Generalization Error of Graph Neural Networks in the Mean-field Regime

Gholamali Aminian* [1], Yixuan He* [2], Gesine Reinert[1,2], Łukasz Szpruch [1,3], Samuel N. Cohen[1,2]
* Equal contribution
[1] The Alan Turing Institute [2] University of Oxford
[3] The University of Edinburgh

July 21, 2024

Objectives

- A novel framework for exploring the generalization errors of Graph Neural Networks, i.e., Graph Convolutional Neural (GCN) and Message Passing Graph Neural Networks (MPGNN), through functional derivative and Rademacher Complexity analyses in Mean-field regime

Objectives

- A novel framework for exploring the generalization errors of Graph Neural Networks, i.e., Graph Convolutional Neural (GCN) and Message Passing Graph Neural Networks (MPGNN), through functional derivative and Rademacher Complexity analyses in Mean-field regime
- The generalization error convergence rate, when training on a sample of size n, is $\mathcal{O}(1/n)$ for **KL-regularized empirical risk minimization problem** via functional derivative

Objectives

- A novel framework for exploring the generalization errors of Graph Neural Networks, i.e., Graph Convolutional Neural (GCN) and Message Passing Graph Neural Networks (MPGNN), through functional derivative and Rademacher Complexity analyses in Mean-field regime
- The generalization error convergence rate, when training on a sample of size n, is $\mathcal{O}(1/n)$ for **KL-regularized empirical risk minimization problem** via functional derivative
- Investigating the generalization error of one-hidden-layer graph neural network for the effect of hidden neurons.

Problem Formulation

One-hidden-layer Graph Convolutional Networks (GCNs)

- $X \in \mathcal{X}$ graph sample as input and $Y \in \mathcal{Y} = \{-1, 1\}$, binary classification.
- $(X, Y) = Z \in \mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ and $Z \sim \mu$.
- Training dataset, $\mathbb{Z}_N = \{Z_i\}_{i=1}^n$ with **i.i.d. assumption,**
- **Empirical measure**, $\mu_n := \frac{1}{n} \sum_{i=1}^n \delta_{Z_i}$.
- Interested in learning a function $f_W : \mathcal{X} \to \mathcal{Y}$ parameterized via a number of parameters from $\mathcal{W}$.
- **Learning algorithm**: $\mu_n \mapsto \mathfrak{m}(\mu_n) \in \mathcal{P}(\mathcal{W})$ outputs a probability distribution (measure) on parameter space.
- loss function $(m, z) \mapsto \ell(m, z) \in \mathbb{R}^+$.
- **Risk function:** $\mathrm{R}(m, \mu) := \int_{\mathcal{Z}} \ell\big(m, z\big)\mu(\mathrm{d}z)$.
- **Empirical risk:** $\mathrm{R}(m, \mu_n) = \int_{\mathcal{Z}} \ell\big(m, z\big)\mu_n(\mathrm{d}z) = \frac{1}{n} \sum_{i=1}^n \ell\big(m, z_i\big)$.
- **KL divergence:** $\mathrm{KL}(m' \| m)$.
- **Symmetrized KL divergence:** $\mathrm{KL}_{\mathrm{sym}}(m \| m') = \mathrm{KL}(m \| m') + \mathrm{KL}(m' \| m)$.

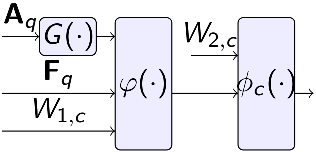


Figure: Neuron Unit

- **Input pair** of a graph sample with N nodes: $\mathbf{X} = (\mathbf{F}, \mathbf{A}) \in \mathcal{X}$ where $\mathbf{F}$ is nodes feature matrix and $\mathbf{A}$ is graph adjacency matrix, $d_{\max}$ and $d_{\min}$: maximum and minimum node degrees among all graph samples.
- **Parameters:** $W_{1,c}$ the parameters of each neuron, where $W_{1,c} \in \mathcal{S}^k \subset \mathbb{R}^k$ and $W_{2,c} \in \mathcal{S} \subset \mathbb{R}$.
- **Activation Function:** $\varphi(W_{1,c} \cdot x)$.
- **Neuron Unit:** $\phi(W_c, x) = W_{2,c}(i)\varphi(W_{1,c}(i) \cdot X(j))$.
- **Number of Neuron Units:** h.

- **Empirical parameter measure:** $m_h := \frac{1}{h}\sum_{i=1}^{h}\delta_{(W_{1,c}(i),W_{2,c}(i))}$.
- **Readout function:** Mean-readout
  $\Psi(m_h^c(\mu_n),\mathbf{X}) := \frac{1}{N}\sum_{j=1}^{N}\mathbb{E}_{W_c\sim m_h^c(\mu_n)}\big[\phi_c(W_c, G(\mathbf{A})[j,:]\mathbf{F})\big]$.
- **Prediction:** $\hat{y} := \hat{f}(\mathbf{X}) = \frac{1}{h}\sum_{i=1}^{h}\phi_c(W_c(i),\mathbf{X}) =$
  $\int \phi(W_c,\mathbf{X})\, m_h(\mathrm{d}W_c) = \mathbb{E}_{W_c\sim m_h}[\phi(W_c,\mathbf{X})]$.
- **Mean-field:** $h\to\infty$ then $m_h(\mu_n)\to m(\mu_n)$.
- **Logistic loss:** $\ell(\Psi(m(\mu_n),\mathbf{x}),y) = \log(1+\exp(-\Psi(m(\mu_n),\mathbf{x})y))$.

Generalization Error

$$\mathrm{R}(\mathfrak{m}(\mu_n),\mu) = \underbrace{\Big(\mathrm{R}(\mathfrak{m}(\mu_n),\mu) - \mathrm{R}(\mathfrak{m}(\mu_n),\mu_n)\Big)}_{\text{generalization error}} + \underbrace{\mathrm{R}(\mathfrak{m}(\mu_n),\mu_n)}_{\text{training error}}.$$

- Expected Generalization Error:
  $\overline{\mathrm{gen}}(\mathfrak{m},\mu) \triangleq \mathbb{E}_{Z_n}\big[\mathrm{R}(\mathfrak{m}(\mu_n),\mu) - \mathrm{R}(\mathfrak{m}(\mu_n),\mu_n)\big].$
- **Replace-one sample empirical measure**:
  $\mu_{n,(1)} = \mu_n + \frac{1}{n}(\delta_{\bar{Z}_1} - \delta_{Z_1})$, where $\bar{Z}_1$ is i.i.d. with respect to $\mathbb{Z}_{\mathbb{N}}$
- (Aminian et al, 2023): $\overline{\mathrm{gen}}(\mathfrak{m},\mu) =$
  $\mathbb{E}_{\mathfrak{Z}_n,\bar{Z}_1}\Big[\ell\big(\mathfrak{m}(\mu_n),\bar{Z}_1\big) - \ell\big(\mathfrak{m}(\mu_{n,(1)}),\bar{Z}_1\big)\Big].$

KL-regularized Empirical risk minimization

KL-Regularized Problem

- **Setup:** $\mathcal{V}^{\alpha}(m,\mu) = \mathrm{R}(m,\mu) + \frac{1}{\alpha}\mathrm{KL}(m\|\pi)$
  - $R(m,\mu) = \mathbb{E}_{Z\sim\mu}[\ell(m,z)]$,
  - Prior: $\pi(w)$,
  - α: inverse temperature

KL-regularized Empirical risk minimization

KL-Regularized Problem

- **Setup:** $\mathcal{V}^\alpha(m,\mu) = \mathrm{R}(m,\mu) + \frac{1}{\alpha}\mathrm{KL}(m\|\pi)$
  - $R(m,\mu) = \mathbb{E}_{Z\sim\mu}[\ell(m,z)]$,
  - Prior: $\pi(w)$,
  - α: inverse temperature
- **Solution (Gibbs Measure):**
  $\mathrm{m}^\alpha(\mu_n) := \frac{\pi}{S_{\alpha,\pi}(\mu_n)} \exp\Big\{ -\alpha\Big[\frac{\delta\mathrm{R}}{\delta m}(m,\mu_n,w)\Big]\Big\}$,
  where $S_{\alpha,\pi}(\mu_n)$ is is the normalizing constant.

Assumptions

1. Bounded loss function, $0 \leq \ell(\hat{y}, y) \leq M_\ell$.
2. Bounded gradient of loss function, $|\partial_{\hat{y}} \ell(\hat{y}, y)| \leq M_{\ell'}$.
3. Convex loss function, $\ell(\hat{y}, y)$ with respect to $\hat{y}$.
4. Bounded Neuron unit function, $|\phi_c(.,.)| \leq M_\phi$.
5. Bounded node features, $\|F[i, :]\| \leq B_f$.

Main Results

Upper Bound

There exists constant C, such that

$$\overline{\mathrm{gen}}(\mathrm{m}(\mu_n), \mu) \leq C\mathbb{E}_{\mathbf{Z}_n, \bar{Z}_1}\left[\sqrt{\mathrm{KL}\big(\mathrm{m}(\mu_n)\|\mathrm{m}(\mu_{n,(1)})\big)}\right],$$

Lower Bound

For Gibbs measure, we have,

$$\overline{\mathrm{gen}}(\mathrm{m}^{\alpha}(\mu_n), \mu) \geq \frac{n}{2\alpha}\mathbb{E}_{\mathbf{Z}_n, \bar{Z}_1}\left[\mathrm{KL}_{\mathrm{sym}}\big(\mathrm{m}^{\alpha}(\mu_n)\|\mathrm{m}^{\alpha}(\mu_{n,(1)})\big)\right].$$

Theorem

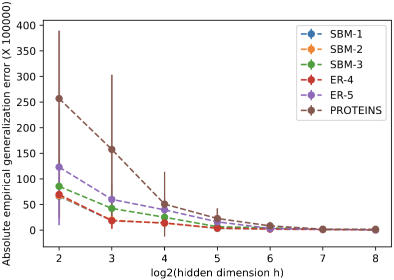
There exists constant C, such that $\overline{\mathrm{gen}}(\mathrm{m}^{\alpha}(\mu_n), \mu) \leq \dfrac{\alpha C}{n}$.

Comparison to previous works

Approach	$\tilde{d}_{\max}, \tilde{d}_{\min}$	Width of GCN (h)	Number of graph samples (n)	Bound Type
VC-Dimension [STH18]	N/A	$O(h^4)$	$O(1/\sqrt{n})$	HP
Rademacher Complexity [GJJ20]	$O(\tilde{d}_{\max} \log^{1/2}(\tilde{d}_{\max}))$	$O(h\sqrt{\log(h)})$	$O(1/\sqrt{n})$	HP
PAC-Bayesian [LUZ20]	$O(\tilde{d}_{\max})$	$O(\sqrt{h \log(h)})$	$O(1/\sqrt{n})$	HP
PAC-Bayesian [JLSZ23]	N/A	$O(\sqrt{h})$	$O(1/\sqrt{n})$	HP
Continuous MPGNN [MLLK22]	N/A	N/A	$O(1/\sqrt{n})$	P
Rademacher Complexity (this paper)	$O((\tilde{d}_{\max}/\tilde{d}_{\min})^{3/4})$	N/A	$O(1/\sqrt{n})$	HP
Functional Derivative (this paper)	$O(\tilde{d}_{\max}/\tilde{d}_{\min})$	**N/A**	$O(1/n)$	E

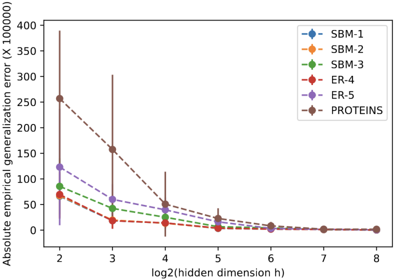
Table: Comparison of generalization bounds. The width of the hidden layer and the number of training samples are denoted as h and n, respectively. "N/A" means not applicable.

Experiments



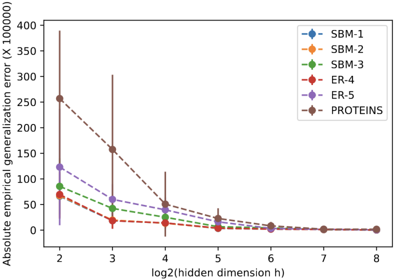
- We investigate the effect of the number of hidden neurons (number of hidden units) h on the true generalization error of GCNs

Experiments



- We investigate the effect of the number of hidden neurons (number of hidden units) h on the true generalization error of GCNs
- Stochastic Block Models (SBMs), Erdos-Rényi (ER) models, and PROTEINS dataset

Experiments



- We investigate the effect of the number of hidden neurons (number of hidden units) h on the true generalization error of GCNs
- Stochastic Block Models (SBMs), Erdos-Rényi (ER) models, and PROTEINS dataset
- As the value of h increases, the absolute generalization error decreases. This observation shows that the upper bounds dependent on the width of the layer fail to capture the trend of generalization error in the over-parameterized regime.

Thank You

Figure: ArXiv Link

- ArXiv ID: 2402.07025
- Code repo: `https://github.com/SherylHYX/GNN_MF_GE`
- Email: `gaminian@turing.ac.uk`
- Acknowledgements:
    - The UKRI Prosperity Partnership Scheme (FAIR)
    - EPSRC Grants EP/V056883/1, EP/W037211/1 and EP/R018472/1
    - The Alan Turing Institute and G-Research

References I

Vikas Garg, Stefanie Jegelka, and Tommi Jaakkola, *Generalization and representational limits of graph neural networks*, International Conference on Machine Learning, PMLR, 2020, pp. 3419–3430.

Haotian Ju, Dongyue Li, Aneesh Sharma, and Hongyang R Zhang, *Generalization in graph neural networks: Improved pac-bayesian bounds on graph diffusion*, arXiv preprint arXiv:2302.04451 (2023).

Renjie Liao, Raquel Urtasun, and Richard Zemel, *A pac-bayesian approach to generalization bounds for graph neural networks*, International Conference on Learning Representations, 2020.

Sohir Maskey, Ron Levie, Yunseok Lee, and Gitta Kutyniok, *Generalization analysis of message passing neural networks on large random graphs*, Advances in Neural Information Processing Systems, 2022.

References II

Franco Scarselli, Ah Chung Tsoi, and Markus Hagenbuchner, *The vapnik–chervonenkis dimension of graph and recursive neural networks*, Neural Networks **108** (2018), 248–259.