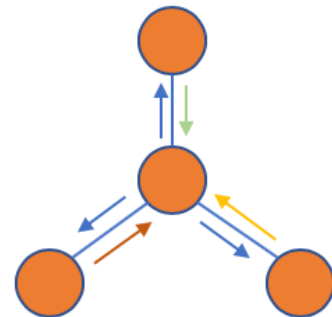




Agent-based Graph Neural Network

Lately, applying deep learning to graph-based problems has gained a lot of interest. Graph neural networks, in particular, achieved great success in knowledge graphs, reinforcement learning, chemistry, or physics simulations. These models are mostly based on the message passing framework, where nodes exchange messages over the edges and update their states based on all of the aggregated incoming messages. However, the current models generalize poorly to larger graphs and have trouble relaying long-range information over the graph. These problems have also been encountered in distributed computing and graph-based optimization problems before. Among the answers to these problems were the sub-linear randomized property testing algorithms, which allowed for probabilistic graph feature detection in time smaller than input size and ant colony optimization algorithms.



In this thesis we will aim to bring the performance boost offered by those random agent-based algorithms to graph neural networks. We will construct a novel graph neural network architecture, where a group of randomized agents act on the graph to come up with a solution to the task.

Requirements: Strong motivation, knowledge in deep learning, or a solid background in machine learning. Experience with Python and TensorFlow or PyTorch is an advantage as well as knowledge in graph theory, distributed computing and graph neural networks.

Interested? Please contact us for more details!

Contact

- Karolis Martinkus: martinkus@ethz.ch, ETZ G60.1

Detailed Project Outline

We denote the following primary tasks mandatory (on the right side you find a rough estimate for the time that we allocate to the respective task):

- Literature research (★)
- Implement baseline graph classification models (★)
- Construct toy datasets based on theoretical problems to evaluate the new model (★★)
- Implement a new agent based graph neural network. (★★★★)
- Investigate, how well this new model performs on different kinds of real-world graph classification tasks. (★)
- Write a report (★★)
- Present your findings. (★)

Extensions

Apart from these requirements, we can think of plenty of ways to extend the project:

- Investigate different agent output aggregation techniques.
- Determine the theoretical limitations of such a model.
- Apply the model to optimization problems.

The Student's Duties

- One meeting per week with the advisors to discuss current matters.
- Regular check-ins into the provided *revision control system*.
- A final report in English, presenting work and results.
- A final presentation (15 min) of the work and results obtained in the project.