Clifford Neural Layers

This project aims to optimize Clifford neural layers, intorduced in [1, 2], to model rigid body transformations and simulate fluid dynamics. These layers are extensions of the usual convolution layers relying on geometric algebra, an emerging tool to model computational geometry [3] based on real Clifford algebras.

Starting from a PyTorch implementation[1], the goal is to create a high-performance version of the layers written in C or C++ (with crucial parts done in C) targeting X86_64 or ARM CPUs. This project focuses exclusively on inference, eliminating the need for implementing backward passes. Additionally, it is essential that the code provides a compatible interface to be integrated with the existing code base. Three types of layers should be optimized: i) geometric Clifford algebra linear layers [2], ii) geometric Clifford algebra nonlinearities (multi-vector sigmoid linear units [4]), and iii) group normalizations [5].

References

[1] Brandstetter, J., Berg, R.V.D., Welling, M. and Gupta, J.K., 2022. Clifford neural layers for pde modeling. arXiv preprint arXiv:2209.04934.
[2] Ruhe, D., Gupta, J.K., De Keninck, S., Welling, M. and Brandstetter, J., 2023, July. Geometric clifford algebra networks. In *International Conference on Machine Learning* (pp. 29306-29337). PMLR.
[3] Dorst, L., Fontijne, D. and Mann, S., 2009. Geometric algebra for computer science (revised edition): An object-oriented approach to geometry. Morgan Kaufmann.
[4] Sabour, S., Frosst, N. and Hinton, G.E., 2017. Dynamic routing between capsules. Advances in neural information processing systems, 30.
[5] Wu, Y. and He, K., 2018. Group normalization. In Proceedings of the European conference on computer vision (ECCV) (pp. 3-19).

[1] https://github.com/microsoft/cliffordlayers