Master Thesis Proposal

Student
Daniele Marchei
daniele19@ru.is

Supervisors
Yngvi Björnsson
Stephan Schiffel
{yngvi, stephans}@ru.is

School of Computer Science
Reykjavik University, Iceland

Abstract

Perfect information games can be represented as a graph, describing the rules that the players have to follow. In this thesis proposal, we explain how we will try to find some common patterns among games, using a *Message Passing Neural Network* to learn from these graph representation.

1 Introduction

1.1 General Game Playing

Introduction The focus of *General Game Playing* [2] (GGP) is to design an agent capable of playing any game by only knowing its rules. Agent of this kind should not be composed by a large number of sub-routines, once for each game, but they should be able to decide what actions to take during the game solely based on the set of rules given for it, ie, they should be as general as possible.
In this thesis, we will focus only on a specific type of games called *perfect information games*, in which there is no randomness involved and every player has the same information about the state of the game. Games such as *Chess*, *Tic-Tac-Toe* and *Go* are *perfect information games*.
The set of rules for a *perfect information game* can be encoded with a formal language called *Game Description Language* (GDL) [6].

Graph representation of games Every GDL rules can be represented as a graph. Some of them are called *Rule Graphs*, *Propositional Networks* and *Cell*

Automata [7, 8]. Each of them has its pros and cons, such as the size of the graph and computational complexity.

1.2 Message Passing Neural Network

Introduced in [3], *Message Passing Neural Networks* (MPNN) are a way to perform Machine Learning on graph-structured data. As the name suggests, the core of this model is to learn some vector representation for each node of the graph using a *message passing function*. At each iteration, each node tweaks its hidden state based on the hidden states of its neighbours using a *state update function*. After a fixed amount of iterations, each node will have its own vector representation. At this point, a *readout function* can be used to aggregate these representations and get a single vector as an output. This output vector will be a compressed representation for the whole graph.

2 Thesis Statement

In this thesis, we will try to see if an MPNN can learn something useful from a dataset of games. We currently have about 230 games in *GDL* format, but some data augmentation is possible. These games will be converted in graphs and fed into an MPNN. The main areas of interest are:

- **Clustering**, to seek for common properties among games
- **Classification** of the best strategy to adopt for a specific game
- Using the vector representation as **additional info for a GGP Agent**, maybe a GGP Agent will have better capabilities of evaluating the current game state or to predict the best set of actions to perform if it has a succinct representation of the game rules

A sketch of an hypotetical pipeline is represented in Figure 1.

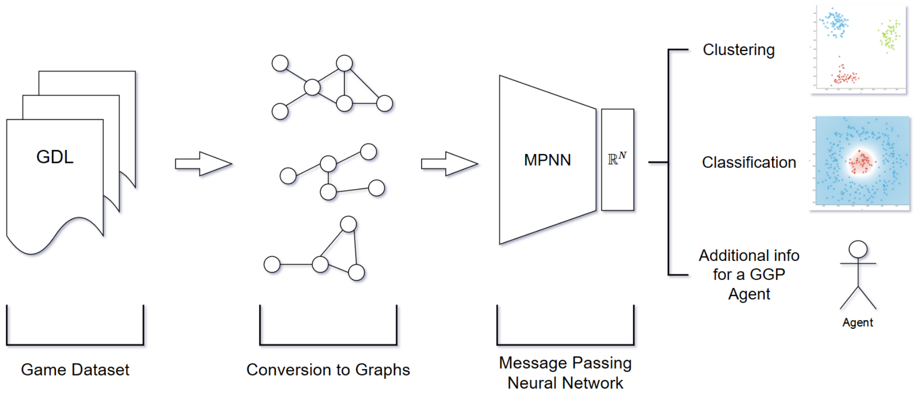

Figure 1: For each game in the dataset (left) we extract its graph representation (center). Each graph is then fed into a MPNN and converted in a vector of dimension N (right). This vector is then used for *Clustering*, *Classification* or used as *additional info for a GGP Agent*.

3 Preliminary results

As a proof of concept, we implemented in PyTorch a Graph Auto Encoder (a particular type of MPNN) to map a dataset of enzymes to a compressed vector representation [1, 5, 4]. With this new latent space, we then used a clustering algorithm to see if we could recover the six types of enzymes present in the dataset. Using a very basic readout function, we managed to predict five out of six clusters.

References

[1] Matthias Fey and Jan E. Lenssen. "Fast Graph Representation Learning with PyTorch Geometric". In: *ICLR Workshop on Representation Learning on Graphs and Manifolds*. 2019.

[2] *General Game Playing*. `http://ggp.stanford.edu/`.

[3] Justin Gilmer et al. *Neural Message Passing for Quantum Chemistry*. 2017. arXiv: `1704.01212 [cs.LG]`.

[4] Kristian Kersting et al. *Benchmark Data Sets for Graph Kernels*. 2016. URL: `http://graphkernels.cs.tu-dortmund.de`.

[5] Thomas N. Kipf and Max Welling. *Variational Graph Auto-Encoders*. 2016. arXiv: `1611.07308 [stat.ML]`.

[6] Nathaniel Love et al. *General Game Playing: Game Description Language Specification*. Standford Logic Group. 2008.

[7] Stephan Schiffel. *Symmetry Detection in General Game Playing*. Association for the Advancement of Artificial Intelligence. 2010. URL: `www.aaai.org`.

[8] Eric Schkufza, Nathaniel Love, and Michael Genesereth. "Propositional Automata and Cell Automata: Representational Frameworks for Discrete Dynamic Systems". In: *AI 2008: Advances in Artificial Intelligence*. Ed. by Wayne Wobcke and Mengjie Zhang. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 56–66. ISBN: 978-3-540-89378-3.

[9] David Silver et al. *Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm*. 2017. arXiv: `1712.01815 [cs.AI]`.

[10] Jie Zhou et al. *Graph Neural Networks: A Review of Methods and Applications*. 2018. arXiv: `1812.08434 [cs.LG]`.