The Journal of Open Source Software

NeuralHydrology — A Python library for Deep Learning research in hydrology

Frederik Kratzert[1], **Martin Gauch**[2], **Grey Nearing**[1], **and Daniel Klotz**[2]

1 Google Research **2** Institute for Machine Learning, Johannes Kepler University Linz, Linz, Austria

DOI: 10.21105/joss.04050

Software

- Review
- Repository
- Archive

Editor: Jayaram Hariharan

Reviewers:

- @ammilten
- @chuckaustin
- @jhamman

Submitted: 21 December 2021
Published: 04 March 2022

Summary and statement of need

Since ancient times humans have strived to describe environmental processes related to water (Angelakis et al., 2012; Biswas, 1970). Throughout this history, hydrologists built various process-based prediction models that simulate processes from soil moisture to streamflow generation (a collection of historical references can be found in Loague (2010)). More recently, Deep Learning models have emerged as extremely powerful and more accurate alternatives to these traditional modeling approaches (Gauch et al., 2021; Klotz et al., 2021; Kratzert, Klotz, Shalev, et al., 2019; Kratzert, Klotz, Herrnegger, et al., 2019; Lees et al., 2021). For hydrologists, embracing this new data-driven paradigm is challenging (Beven, 2020; Nearing et al., 2021): Not only are the models conceptually different, but they are also built, optimized, and evaluated with different strategies and toolsets.

NeuralHydrology is a Python library based on PyTorch (Paszke et al., 2019) that is designed to build, apply, and experiment with Deep Learning models with a strong focus on hydrological applications. Originally designed for our internal research needs, the library was generalized and open-sourced to allow anyone to experiment with Deep Learning models as easily as possible: pre-built models and data loaders allow for quick experiments, yet the framework is also easily extensible to new models, data sets, loss functions, or metrics to suit more advanced use-cases.

NeuralHydrology is targeted towards students and researchers who want to experiment with Deep Learning models for rainfall-runoff modeling or other problems related to hydrology. As such, the library was designed to be picked up by beginners with little programming experience. For example, NeuralHydrology allows to train state-of-the-art rainfall-runoff models by only editing a config file and without touching a single line of code.

Functionality

NeuralHydrology is available on GitHub and can be installed via the pip package manager or by cloning the repository. The documentation provides a detailed explanation of the installation steps.

Basic Concepts

At the core of each experiment with NeuralHydrology is a YAML configuration file that specifies input and target data, model architecture, metrics to calculate, as well as training, validation, and test periods, along with some more technical settings that are described in the documentation. The configuration file also specifies whether the experiment is conducted on a CPU or a GPU — for Deep Learning experiments, the latter can drastically improve runtimes.

The library can be used in two ways: First, via the command line interface (CLI). After installation via pip, the package registers several commands that can be executed via the command line, most importantly `nh_run`. This is the main entrypoint to start training and

validation of models. Second, via the Python API. This option allows for more sophisticated use-cases, as users can define more precisely which parts of the framework they intend to use. There exist methods to conduct a full training or evaluation just like it is possible through the CLI, but it is equally possible to only use specific models, metrics, or data sets in the context of one's own code. NeuralHydrology also provides scripts for job scheduling to run multiple experiments in parallel on one or more GPUs.

Lastly, the modular design of the library allows contributors not only to extend the functionality by adding new models, data sets, or metrics, but also to run comparative experiments: for instance, one can compare different models in the same setting or one model on different data sets.

Example use-cases

The most basic use-case for the NeuralHydrology library is basic rainfall–runoff modeling as described in Kratzert et al. (2018) and Kratzert, Klotz, Shalev, et al. (2019), where we train a machine learning model on meteorological data and discharge observations from a set of basins (a process also known as calibration) and subsequently apply the model to a different time period, either to evaluate its goodness-of-fit, or to actually generate predictions for practical use. Since the experiment configuration is highly flexible, users can define which inputs they want to ingest into the model (e.g., whether discharge from previous time steps should be an input variable) and which variable(s) they want to predict. Users can easily train an ensemble of multiple models and average their predictions, which often greatly enhances the overall accuracy (see e.g. Kratzert, Klotz, Shalev, et al. (2019) in the context of LSTM-based rainfall-runoff modeling).

Beyond this most basic application, NeuralHydrology also supports prediction in ungauged basins, where a model is trained on one set of basins and subsequently applied to another set of basins for which no discharge observations exist (see e.g. Kratzert, Klotz, Herrnegger, et al. (2019)). Lastly, NeuralHydrology also includes different options to train neural networks for uncertainty estimation, where the model does not only regress a single target value (e.g. discharge) per time step, but rather outputs probability distributions for each target variable (for details, see Klotz et al. (2021)).

References

Angelakis, A., Mays, L., Koutsoyiannis, D., & Mamassis, N. (2012). *Evolution of water supply through the millennia*. IWA Publishing. https://doi.org/10.2166/9781780401041

Beven, K. (2020). Deep learning, hydrological processes and the uniqueness of place. *Hydrological Processes*, *34*(16), 3608–3613. https://doi.org/10.1002/hyp.13805

Biswas, A. (1970). *History of hydrology*. North Holland Publishing Company.

Gauch, M., Kratzert, F., Klotz, D., Nearing, G., Lin, J., & Hochreiter, S. (2021). Rainfall–runoff prediction at multiple timescales with a single long short-term memory network. *Hydrology and Earth System Sciences*, *25*(4), 2045–2062. https://doi.org/10.5194/hess-25-2045-2021

Klotz, D., Kratzert, F., Gauch, M., Sampson, A., Brandstetter, J., Klambauer, G., Hochreiter, S., & Nearing, G. (2021). Uncertainty estimation with deep learning for rainfall–runoff modelling. *Hydrology and Earth System Sciences Discussions*, *2021*, 1–32. https://doi.org/10.5194/hess-2021-154

Kratzert, F., Klotz, D., Brenner, C., Schulz, K., & Herrnegger, M. (2018). Rainfall–runoff modelling using long short-term memory (LSTM) networks. *Hydrology and Earth System Sciences*, *22*(11), 6005–6022. https://doi.org/10.5194/hess-22-6005-2018

Kratzert, F., Klotz, D., Herrnegger, M., Sampson, A., & Hochreiter, G., S.and Nearing. (2019). Toward improved predictions in ungauged basins: Exploiting the power of machine learning. *Water Resources Research*, *55*(12), 11344–11354. https://doi.org/10.1029/2019WR026065

Kratzert, F., Klotz, D., Shalev, G., Klambauer, G., Hochreiter, S., & Nearing, G. (2019). Towards learning universal, regional, and local hydrological behaviors via machine learning applied to large-sample datasets. *Hydrology and Earth System Sciences*, *23*(12), 5089–5110. https://doi.org/10.5194/hess-23-5089-2019

Lees, T., Buechel, M., Anderson, B., Slater, L., Reece, S., Coxon, G., & Dadson, S. J. (2021). Benchmarking data-driven rainfall-runoff models in great britain: A comparison of LSTM-based models with four lumped conceptual models. *Hydrology and Earth System Sciences Discussions*, *2021*, 1–41. https://doi.org/10.5194/hess-2021-127

Loague, K. M. (2010). *Rainfall-runoff modelling* (Vol. 4). IAHS Press Wallingford, UK.

Nearing, G., Kratzert, F., Sampson, A., Pelissier, C., Klotz, D., Frame, J., Prieto, C., & Gupta, H. (2021). What role does hydrological science play in the age of machine learning? *Water Resources Research*, *57*(3). https://doi.org/10.1029/2020WR028091

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., … Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems* (Vol. 32). Curran Associates, Inc.