Experiment Reproducibility

Why does this matter?

You may have heard the phrase ‘reproducibility crisis’ thrown around.

It is a big problem in the data science community. People publish papers which make assertions, yet others are unable to reproduce the results.

This is not acceptable, in other sciences, e.g. physics, results must be repeated by other teams before they are considered authentic.

Without this the scientific method fails.

The data science community is gradually maturing in this regard.

What do we need to accomplish it?

We covered the formula before:

```
same code + same inputs = same output
```

For somebody else to reproduce our results, we must be able to tell them *exactly* which code and which data was used to produce a result.

- This means tracking parameters as we go
- This means tracking software versions against results

Thankfully, all of this can be automated

Using our cli tool, we will automate tracking all the versions of our experiments and the parameters.

We strongly encourage you to adopt a practice like this from the start. Not only will it make your work better it will actually make life much easier long run.

Our tool will work like this:

- For each experiment we run, we are going to create a directory with all the results in.
- We will also save a file with all of the parameters used to generate those results.
- We will also record the git commit which produced the results.

A few limits on reproducibility

It’s worth pointing out that there are a couple of limits to reproducibility. They essentially boil down to a degree of non determinism in modern methods and tools:

- Random seeds, non-determinism of GPU operations
- Floating point errors

Overall these won’t affect an outcome given enough trials, but the details will often be very slightly different.

Resources

Rolling our own experiment tracking system is a great learning experience. There is nothing wrong with doing it this way.

There are a lot of projects out there that offer this kind of functionality though, they're worth looking at:

- https://metaflow.org/
- https://mlflow.org/
- https://neptune.ai/

PyTorch lightning offers lots of utilities that make some of this stuff easier too:

https://www.pytorchlightning.ai/