The Journal of Open Source Software

Delira: A High-Level Framework for Deep Learning in Medical Image Analysis

Christoph Haarburger[1], Justus Schock[1], Michael Baumgartner[1], Oliver Rippel[1], and Dorit Merhof[1]

1 Institute of Imaging and Computer Vision, RWTH Aachen University, Germany

DOI: 10.21105/joss.01488

Software

- Review
- Repository
- Archive

Submitted: 27 May 2019
Published: 17 June 2019

Summary

Medical image analysis research using deep neural networks often involves the development of problem-specific network architectures and the evaluation of models on several datasets. Contemporary deep learning frameworks such as PyTorch (Paszke et al., 2017) and Tensorflow (Abadi et al., 2016), however, operate on a low level, such that for comparing different models on several datasets, a lot of boilerplate code is necessary. So far, this boilerplate code is often copied and pasted for new projects and experiments. Reference implementations of new methods may be implemented in either PyTorch or Tensorflow, leading to a lot of friction when comparing two methods that are implemented in different low-level frameworks. Moreover, data augmentation for 3D medical images such as from computed tomography or magnetic resonance images is not natively supported by many low-level frameworks. As a result, stand alone data augmentation solutions are often applied (MIC-DKFZ, 2019).

In order to integrate high level functionalities such as logging, data structures for image datasets, data augmentation, trainer classes and model save and load functionality in a way that is agnostic with respect to the low-level framework, we developed `Delira` (Deep Learning in Radiology).

`Delira` sonsists of serveral subpackages and modules that are structured into `data_loading`, `io`, `logging`, `models`, `training` and `utils`. This modular structure enables the reuse of datasets and data loading pipelines across different models. Moreover, reference models for classification, segmentation and data synthesis problems using generative adversarial networks (Goodfellow et al., 2014) are provided in the `models` subpackage.

The actual training is carried out using a `NetworkTrainer` class that implements the actual training routine given a dataset and model. An `Experiment` class runs the training using `NetworkTrainer`, e.g. in a cross validation scheme. A quick tutorial showing how the most important data structures interact with each other and HTML documentation is provided at https://delira.readthedocs.io/en/master/classification_pytorch.html.

Currently, PyTorch and Tensorflow backends are supported and tested. Adding more backends is easily possible if needed.

`Delira` is released under BSD Clause-2 license. The source code can be found at https://github.com/justusschock/delira.

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).

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., et al. (2014). Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems 27* (pp. 2672–2680). Curran Associates, Inc. Retrieved from http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf

MIC-DKFZ. (2019). Batchgenerators. Retrieved May 17, 2019, from https://github.com/MIC-DKFZ/batchgenerators

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., et al. (2017). Automatic differentiation in pytorch. In *NIPS 2017 autodiff workshop*.