

Stability and Generalization Capabilities of Message Passing Graph Neural Networks

Sohir Maskey
LMU Munich
maskey@math.lmu.de

Ron Levie
Technion -
Israel Institute of Technology
levieron@technion.ac.il

Yunseok Lee
LMU Munich
ylee@math.lmu.de

Gitta Kutyniok
LMU Munich
University of Tromsø
kutyniok@math.lmu.de

Abstract—Message Passing Neural Networks (MPNNs) are considered state-of-the-art tools for solving a large variety of graph-focused problems. We study the generalization capabilities of MPNNs in graph classification. We assume that graphs of different classes are sampled from different random graph models. Based on this data distribution, we derive a non-asymptotic bound on the generalization gap, that decreases to zero as the graphs become larger. This is proven by showing that a MPNN, applied on a graph, approximates the MPNN applied on the geometric model that the graph discretizes.

I. INTRODUCTION

In many important fields, such as chemistry, social networks, or drug design, data can be described by graphs. This has led to a tremendous interest in the development of machine learning models for graph-structured data in recent years. A ubiquitous tool for processing such data are graph convolutional neural networks (GCNNs), which extend standard convolutional neural networks (CNNs) to graph-structured data.

Most GCNNs used in practice can be described using the general architecture of *Message Passing Neural Networks (MPNNs)*. MPNNs generalize the convolution operator to graph domains by a neighborhood aggregation or message passing scheme. By $\mathbf{f}_i^{(t-1)}$ denoting the feature of node i in layer $t-1$ and $\mathbf{e}_{j,i}$ denoting edge features from node j to i , one layer in a message passing graph neural networks is given by

$$\mathbf{f}_i^{(t)} = \Psi^{(t)}\left(\mathbf{f}_i^{(t-1)}, \mathbf{AGG}\left\{\Phi^{(t)}\left(\mathbf{f}_i^{(t-1)}, \mathbf{f}_j^{(t-1)}, \mathbf{e}_{j,i}\right)\right\}_{j \in \mathcal{N}(i)}\right), \quad (1)$$

where $\mathcal{N}(i)$ is the set of nodes connected to node i , \mathbf{AGG} denotes a differentiable and permutation invariant function, e.g., sum, mean, or max, and $\Psi^{(t)}$ and $\Phi^{(t)}$ denote differentiable functions such as MLPs (Multi-Layer Perceptrons) [FL19].

In this paper we study the generalization capabilities of MPNNs in a graph classification task. We are given pairs of graphs and graph signals $\mathbf{x} = (G, \mathbf{f})$ and a target output \mathbf{y} , where (\mathbf{x}, \mathbf{y}) are jointly drawn from a distribution $p(x, y)$. The goal of the MPNN Θ is to approximate \mathbf{y} by $\Theta(\mathbf{x})$. For this, one uses a loss function \mathcal{L} , which measures the discrepancy between the true label \mathbf{y} and the output of the MPNN $\Theta(\mathbf{x})$. The aim of a machine learning algorithm is to minimize the statistical loss (also called expected loss)

$$R_{exp}(\Theta) = \mathbb{E}_{(x, y) \sim p} [\mathcal{L}(\Theta(x), y)].$$

In (data-driven) machine learning one has only access to a training set instead of knowing the distribution p . Namely, we consider a multi-graph setting, where the training set $\mathcal{T} = (\mathbf{x}^i = (G^i, \mathbf{f}^i), \mathbf{y}^i)_{i=1}^m$ is a collection of m samples drawn i.i.d. from the distribution $p(x, y)$. Then, instead of minimizing the statistical loss,

one minimizes the empirical loss, given by

$$R_{emp}(\Theta) = \frac{1}{m} \sum_{i=1}^m \mathcal{L}(\Theta(\mathbf{x}^i), \mathbf{y}^i).$$

The *generalization error* is then defined by

$$GE(\Theta) = |R_{exp}(\Theta) - R_{emp}(\Theta)|. \quad (2)$$

In deep learning on Euclidean data, various measures, such as the VC-dimension [Vap99] or the Rademacher complexity [BM03], have been used to bound the generalization error. Those complexity measures were recently used to provide generalization bounds for graph neural networks [STH18] and for 2-layered MPNNs in binary graph classification tasks [GJJ20]. Furthermore, [VZI19] consider generalization abilities of single-layer spectral GCNNs for node-classification tasks and provide a bound for the generalization error that is directly proportional to the largest eigenvalue of the graph Laplacian.

Closely related to good generalization abilities is *stability* (or *transferability*) of GCNNs: if two graphs represent the same phenomenon, the network has similar repercussions on those. In recent years there have been multiple works showing stability of graph neural networks in different frameworks (e.g. [LHB⁺21], [RWR21], [KTD21], [MLK21]). The work closest related to our stability results is [KBV20], where the authors show that spectral-based methods are pointwise stable under graphs that approximate the same infinite-node limit.

In this work, we show that MPNNs with mean aggregation, which include spectral-based methods, are pointwise stable under graphs sampled from the same infinite-node limit with similar asymptotics as in [KBV20]. Furthermore, we present, up to our knowledge, the first uniform stability results, which we leverage to prove a novel upper bound on the generalization error for MPNNs. This upper bound decays not only with respect to the number of training examples, but also with the average number of nodes in the graphs. We elaborate on these contributions in the following section.

II. MAIN RESULTS

We consider graphs as discretizations of continuous spaces in our analysis, called random graph models (RGM). Formally, a RGM on a metric-measure space (χ, d, μ) is defined as a tuple (W, f) of a kernel $W : \chi \times \chi \rightarrow \mathbb{R}$ and a metric-space signal $f : \chi \rightarrow \mathbb{R}^F$. The RGM (W, f) is then a generative model for a random graph (G, \mathbf{f}) with corresponding graph features by sampling N i.i.d. points X_1, \dots, X_N from μ on χ . The weight matrix $W = (w_{i,j})_{i,j}$ of G is defined by $w_{i,j} = W(X_i, X_j)$ and the graph signal $\mathbf{f} \in \mathbb{R}^{N \times F}$ is defined by $\mathbf{f}_i = f(X_i)$.

We then define continuous message passing neural networks (cMPNNs) that act on RGMs (W, f) , by replacing the graph node

