The Journal of Open Source Software

MiniSom, a minimalistic and Numpy based implementation of the Self Organizing Maps

Giuseppe Vettigli[1, 2, 3]

1 Centrica plc (current) **2** Cybernetics Institute, Italian National Research Council (previous) **3** Parthenope University (previous)

DOI: DOIunavailable

Software

- Review
- Repository
- Archive

Editor: Pending Editor

Reviewers:

- @Pending Reviewers

Submitted: N/A
Published: N/A

Summary

Self Organizing Maps (SOM) is a type of Artificial Neural Network (Kohonen, 1990) that is able to organize itself so that specific areas respond in a similar way to input patterns that are similar. Since its first formulation, it has been successfully used for a plethora of applications in many scientific fields and the Machine Learning community has developed a staggering amount of variants of the original model. In (Kohonen, 2014) is reported that over 10,000 scientific papers and more than 20 books have been published on it. `MiniSom` is a minimalistic and Numpy (Harris et al., 2020) based implementation of SOM.

Statement of need

In a scenario where Python has become one of the major languages for scientific development, `MiniSom` serves three main purposes. First, offer an implementation of SOM in Python which is easy to use and adapt. Second, give researchers the ability to easily create variants of the main SOM model. Third, offer students an implementation of SOM which is easy to understand.

The interface of `MiniSom` has evolved to blend with popular Machine Learning frameworks, as `scikit-learn` (Pedregosa et al., 2011), and the visualization library `matplotlib` (Hunter, 2007). The documentation of the library is proposed through examples based on `ipython` notebooks (Pérez & Granger, 2007) and uses the cited libraries.

Applications

At the time I am writing, `Minisom` has been cited in more than 50 scientific publications[1]. It has been used in many typical Machine Learning applications, such as time series modeling (Fortuin et al., 2018) and text mining (Makiyama et al., 2015). And it has also been used as a tool in a variety of fields, such as Geophysics (Lessin et al., 2020), Climatology (Thompson et al., 2020), and Network Security (Nam et al., 2018). Also, `MiniSom` has been used for the creation of teaching material for courses at University level and MOOCs, see (Ludwig Krippahl, n.d.) for an example of teaching material based on `MiniSom`. I'm also aware of industrial applications of `MiniSom` at TrendMiner[2].

Historical note

`MiniSom` was developed while creating a Machine Learning methodology to embed structured data (graphs and trees) into vectorial spaces (Vettigli, 2012; Vettigli & Ciaramella, 2017). The development has been made while the author was affiliated with institutions 2 and 3.

[1] This was estimated via Google Scholar including theses and dissertations.
[2] https://www.trendminer.com

References

Fortuin, V., Hüser, M., Locatello, F., Strathmann, H., & Rätsch, G. (2018). Som-vae: Interpretable discrete representation learning on time series. *arXiv Preprint arXiv:1806.02199*.

Harris, C. R., Millman, K. J., Walt, S. J. van der, Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Kerkwijk, M. H. van, Brett, M., Haldane, A., Río, J. F. del, Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, *9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE*, *78*(9), 1464–1480.

Kohonen, T. (2014). MATLAB implementations and applications of the self-organizing map. *Unigrafia Oy, Helsinki, Finland*, *177*.

Lessin, G., Polimene, L., Artioli, Y., Butenschön, M., Clark, D. R., Brown, I., & Rees, A. P. (2020). Modeling the seasonality and controls of nitrous oxide emissions on the northwest european continental shelf. *Journal of Geophysical Research: Biogeosciences*, e2019JG005613.

Ludwig Krippahl, F. A. (n.d.). *Lecture notes for the course of automatic learning at the university of lisbon*. http://aa.ssdi.di.fct.unl.pt/files/AA-16_notes.pdf.

Makiyama, V. H., Raddick, J., & Santos, R. D. (2015). Text mining applied to SQL queries: A case study for the SDSS SkyServer. *SIMBig*, 66–72.

Nam, T. M., Phong, P. H., Khoa, T. D., Huong, T. T., Nam, P. N., Thanh, N. H., Thang, L. X., Tuan, P. A., Loi, V. D., & others. (2018). Self-organizing map-based approaches in DDoS flooding detection using SDN. *2018 International Conference on Information Networking (ICOIN)*, 249–254.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., & others. (2011). Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research*, *12*, 2825–2830.

Pérez, F., & Granger, B. E. (2007). IPython: A system for interactive scientific computing. *Computing in Science and Engineering*, *9*(3), 21–29. https://doi.org/10.1109/MCSE.2007.53

Thompson, H. D., Déry, S. J., Jackson, P. L., & Laval, B. E. (2020). A synoptic climatology of potential seiche-inducing winds in a large intermontane lake: Quesnel lake, british columbia, canada. *International Journal of Climatology*.

Vettigli, G. (2012). *Schemi di apprendimento ispirati a mappe auto organizzanti per l'elaborazione di dati strutturati* [Master's thesis]. Parthenope University.

Vettigli, G., & Ciaramella, A. (2017). Fuzzy clustering of structured data: Some preliminary results. *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 1–6.