Reproducibility of deep learning models in cognitive computational neuroscience

by **Martina Vilas** (she/her)

@martinagvilas

who am I?

- cognitive neuroscience PhD student

  at Max-Planck-Institute AE, Germany
- core contributor of The Turing Way
- from Argentina

what is **reproducible research**?

what is **reproducible research**?

same analytic steps on the same dataset produces same answer

		data	
		same	different
analysis	same	reproducible	replicable
	different	robust	generalisable

reproducibility **crisis**

SHARE

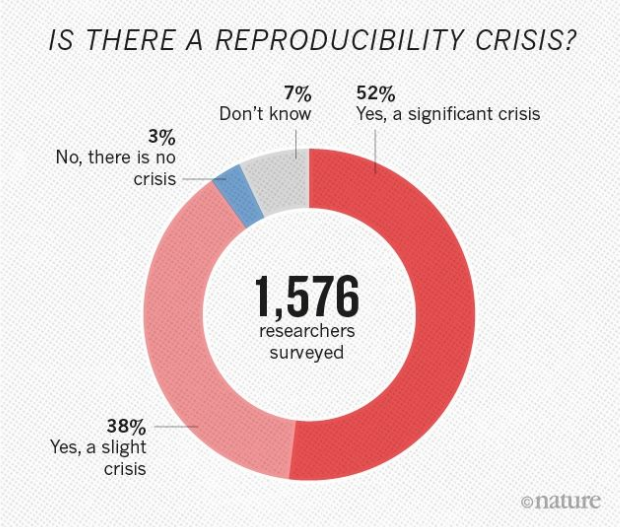
IN DEPTH | COMPUTER SCIENCE

Artificial intelligence faces reproducibility crisis

Matthew Hutson

+ See all authors and affiliations

Science 16 Feb 2018:
Vol. 359, Issue 6377, pp. 725-726
DOI: 10.1126/science.359.6377.725



GREGORY BARBER BUSINESS 09.16.2019 07:00 AM

Artificial Intelligence Confronts a 'Reproducibility' Crisis

Machine-learning systems are black boxes even to the researchers that build them. That makes it hard for others to assess the results.

Forbes

Oct 26, 2018, 08:00am EDT

How Do We Address The Reproducibility Crisis In Artificial Intelligence?

Matt Jones Forbes Councils Member
Forbes Technology Council COUNCIL POST | Membership (fee-based)
Innovation

POST WRITTEN BY
Matt Jones

tools for reproducible research

tools for reproducible research

- sharing code and data is not enough

tools for reproducible research

- sharing code and data is not enough
- we also need to:

→ **tools** for reproducible research

- sharing code and data is not enough
- we also need to:
  - ✓ capture the computational environment

→ **tools** for reproducible research

- sharing code and data is not enough
- we also need to:
  - ✓ capture the computational environment

tools for reproducible research

- sharing code and data is not enough
- we also need to:
  - ✓ capture the computational environment
  - ✓ use version control systems

→ **tools** for reproducible research

- sharing code and data is not enough
- we also need to:

✓ capture the computational environment

✓ use version control systems

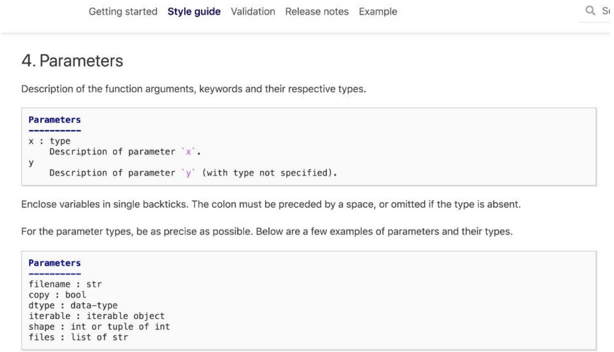
git

tools for reproducible research

- sharing code and data is not enough
- we also need to:
  - ✓ capture the computational environment
  - ✓ use version control systems
  - ✓ have good documentation of the code and data

tools for reproducible research

- sharing code and data is not enough
- we also need to:
  - ✓ capture the computational environment
  - ✓ use version control systems
  - ✓ have good documentation of the code and data



Getting started **Style guide** Validation Release notes Example

4. Parameters

Description of the function arguments, keywords and their respective types.

```
Parameters
----------
x : type
    Description of parameter `x`.
y
    Description of parameter `y` (with type not specified).
```

Enclose variables in single backticks. The colon must be preceded by a space, or omitted if the type is absent.

For the parameter types, be as precise as possible. Below are a few examples of parameters and their types.

```
Parameters
----------
filename : str
copy : bool
dtype : data-type
iterable : iterable object
shape : int or tuple of int
files : list of str
```

tools for reproducible research

- sharing code and data is not enough
- we also need to:
  - ✓ capture the computational environment
  - ✓ use version control systems
  - ✓ have good documentation of the code and data
  - ✓ test the code

→ **tools** for reproducible research

- sharing code and data is not enough
- we also need to:
  - ✓ capture the computational environment
  - ✓ use version control systems
  - ✓ have good documentation of the code and data
  - ✓ test the code

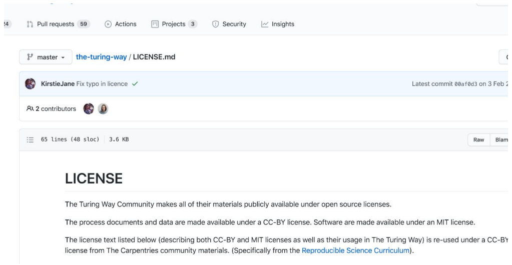
pytest

tools for reproducible research

- sharing code and data is not enough
- we also need to:
  - ✓ capture the computational environment
  - ✓ use version control systems
  - ✓ have good documentation of the code and data
  - ✓ test the code
  - ✓ make the project open source

→ **tools** for reproducible research

- sharing code and data is not enough
- we also need to:
  - ✓ capture the computational environment
  - ✓ use version control systems
  - ✓ have good documentation of the code and data
  - ✓ test the code
  - ✓ make the project open source



https://the-turing-way.netlify.app/reproducible-research/open/open-source.html
https://the-turing-way.netlify.app/reproducible-research/licensing.html

tools for reproducible research

- sharing code and data is not enough
- we also need to:
  - ✓ capture the computational environment
  - ✓ use version control systems
  - ✓ have good documentation of the code and data
  - ✓ test the code
  - ✓ make the project open source
  - ✓ etc.

The Turing Way

The Turing Way

online guide to

- reproducible
- ethical
- inclusive
- collaborative

… data science

The Turing Way

Search this book...

Visit our GitHub Repository
This book is powered by Jupyter Book

Contents
Our Community
Citing *The Turing Way*

Welcome

The Turing Way is an open source community-driven guide to reproducible, ethical, inclusive and collaborative data science.

Our goal is to provide all the information that data scientists in academia, industry, government and the third sector need at the start of their projects to ensure that they are easy to reproduce and reuse at the end.

The book started as a guide for reproducibility, covering version control, testing, and continuous integration. However, technical skills are just one aspect of making data science research "open for all".

In February 2020, *The Turing Way* expanded to a series of books covering reproducible research, project design, communication, collaboration, and ethical research.

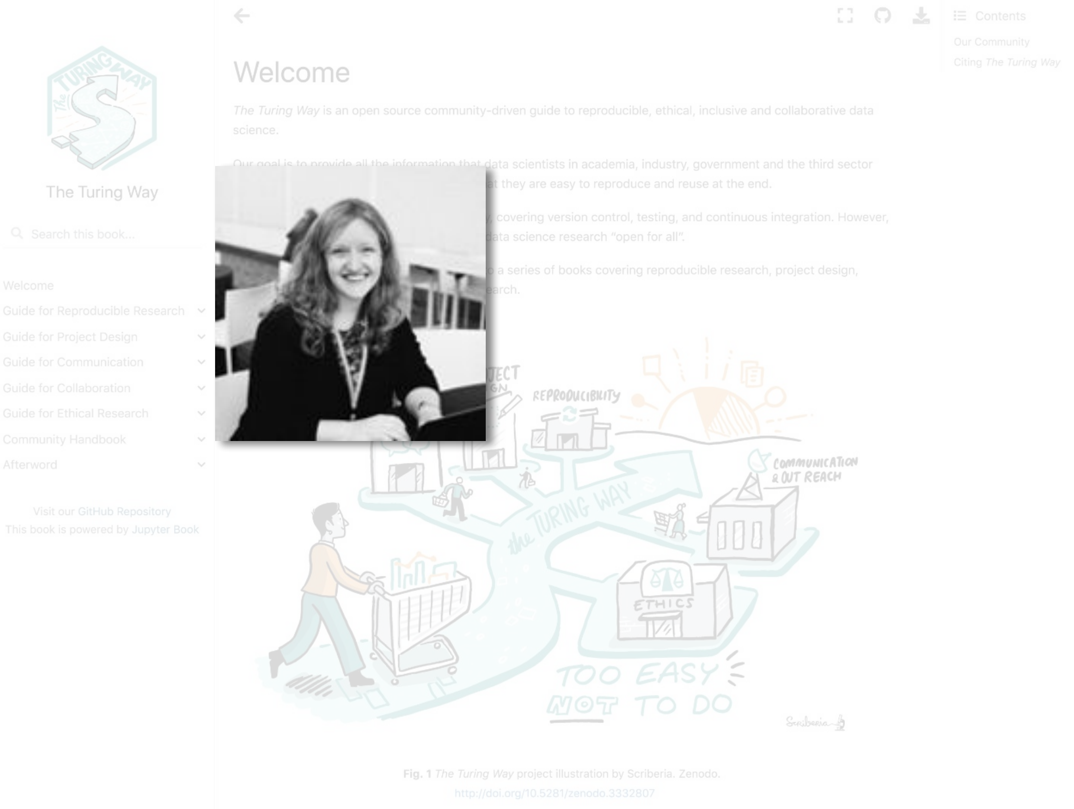
Fig. 1 *The Turing Way* project illustration by Scriberia. Zenodo. http://doi.org/10.5281/zenodo.3332807

The Turing Way

online guide to

- reproducible
- ethical
- inclusive
- collaborative

... data science

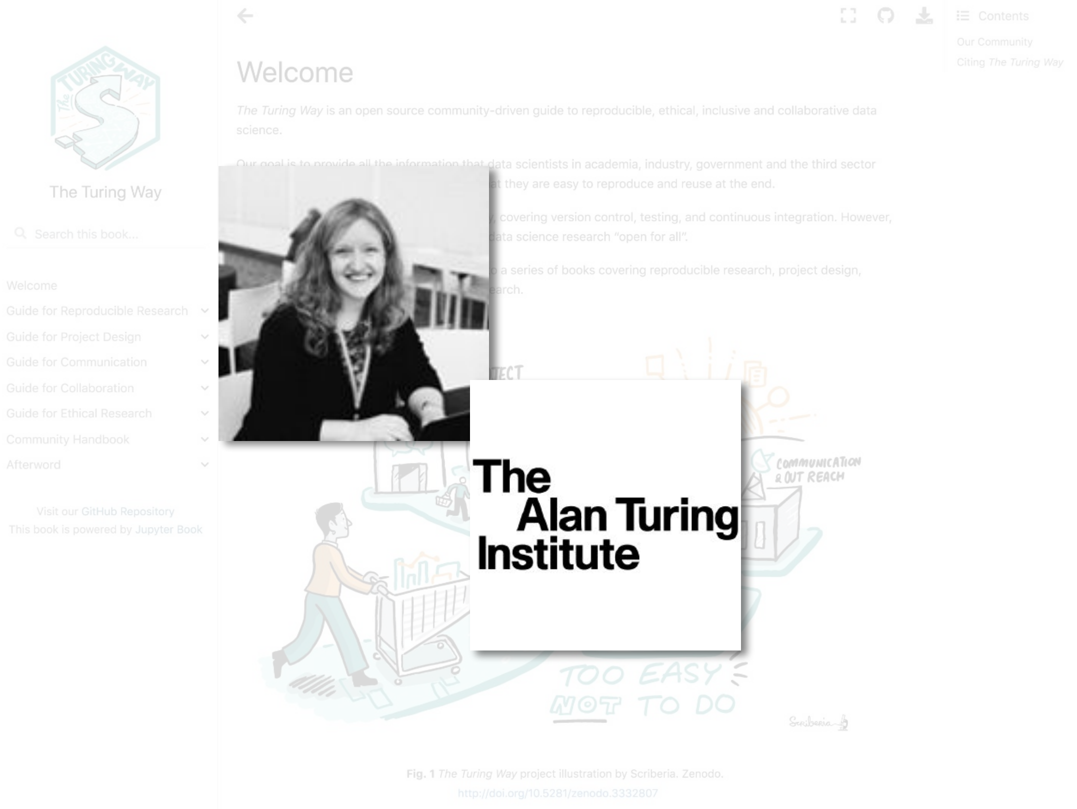


The Turing Way

online guide to

- reproducible
- ethical
- inclusive
- collaborative

… data science



https://the-turing-way.netlify.app/ @martinagvilas

The Turing Way

The Turing Way

Search this book...

Guide for Reproducible Research

This guide covers topics related to skills, tools and best practices for research reproducibility.

The Turing Way defines reproducibility in data research as data and code being available to fully rerun the analysis.

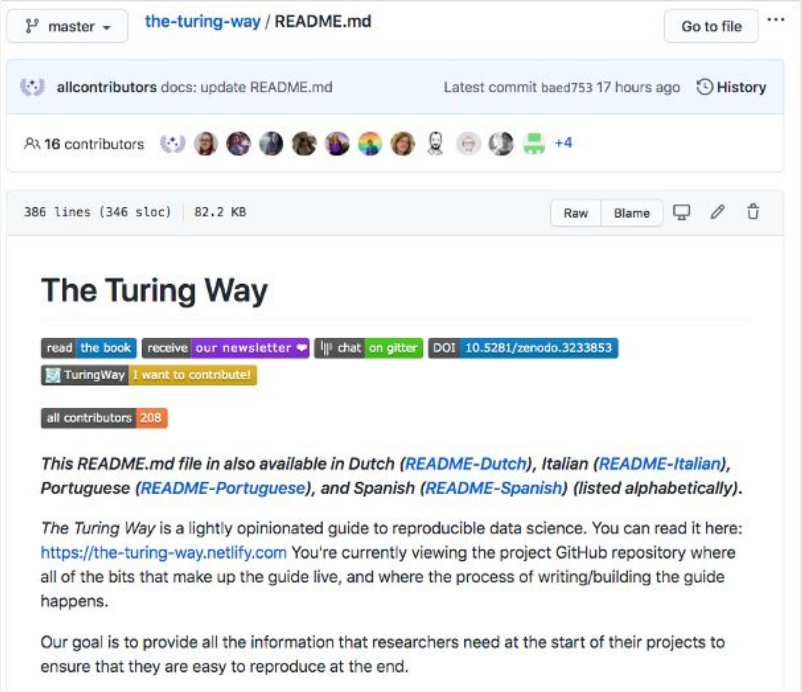
There are several definitions of reproducibility in use, and we discuss these in more detail in the Definitions section of this chapter. While it is absolutely fine for us each to use different words, it will be useful for you to know how *The Turing Way* defines *reproducibility* to avoid misunderstandings when reading the rest of the handbook.

Fig. 2 *The Turing Way* project illustration by Scriberia. Zenodo. http://doi.org/10.5281/zenodo.3332807

The Turing Way started by defining reproducibility in the context of this handbook, laying out its importance for science and scientists, and providing an overview of the common concepts, tools and resources. The first few chapters were on version control, testing, and reproducible computational environments. Since the start of this project in 2019, many additional chapters have been written, edited, reviewed, read and promoted by over 100

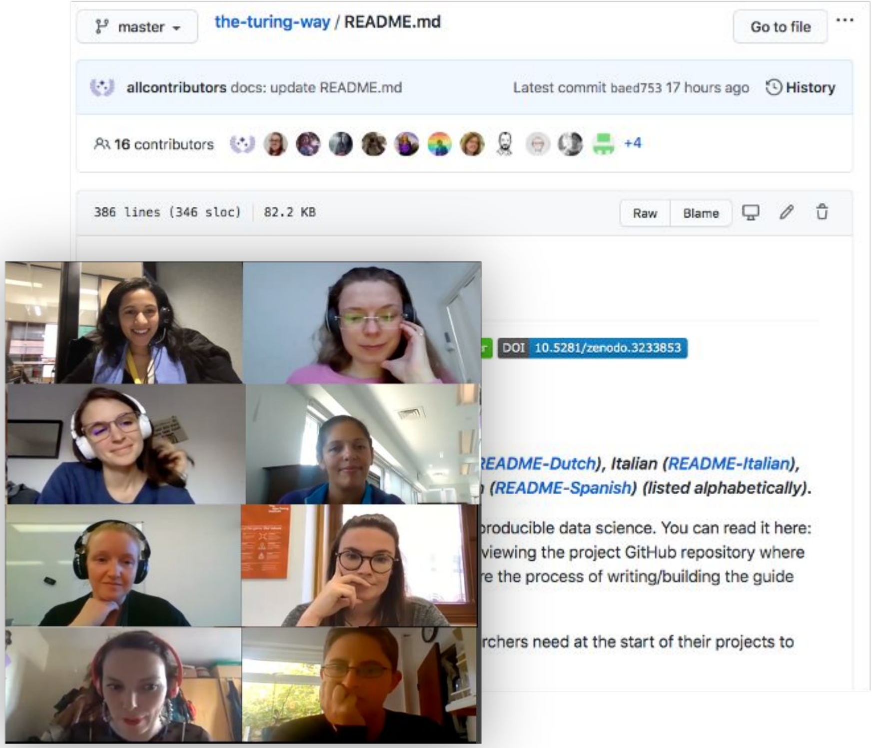
The Turing Way

- open source project



The Turing Way

- open source project
- community



The Turing Way

- Twitter:

twitter.com/turingway

- Newsletter:

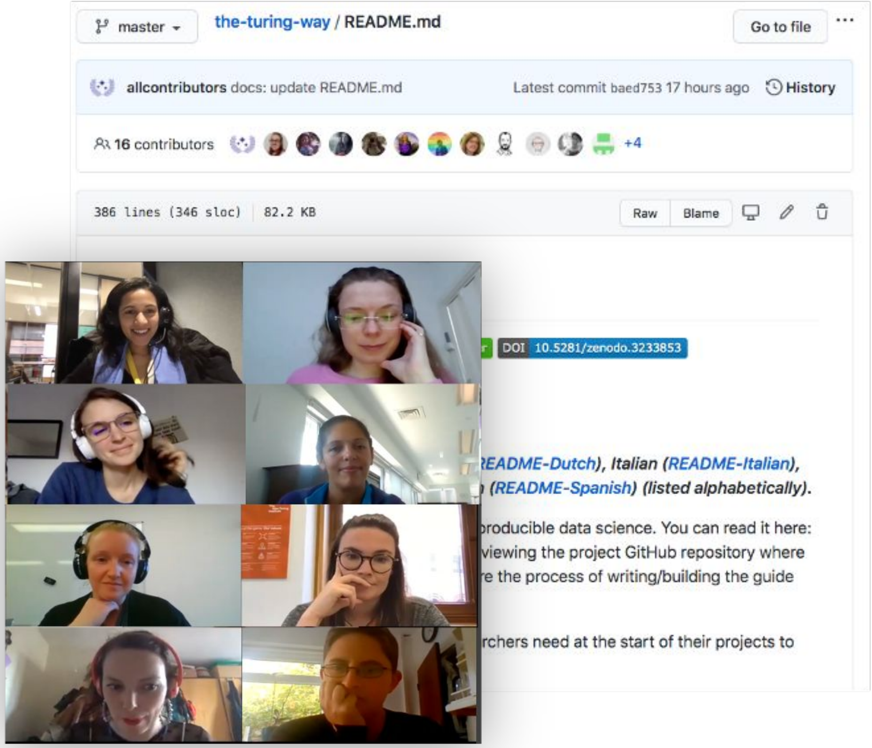
tinyletter.com/TuringWay

- GitHub:

github.com/alan-turing-institute/the-turing-way

- Slack:

https://tinyurl.com/jointuringwayslack



reproducibility in Deep Learning

reproducibility in Deep Learning

results vary by, for example:

- hyperparameters
- random initialization
- train/test split
- dataset
- etc.

ACCOUNTING FOR VARIANCE IN MACHINE LEARNING BENCHMARKS

Xavier Bouthillier[1 2] **Pierre Delaunay**[3] **Mirko Bronzi**[1] **Assya Trofimov**[1 2 4] **Brennan Nichyporuk**[1 5 6]
Justin Szeto[1 5 6] **Naz Sepah**[1 5 6] **Edward Raff**[7 8] **Kanika Madan**[1 2] **Vikram Voleti**[1 2]
Samira Ebrahimi Kahou[1 6 9 10] **Vincent Michalski**[1 2] **Dmitriy Serdyuk**[1 2] **Tal Arbel**[1 5 6 10] **Chris Pal**[1 11 12]
Gaël Varoquaux[1 6 13] **Pascal Vincent**[1 2 10]

Bouthillier et al. (2021). Accounting for Variance in Machine Learning Benchmarks.

➔ reproducibility in Deep Learning

results vary by, for example:

- hyperparameters
- random initialization
- train/test split
- dataset
- etc.

➔ report all these in detail, keep them constant, or log their variation

logging

MLflow

- Quickstart
- Tutorials and Examples
- Concepts
- − MLflow Tracking
  - Concepts
  - Where Runs Are Recorded
  - \+ How Runs and Artifacts are Recorded
  - \+ Logging Data to Runs
  - − Automatic Logging
    - Scikit-learn (experimental)
    - TensorFlow and Keras (experimental)
    - Gluon (experimental)
    - XGBoost (experimental)
    - LightGBM (experimental)
    - Statsmodels (experimental)
    - Spark (experimental)
    - Fastai (experimental)
    - Pytorch (experimental)
  - \+ Organizing Runs in Experiments

Concepts

MLflow Tracking is organized around the concept of *runs*, which are executions of some piece of data science code. Each run records the following information:

Code Version

Git commit hash used for the run, if it was run from an MLflow Project.

Start & End Time

Start and end time of the run

Source

Name of the file to launch the run, or the project name and entry point for the run if run from an MLflow Project.

Parameters

Key-value input parameters of your choice. Both keys and values are strings.

Metrics

Key-value metrics, where the value is numeric. Each metric can be updated throughout the course of the run (for example, to track how your model's loss fun MLflow records and lets you visualize the metric's full history.

Artifacts

Output files in any format. For example, you can record images (for example, PNGs), models (for example, a pickled scikit-learn model), and data files (for exa artifacts.

https://mlflow.org/docs/latest/tracking.html

packaging

MLproject File

You can get more control over an MLflow Project by adding an `MLproject` file, which is a text file in YAML syntax, to the project's root directory. The following is an example of an `MLproject` file:

```
name: My Project

conda_env: my_env.yaml
# Can have a docker_env instead of a conda_env, e.g.
# docker_env:
#    image:  mlflow-docker-example

entry_points:
  main:
    parameters:
      data_file: path
      regularization: {type: float, default: 0.1}
    command: "python train.py -r {regularization} {data_file}"
  validate:
    parameters:
      data_file: path
    command: "python validate.py {data_file}"
```

The file can specify a name and a Conda or Docker environment, as well as more detailed information about each entry point. Specifically, each entry point defines a command to run an parameters to pass to the command (including data types).

Specifying an Environment

This section describes how to specify Conda and Docker container environments in an `MLproject` file. `MLproject` files cannot specify *both* a Conda environment and a Docker environment.

Conda environment

Include a top-level **`conda_env`** entry in the **`MLproject`** file. The value of this entry must be a *relative* path to a Conda environment YAML file within the MLflow project's directory. In following example:

```
conda_env: files/config/conda_environment.yaml
```

`conda_env` refers to an environment file located at **`<MLFLOW_PROJECT_DIRECTORY>/files/config/conda_environment.yaml`**, where **`<MLFLOW_PROJECT_DIRECTORY>`** is the path to the MLflow project's root directory.

Docker container environment

https://mlflow.org/docs/latest/projects.html

→ **tools** for reproducible deep learning

Out-of-the-box Reproducibility: A Survey of Machine Learning Platforms

Richard Isdahl
Department of Computer Science
Norwegian University of Science and Technology
Trondheim, Norway

Odd Erik Gundersen
Department of Computer Science
Norwegian University of Science and Technology
Trondheim, Norway
odderik@ntnu.no

Koustuv Sinha — about blog activities projects publications

Tools

Updated : 21st December, 2020

	Practice	Tools
1	Config Management	Hydra, OmegaConf, Pytorch Lightning
2	Checkpoint Management	Pytorch Lightning, TestTube
3	Logging	Tensorboard, Comet.ML, Weights & Biases, MLFlow, Visdom, Neptune
4	Seed	*Check best practices below*
-	Experiment Management	Pytorch Lightning, MLFlow, Determined.AI
5	Versioning	Github, Gitlab, Replicate.AI
6	Data Management	DVC, CML, Replicate.AI
7	Data analysis	Jupyter Notebook, papermill, JupyterLab, Google Colab
8	Reporting	Matplotlib, Seaborn , Pandas, Overleaf
9	Dependency Management	pip, conda, Poetry, Docker, Singularity, repo2docker
10	Open Source Release	Squash Commits, Binder
11	Effective Communication	ML Code Completeness Checklist, ML Reproducibility Checklist
12	Test and Validate	AWS, GCP, CodeOcean

Gundersen & Kjensmo (2019). Out-of-the-box Reproducibility: A Survey of Machine Learning Platforms.
https://www.cs.mcgill.ca/~ksinha4/practices_for_reproducibility/

ML reproducibility challenges

ML reproducibility challenges

ML Reproducibility Challenge 2020 and Spring 2021

Welcome to the ML Reproducibility Challenge 2020! This is already the fourth edition of this event (see V1, V2, V3), and we are excited this year to announce that we are broadening our coverage of conferences and papers to cover several new top venues, including: NeurIPS, ICML, ICLR, ACL, EMNLP, CVPR and ECCV.

The primary goal of this event is to encourage the publishing and sharing of scientific results that are reliable and reproducible. In support of this, the objective of this challenge is to investigate reproducibility of papers accepted for publication at top conferences by inviting members of the community at large to select a paper, and verify the empirical results and claims in the paper by reproducing the computational experiments, either via a new implementation or using code/data or other information provided by the authors.

All submitted reports will be peer reviewed and shown next to the original papers on Papers with Code. Reports will be peer-reviewed via OpenReview. Every year, a small number of these reports, selected for their clarity, thoroughness, correctness and insights, are selected for publication in a special edition of the journal ReScience. (see J1, J2).

OpenReview.net Search OpenReview...

ML Reproducibility Challenge 2020

RC2020

TBD Mar 12 2021 https://paperswithcode.com/rc2020 reproducibility.challenge@gmail.com

Spring 2021

Submission Start: Oct 05 2020 12:00AM UTC-0, End: Jan 30 2021 11:59AM UTC-0

Accepted for ReScience | Submissions

[Re] Satellite Image Time Series Classification with Pixel-Set Encoders and Temporal Self-Attention
Maja Schneider, Marco Körner
06 Dec 2020 (modified: 01 Apr 2021) RC2020 Readers: Everyone 4 Replies
Show details

Reimplementation of FixMatch and Investigation on Noisy (Pseudo) Labels and Confirmation Errors of FixMatch
Ci Li, Ruibo Tu, Hui Zhang
06 Dec 2020 (modified: 01 Apr 2021) RC2020 Readers: Everyone 4 Replies
Show details

[Reproducibility Report] Rigging the Lottery: Making All Tickets Winners
Varun Sundar, Rajat Vadiraj Dwaraknath
22 Jan 2021 (modified: 03 Apr 2021) RC2020 Readers: Everyone 3 Replies
Show details

[Re] Can gradient clipping mitigate label noise?
David Mizrahi, Oğuz Kaan Yüksel, Aiday Marlen Kyzy
31 Jan 2021 (modified: 08 Apr 2021) RC2020 Readers: Everyone 4 Replies
Show details

checklists for ML publications

The Machine Learning Reproducibility Checklist (v2.0, Apr.7 2020)

For all **models** and **algorithms** presented, check if you include:

- ❑ A clear description of the mathematical setting, algorithm, and/or model.
- ❑ A clear explanation of any assumptions.
- ❑ An analysis of the complexity (time, space, sample size) of any algorithm.

For any **theoretical claim**, check if you include:

- ❑ A clear statement of the claim.
- ❑ A complete proof of the claim.

For all **datasets** used, check if you include:

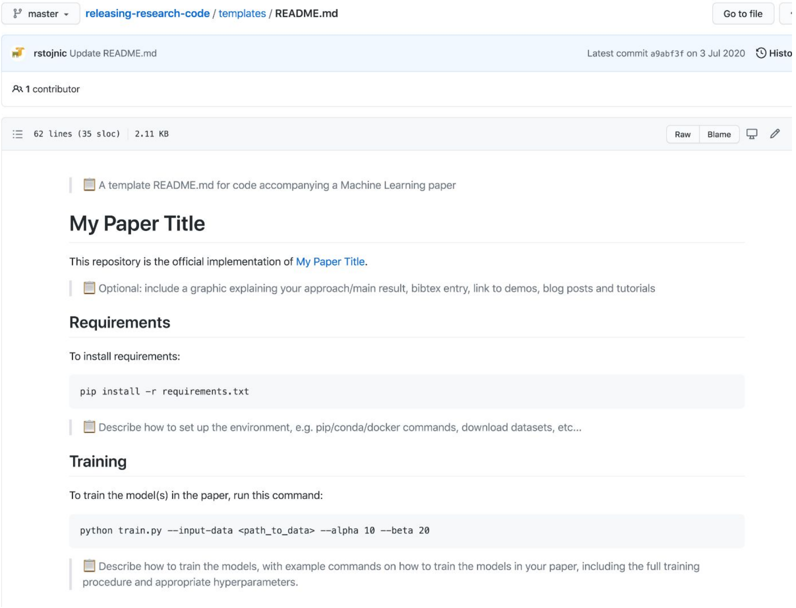
- ❑ The relevant statistics, such as number of examples.
- ❑ The details of train / validation / test splits.
- ❑ An explanation of any data that were excluded, and all pre-processing step.
- ❑ A link to a downloadable version of the dataset or simulation environment.
- ❑ For new data collected, a complete description of the data collection process, such as instructions to annotators and methods for quality control.

For all shared **code** related to this work, check if you include:

- ❑ Specification of dependencies.
- ❑ Training code.
- ❑ Evaluation code.
- ❑ (Pre-)trained model(s).
- ❑ README file includes table of results accompanied by precise command to run to produce those results.

For all reported **experimental results**, check if you include:

- ❑ The range of hyper-parameters considered, method to select the best hyper-parameter configuration, and specification of all hyper-parameters used to generate results.
- ❑ The exact number of training and evaluation runs.
- ❑ A clear definition of the specific measure or statistics used to report results.
- ❑ A description of results with central tendency (e.g. mean) & variation (e.g. error bars).
- ❑ The average runtime for each result, or estimated energy cost.
- ❑ A description of the computing infrastructure used.



master / releasing-research-code / templates / README.md

Go to file

rstojnic Update README.md — Latest commit a9abf3f on 3 Jul 2020 — Histo

1 contributor

62 lines (35 sloc) | 2.11 KB — Raw — Blame

> 📋 A template README.md for code accompanying a Machine Learning paper

My Paper Title

This repository is the official implementation of My Paper Title.

> 📋 Optional: include a graphic explaining your approach/main result, bibtex entry, link to demos, blog posts and tutorials

Requirements

To install requirements:

```
pip install -r requirements.txt
```

> 📋 Describe how to set up the environment, e.g. pip/conda/docker commands, download datasets, etc...

Training

To train the model(s) in the paper, run this command:

```
python train.py --input-data <path_to_data> --alpha 10 --beta 20
```

> 📋 Describe how to train the models, with example commands on how to train the models in your paper, including the full training procedure and appropriate hyperparameters.

https://github.com/paperswithcode/releasing-research-code/blob/master/templates/README.md

https://www.cs.mcgill.ca/~jpineau/ReproducibilityChecklist.pdf

beyond reproducible research

beyond reproducible research

		data	
		same	different
analysis	same	reproducible	replicable
	different	robust	generalisable

https://the-turing-way.netlify.app/reproducible-research/overview/overview-definitions.html

@martinagvilas

beyond reproducible research

model vs. instantiation of the model

Bouthillier et al. (2019). *Unreproducible research is reproducible.*

beyond reproducible research

model vs. instantiation of the model

what we try to estimate

in science

beyond reproducible research

model vs. instantiation of the model

"specific set of (trained) parameter values for a given model"

Bouthillier et al. (2019). *Unreproducible research is reproducible.*

@martinagvilas

beyond reproducible research

model vs. instantiation of the model

"useful as a probe to better understand a model"

Bouthillier et al. (2019). *Unreproducible research is reproducible.*

beyond reproducible research

"Sources of variations such as the initialization should not be fixed. Conclusions on a model that are limited to a single instance are very weak."

Bouthillier et al. (2019). *Unreproducible research is reproducible.*

beyond reproducible research

"Sources of variations such as the initialization should not be fixed. Conclusions on a model that are limited to a single instance are very weak."

- all sources of variation kept constant → very poor generalizability of scientific claim

Bouthillier et al. (2019). *Unreproducible research is reproducible.*

beyond reproducible research

1. make the scientific claim very clear

Bouthillier et al. (2019). *Unreproducible research is reproducible.*

beyond reproducible research

1. make the scientific claim very clear

    → e.g. model A performs better than model B in visual segmentation tasks

Bouthillier et al. (2019). *Unreproducible research is reproducible.*

beyond reproducible research

1. make the scientific claim very clear
   - e.g. model A performs better than model B in visual segmentation tasks
2. ask yourself: which sources of variation should not affect this scientific claim?

beyond reproducible research

1. make the scientific claim very clear
   - e.g. model A performs better than model B in visual segmentation tasks
2. ask yourself: which sources of variation should not affect this scientific claim?
   - e.g. computational environment, initialization, test/train split, dataset

Bouthillier et al. (2019). *Unreproducible research is reproducible.*

beyond reproducible research

1. make the scientific claim very clear
   - e.g. model A performs better than model B in visual segmentation tasks
2. ask yourself: which sources of variation should not affect this scientific claim?
   - e.g. computational environment, initialization, test/train split, dataset
3. investigate the generalizability of the claim under those irrelevant conditions

Bouthillier et al. (2019). *Unreproducible research is reproducible.*

beyond reproducible research

≠ types of scientific claims

↓

≠ types of generalizability checks

@martinagvilas

→ cognitive computational neuroscience (CCN)

cognitive computational neuroscience (CCN)

- understand how the human brain implements cognitive functions

cognitive computational neuroscience (CCN)

- understand how the human brain implements cognitive functions

examples:

- perception
- attention
- memory
- problem solving

→ cognitive computational neuroscience (CCN)

- DNN research → build a deep learning model that achieves the best performance for a task or a task/dataset
- CCN research → build a neurobiologically plausible computational model that performs a cognitive tasks similarly to humans

Kriegeskorte & Douglas (2018). *Cognitive Computational Neuroscience.*

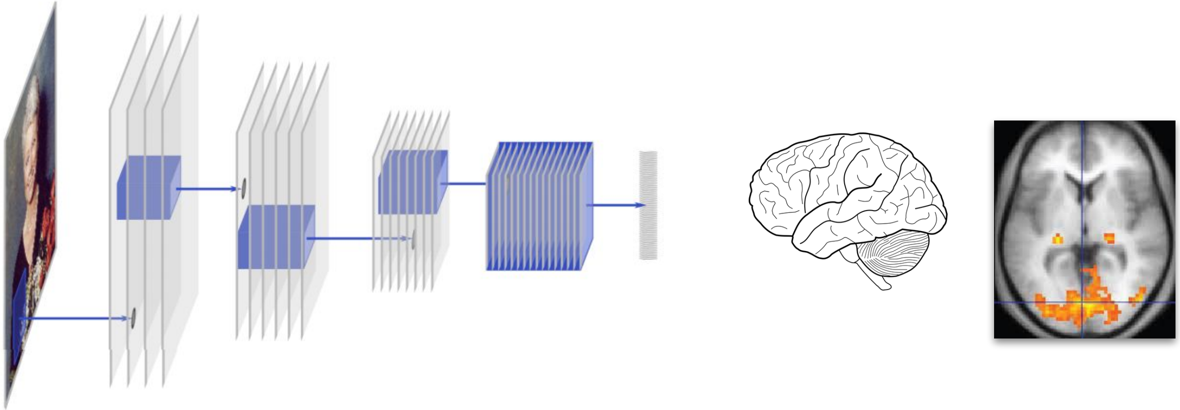
cognitive computational neuroscience (CCN)

- DNN research → build a deep learning model that achieves the best performance for a task or a task/dataset
- CCN research → build a neurobiologically plausible computational model that performs a cognitive tasks similarly to humans

deep learning model

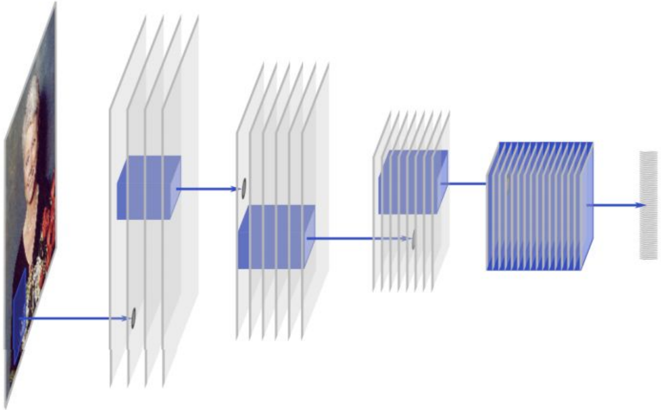
Kriegeskorte & Douglas (2018). *Cognitive Computational Neuroscience.*

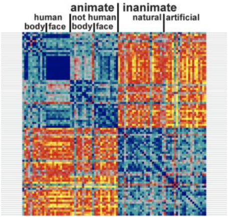
deep learning modeling in **CCN**



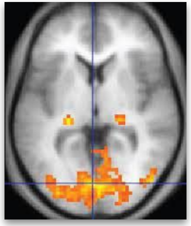
Kietzmann et al. (2018). Deep Neural Networks in Computational Neuroscience.

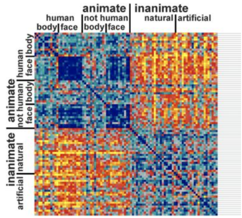
→ **deep learning** modeling in **CCN**





IT-geometry-supervised deep conv. network





human IT

types of deep learning modeling in CCN

types of deep learning modeling in CCN

- mechanistic model

types of deep learning modeling in CCN

- mechanistic model
  - *goal* → understand how the brain computes information

types of deep learning modeling in CCN

- mechanistic model
  - *goal* → understand how the brain computes information
  - *e.g.* inspect:
    - ✓ network architecture
    - ✓ learning goal (objective function)
    - ✓ learning update rule

types of deep learning modeling in CCN

- mechanistic model
- representational model

types of deep learning modeling in CCN

- mechanistic model

- representational model

  - *goal* → understand how information is represented in the brain

types of deep learning modeling in CCN

- mechanistic model
- representational model
- decoding model

types of deep learning modeling in CCN

- mechanistic model
- representational model
- decoding model
    - *goal* → understand what information is used by the brain

generalizability of deep learning models in CCN

- mechanistic model
  - network architecture
  - learning goal (objective function)
  - learning update rule
- representational model
- decoding model

generalizability of deep learning models in CCN

approaches for generalizability

Cooper & Guest (2013) Implementations are not specifications: Specification, replication and experimentation in computational cognitive modeling.

@martinagvilas

generalizability of deep learning models in CCN

approaches for generalizability

- computational experimentation → systematically manipulate sources of variation and study how the behavior of the model changes

Cooper & Guest (2013) Implementations are not specifications: Specification, replication and experimentation in computational cognitive modeling.

approaches for generalizability

- computational experimentation → systematically manipulate sources of variation and study how the behavior of the model changes
- analyze multiple instantiations of the model

Mehrer et al. (2020). Individual differences among deep neural network models.

generalizability of deep learning models in CCN

approaches for generalizability

- computational experimentation → systematically manipulate sources of variation and study how the behavior of the model changes
- analyze multiple instantiations of the model
- re-implement the model

Cooper & Guest (2013) Implementations are not specifications: Specification, replication and experimentation in computational cognitive modeling.

→ **generalizability** of deep learning models in CCN

approaches for generalizability

- computational experimentation → systematically manipulate sources of variation and study how the behavior of the model changes
- analyze multiple instantiations of the model
- re-implement the model
- etc.

take home messages

take home messages

- is important to ensure research reproducibility across scientific fields

take home messages

- is important to ensure research reproducibility across scientific fields
- reproducibility is more than sharing your code and data

→ **take home** messages

- is important to ensure research reproducibility across scientific fields
- reproducibility is more than sharing your code and data
- many tools for ensuring and learning about reproducibility exist → get to know them!

→ take home messages

- is important to ensure research reproducibility across scientific fields
- reproducibility is more than sharing your code and data
- many tools for ensuring and learning about reproducibility exist → get to know them!
- we should also go beyond reproducibility and think about generalizability

→ take home messages

- is important to ensure research reproducibility across scientific fields
- reproducibility is more than sharing your code and data
- many tools for ensuring and learning about reproducibility exist → get to know them!
- we should also go beyond reproducibility and think about generalizability
- each scientific field has its own reproducibility and generalizability challenges, even if they use the same analytical tool

thank you!

@martinagvilas