Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing

Prof. R. Wattenhofer

Scaling Graph Neural Networks Using Differentiable Trees

The number of pairwise relations between object scales quadratically with the number of objects. For classical problems such as finding the nearest neighbor, we have developed fast algorithms that trade some precision for speed. Unfortunately, in many Machine Learning (ML) problems these approximations have yet to be developed. For example, in the field of Natural Language Processing (NLP), the attention layer between the encoder and the decoder is usually the main bottleneck of the neural network [1]–[4]. Similarly, in Graph Neural Networks (GNNs), the number of possible edges scales quadratically with the number of graph nodes. This limits GNNs to *predefined and fixed* sparse graphs or a small number of nodes.

The goal of this project is to develop a new type of architecture for neural networks that takes a list of n feature vectors as input and outputs a relational pattern, i.e. a graph with a computational cost smaller than $\mathcal{O}\left(n^2\right)$. To achieve this, a special learnable tree which can be backpropagated through will be constructed and studied. The student will test the architecture first on a simple toy problem, then on standard graph benchmarks and, if the time allows for it, on an NLP problem.

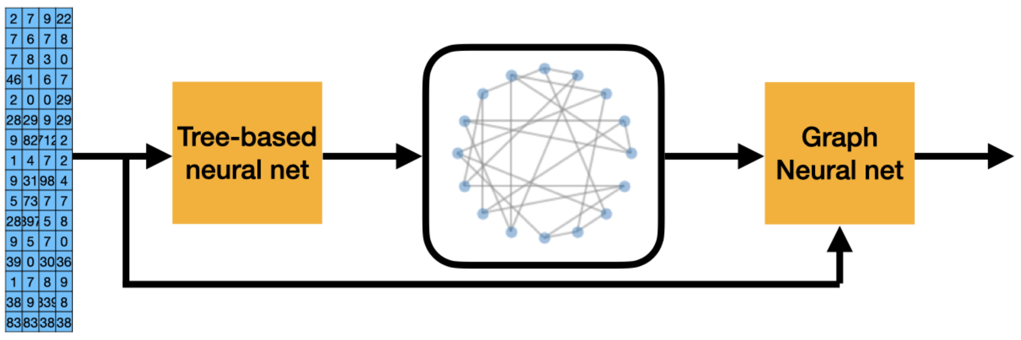

Figure 1: General architecture. In a first step, the data is used to generate a graph. In a second step, a graph is used as an input to a graph neural network.

Project description

Most nearest neighbor search algorithms [5]–[9] rely on a tree to split the samples into different areas and then reduce the computation time by only computing the distances between the points on the same leaf. This technique reduces the computational complexity from $\mathcal{O}\left(d^2\right)$ to $\mathcal{O}\left(d\log(d)\right)$. Inspired by this idea, the main goal of this task is to build a differentiable tree structure (similar to [10]) that allows us to dynamically build a sparse support for the graph edges. Practically, two parts need to be learned: an embedding function that projects the nodes into a discriminative space (often low-dimensional) and a tree splitting this space.

Similar ideas have been explored to scale attention, which is also $\mathcal{O}\left(n^2\right)$, in language models [11], [12]. Closer to our setup, [13] sort the vertices by an attention coefficient and then perform a 1-dimensional convolution. Thus, they performed computations on non-local edges efficiently. Instead of the sorting, we plan to learn a differentiable tree to group the samples that should be related.

Datasets We will start with a synthetic n-body dataset. The goal of the project will be to predict the gravitational displacement of n particles. Figure 1 illustrates the general architecture we will use. As a second task, we will work with a weather dataset and predict temperature and humidity. Eventually, if the time allows for it, we might attempt to scale the attention mechanism in an NLP problem.

Additional information

- **Difficulty of the project:** Very challenging!
- **What will you learn?** Graph neural networks, deep learning, optimization, doing research, PyTorch.
- **Requirements:** Strong motivation, machine Learning fundamentals, linear algebra, good Python knowledge, experience with git.
- **Weekly meetings** will be held to address questions, discuss progress and think about future ideas.

Contact

- Nathanaël Perraudin: nathanael.perraudin@sdsc.ethz.ch, SDSC
- Karolis Martinkus: martinkus@ethz.ch, ETZ G95

Bibliography

[1] K. Choromanski, V. Likhosherstov, D. Dohan, X. Song, A. Gane, T. Sarlos, P. Hawkins, J. Davis, A. Mohiuddin, L. Kaiser, *et al.*, “Rethinking attention with performers,” *arXiv preprint arXiv:2009.14794*, 2020.

[2] R. Child, S. Gray, A. Radford, and I. Sutskever, “Generating long sequences with sparse transformers,” *arXiv preprint arXiv:1904.10509*, 2019.

[3] N. Kitaev, Ł. Kaiser, and A. Levskaya, “Reformer: The efficient transformer,” *arXiv preprint arXiv:2001.04451*, 2020.

[4] A. Katharopoulos, A. Vyas, N. Pappas, and F. Fleuret, “Transformers are rnns: Fast autoregressive transformers with linear attention,” in *Proceedings of the International Conference on Machine Learning (ICML)*, 2020.

[5] J. H. Friedman, J. L. Bentley, and R. A. Finkel, “An algorithm for finding best matches in logarithmic expected time,” *ACM Transactions on Mathematical Software (TOMS)*, vol. 3, no. 3, pp. 209–226, 1977.

[6] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu, “An optimal algorithm for approximate nearest neighbor searching fixed dimensions,” *Journal of the ACM (JACM)*, vol. 45, no. 6, pp. 891–923, 1998.

[7] A. Andoni and P. Indyk, “Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions,” in *2006 47th annual IEEE symposium on foundations of computer science (FOCS’06)*, IEEE, 2006, pp. 459–468.

[8] M. Muja and D. G. Lowe, “Fast approximate nearest neighbors with automatic algorithm configuration.,” *VISAPP (1)*, vol. 2, no. 331-340, p. 2, 2009.

[9] ——, “Scalable nearest neighbor algorithms for high dimensional data,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 11, pp. 2227–2240, 2014.

[10] G. Farquhar, T. Rocktäschel, M. Igl, and S. Whiteson, “Treeqn and atreec: Differentiable tree-structured models for deep reinforcement learning,” in *International Conference on Learning Representations*, 2018.

[11] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, “Hierarchical attention networks for document classification,” in *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, 2016, pp. 1480–1489.

[12] G. Daras, N. Kitaev, A. Odena, and A. G. Dimakis, “Smyrf: Efficient attention using asymmetric clustering,” *arXiv preprint arXiv:2010.05315*, 2020.

[13] M. Liu, Z. Wang, and S. Ji, “Non-local graph neural networks,” *arXiv preprint arXiv:2005.14612*, 2020.