDP accounting:

- Epsilon with each example occurring once per epoch: **138.688**
- Epsilon assuming Poisson sampling (*): **139.347**

v2 Non-DP vs. v2 DP Results (Effect of DP on Metrics):
The non-DP model performed better than the DP model on the cleaned (v2) dataset. This is to be expected as DP intentionally adds noise to the dataset which increases the number of patterns the ML algorithm must learn. The trade-off is that the DP dataset masks personal data while keeping individual contributions within the dataset.