Graph Element Networks: adaptive, structured computation and memory

Ferran Alet [1] **Adarsh K. Jeewajee** [1] **Maria Bauza** [2]
Alberto Rodriguez [2] **Tomás Lozano-Pérez** [1] **Leslie Pack Kaelbling** [1]

Abstract

We explore the use of graph neural networks (GNNs) to model spatial processes in which there is no *a priori* graphical structure. Similar to *finite element analysis*, we assign nodes of a GNN to spatial locations and use a computational process defined on the graph to model the relationship between an initial function defined over a space and a resulting function in the same space. We use GNNs as a computational substrate, and show that the locations of the nodes in space as well as their connectivity can be optimized to focus on the most complex parts of the space. Moreover, this representational strategy allows the learned input-output relationship to generalize over the size of the underlying space and run the same model at different levels of precision, trading computation for accuracy. We demonstrate this method on a traditional PDE problem, a physical prediction problem from robotics, and learning to predict scene images from novel viewpoints.

1. Introduction

A great deal of success in deep learning has come from finding appropriate structural inductive biases to impose on network architectures. For an architectural assumption to be useful, it has to exploit a structural property that is (approximately) satisfied in a broad set of tasks. For instance, convolutional neural networks exploit the locality and spatial invariance found in many computer vision problems. Similarly, graph neural networks (GNNs) exploit underlying relational structure, which makes them a good fit for tasks consisting of a set of entities with pairwise interactions.

Traditional applications of GNNs assume an *a priori* notion of entity (such as bodies, links or particles) and match every node in the graph to an entity. We propose to apply GNNs to the problem of modeling transformations of functions defined on continuous spaces, using a structure we call *graph element networks* (GENs). Inspired by finite element methods, we use graph neural networks to mesh a continuous space and define an iterative computation that propagates information from some sampled input values in the space to an output function defined everywhere in the space. GENs allow us to model systems that have spatial structure but lack a clear notion of entity, such as the spatial temperature function in a room or the dynamics of an arbitrarily-shaped object being pushed by a robot.

Finite element methods (Hughes, 2012) are used to numerically solve partial differential equations (PDEs) by dividing a space into small subparts (elements) where the PDE can be approximated. Analogously, in a GEN, we place nodes of a GNN in the space, allowing each node to model the local state of the system, and establish a connectivity graph among the nodes (it may be a regular mesh or any other arbitrary graph). Then, input values are specified for multiple spatial locations, and used to initialize the states of the GNN nodes, which are then propagated according to the GNN's update functions. Finally, output values may be read back from any point in the system, by interpolating between values of nodes in the GNN. A critical aspect of the GEN is that although the model has a fixed set of weights, it can be used to model small or large spaces, with coarse, fine or variable-resolution sampling of the space by reusing the same weights in different nodes and edges.

Although GENs were originally inspired by finite element methods that model PDEs, we show that they are much more widely applicable. We are able to model pushing of arbitrarily shaped objects, whose discrete dynamics do not follow a PDE, with the same structure. Furthermore, we apply GENs to the task of scene representation, where the system is presented with images of a scene from various camera poses and is asked to predict the view from an unseen pose. We show that GENs naturally adapt to scenes of increasing size without further training.

In the following sections, we discuss related work, define the graph element network model and provide an algorithm

[1]CSAIL - MIT, Cambridge, MA, USA [2]Mechanical Engineering - MIT, Cambridge, MA, USA. Correspondence to: Ferran Alet <alet@mit.edu>.

Proceedings of the 36th International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.