The Journal of Open Source Software

Foolbox Native: Fast adversarial attacks to benchmark the robustness of machine learning models in PyTorch, TensorFlow, and JAX

Jonas Rauber[1, 2], **Roland Zimmermann**[1, 2], **Matthias Bethge***[1, 3], **and Wieland Brendel**[1, 3]

1 Tübingen AI Center, University of Tübingen, Germany **2** International Max Planck Research School for Intelligent Systems, Tübingen, Germany **3** Bernstein Center for Computational Neuroscience Tübingen, Germany

DOI: 10.21105/joss.02607

Software

- Review
- Repository
- Archive

Editor: Yuan Tang

Reviewers:

- @GregaVrbancic
- @ethanwharris

Submitted: 10 August 2020
Published: 27 September 2020

Summary

Machine learning has made enormous progress in recent years and is now being used in many real-world applications. Nevertheless, even state-of-the-art machine learning models can be fooled by small, maliciously crafted perturbations of their input data. Foolbox is a popular Python library to benchmark the robustness of machine learning models against these adversarial perturbations. It comes with a huge collection of state-of-the-art adversarial attacks to find adversarial perturbations and thanks to its framework-agnostic design it is ideally suited for comparing the robustness of many different models implemented in different frameworks. Foolbox 3 aka Foolbox Native has been rewritten from scratch to achieve native performance on models developed in PyTorch (Paszke et al., 2019), TensorFlow (Abadi et al., 2016), and JAX (Bradbury et al., 2018), all with one codebase without code duplication.

Statement of need

Evaluating the adversarial robustness of machine learning models is crucial to understanding their shortcomings and quantifying the implications on safety, security, and interpretability. Foolbox Native is the first adversarial robustness toolbox that is both fast and framework-agnostic. This is important because modern machine learning models such as deep neural networks are often computationally expensive and are implemented in different frameworks such as PyTorch and TensorFlow. Foolbox Native combines the framework-agnostic design of the original Foolbox (Rauber, Brendel, & Bethge, 2017) with real batch support and native performance in PyTorch, TensorFlow, and JAX, all using a single codebase without code duplication. To achieve this, all adversarial attacks have been rewritten from scratch and now use EagerPy (Rauber et al., 2020) instead of NumPy (Oliphant, 2006) to interface *natively* with the different frameworks.

This is great for both users and developers of adversarial attacks. Users can efficiently evaluate the robustness of different models in different frameworks using the same set of state-of-the-art adversarial attacks, thus obtaining comparable results. Attack developers do not need to choose between supporting just one framework or reimplementing their new adversarial attack multiple times and dealing with code duplication. In addition, they both benefit from the comprehensive type annotations (Rossum, Lehtosalo, & Langa, 2015) in Foolbox Native to catch bugs even before running their code.

*joint senior authors

The combination of being framework-agnostic and simultaneously achieving native performance sets Foolbox Native apart from other adversarial attack libraries. The most popular alternative to Foolbox is CleverHans[1]. It was the first adversarial attack library and has traditionally focused solely on TensorFlow (plans to make it framework-agnostic *in the future* have been announced). The original Foolbox was the second adversarial attack library and the first one to be framework-agnostic. Back then, this was achieved at the expense of performance. The adversarial robustness toolbox ART[2] is another framework-agnostic adversarial attack library, but it is conceptually inspired by the original Foolbox and thus comes with the same performance trade-off. AdverTorch[3] is a popular adversarial attack library that was inspired by the original Foolbox but improved its performance by focusing soley on PyTorch. Foolbox Native is our attempt to improve the performance of Foolbox without sacrificing the framework-agnostic design that is crucial to consistently evaluate the robustness of different machine learning models that use different frameworks.

Use Cases

Foolbox was designed to make adversarial attacks easy to apply even without expert knowledge. It has been used in numerous scientific publications and has already been cited more than 220 times. On GitHub it has received contributions from several developers and has gathered more than 1.500 stars. It provides the reference implementations of various adversarial attacks, including the Boundary Attack (Brendel, Rauber, & Bethge, 2018), the Pointwise Attack (Schott, Rauber, Bethge, & Brendel, 2019), clipping-aware noise attacks (Rauber & Bethge, 2020), the Brendel Bethge Attack (Brendel, Rauber, Kümmerer, Ustyuzhaninov, & Bethge, 2019), and the HopSkipJump Attack (Chen, Jordan, & Wainwright, 2020), and is under active development since 2017.

Acknowledgements

J.R. acknowledges support from the Bosch Research Foundation (Stifterverband, T113/30057/17) and the International Max Planck Research School for Intelligent Systems (IMPRS-IS). This work was supported by the German Federal Ministry of Education and Research (BMBF): Tübingen AI Center, FKZ: 01IS18039A, and by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/Interior Business Center (DoI/IBC) contract number D16PC00003. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/IBC, or the U.S. Government.

We thank all contributors to Foolbox, in particular Behar Veliqi, Evgenia Rusak, Jianbo Chen, Rene Bidart, Jerome Rony, Ben Feinstein, Eric R Meissner, Lars Holdijk, Lukas Schott, Carl-Johann Simon-Gabriel, Apostolos Modas, William Fleshman, Xuefei Ning, and many others.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., et al. (2016). TensorFlow: A system for large-scale machine learning. In *12th USENIX symposium on operating systems design and implementation (OSDI 16)* (pp. 265–283).

[1] https://github.com/tensorflow/cleverhans
[2] https://github.com/Trusted-AI/adversarial-robustness-toolbox
[3] https://github.com/BorealisAI/advertorch

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., & Wanderman-Milne, S. (2018). JAX: Composable transformations of Python+NumPy programs. Retrieved from http://github.com/google/jax

Brendel, W., Rauber, J., & Bethge, M. (2018). Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In *International conference on learning representations*. Retrieved from https://openreview.net/forum?id=SyZI0GWCZ

Brendel, W., Rauber, J., Kümmerer, M., Ustyuzhaninov, I., & Bethge, M. (2019). Accurate, reliable and fast robustness evaluation. In *Advances in neural information processing systems 32*.

Chen, J., Jordan, M. I., & Wainwright, M. J. (2020). HopSkipJumpAttack: A query-efficient decision-based attack. In *2020 ieee symposium on security and privacy (sp)* (pp. 1277–1294). IEEE. doi:10.1109/SP40000.2020.00045

Oliphant, T. (2006). NumPy: A guide to NumPy. USA: Trelgol Publishing. Retrieved from http://www.numpy.org/

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., et al. (2019). PyTorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems* (pp. 8026–8037).

Rauber, J., & Bethge, M. (2020). Fast differentiable clipping-aware normalization and rescaling. *arXiv preprint arXiv:2007.07677*. Retrieved from https://github.com/jonasrauber/clipping-aware-rescaling

Rauber, J., Bethge, M., & Brendel, W. (2020). EagerPy: Writing code that works natively with PyTorch, TensorFlow, JAX, and NumPy. *arXiv preprint arXiv:2008.04175*. Retrieved from https://eagerpy.jonasrauber.de

Rauber, J., Brendel, W., & Bethge, M. (2017). Foolbox: A Python toolbox to benchmark the robustness of machine learning models. In *Reliable machine learning in the wild workshop, 34th international conference on machine learning*. Retrieved from https://arxiv.org/abs/1707.04131

Rossum, G. van, Lehtosalo, J., & Langa, Ł. (2015). *Type hints* (PEP No. 484). Python Software Foundation. Retrieved from https://www.python.org/dev/peps/pep-0484/

Schott, L., Rauber, J., Bethge, M., & Brendel, W. (2019). Towards the first adversarially robust neural network model on MNIST. In *International conference on learning representations*. Retrieved from https://openreview.net/forum?id=S1EHOsC9tX