A Practical Taxonomy of Reproducibility for Machine Learning Research

Rachael Tatman
Kaggle
@rctatman
rachael@kaggle.com

Jake VanderPlas
eScience Institute
University of Washington
jakevp@uw.edu

Sohier Dane
Kaggle
sohier@kaggle.com

Background

Why care about reproducibility?
- It’s good science, but it also helps ML practitioners
- Research code is sample code; if people can’t get it to run they can’t apply your findings
- Reproducible code has a broader impact than non-reproducible code

Related work & contribution
- There’s been a lot of discussion of the importance of reproducibility [1,2,3,4,5,6]
- Here, we offer a practical framework for evaluating the reproducibility of a project and tips for improving reproducibility

As a scale
- Reproducibility is a spectrum, not a binary
- We propose an updated version of Peng (2011’s) reproducibility scale (see below) with three levels [see sidebar ->]
- The less time a reproducer needs to spend on a project, the more reproducible it is

Low Reproducibility
- Only sharing the paper
- Was standard before the ability to easily share code & data
- Example: Gelly & Silver 2007 [9]

Medium Reproducibility
- Sharing both code and data (if data was used, it should be anonymized & shared)
- Currently the most common way of sharing research code
- Still requires substantial time investment to get environment set up (need to account for versions and subservations of requirements)
- Some tips for improving medium reproducibility research:
  a. Separate preprocessing, modeling & evaluation and distribute data and code for each step
  b. Document the original environment
  c. Ensure that your code and data are licensed for reuse (see Morin et al. [11])

High Reproducibility
- Sharing data, code, and the environment needed to run the code
- Three options for sharing executable environments (in order of decreasing time commitment)
  a. Virtual machines
  b. Containers
  c. Hosted notebooks/scripts

Other Considerations

Too few researchers share their code/data ●
- <40% of papers from NIPS 2017 shared their code
- There were high reproducibility papers, through, like Liu et al [14]

Link rot is a big problem ●
- Code that is shared does not always remain available
- 20% of links in NLP papers were deprecated within 5 years [15]

Reproducibility Taxonomy

High:
- Code shared
- Data shared
- Environment shared

Medium:
- Code shared
- Data shared

Low:
- Finished paper only
- No code or data shared

"Reproducible" here means achieving the same results/output as the original paper using the same data*
* See [7] & [8] for a discussion of terminology: in many fields this is known code "reproducible"

Reproducibility scale from Peng (2011) [1]

Virtual machines
- Code, data, all dependencies and a complete OS
- Excellent reproducibility, but files can be very large and slow to spin up
- Popular options: VirtualBox, VMware

Containers
- Code, data and all the dependencies needed to run the code in a simple portable format
- Use the OS of the local system, so can be difficulties moving between OS’s
- Popular options: Docker

Hosted notebooks/scripts
- Allows reproduction from a browser
- Generally faster and easier to run (don’t require large downloads or set up time)
- Most services don’t provide enough free compute to reproduce very computationally intensive studies

Commercial options: Kaggle Kernels, Google Collaboratory, Amazon SageMaker, IBM Watson Studio, Azure Notebooks

Not-for-profit options: MyBinder, PanGeo, Codalab

---

[9] Sylvain Gelly and David Silver. Combining online and offline knowledge in UCT. In Pro-