그래프 신경망의 확률적 경사 소음의 통계적 분석

이정현, 정민찬, 허남규, 윤세영

김재철 AI 대학원, KAIST, 서울, 대한민국

{jh_lee00,mcjeong,itsnamgyu,yunseyoung}@kaist.ac.kr

A Statistical Analysis of Stochastic Gradient Noises for GNNs

Junghyun Lee, Minchan Jeong, Namgyu Ho, Se-Young Yun

Kim Jaechul Graduate School of AI, KAIST, Seoul, Republic of Korea

요 약

확률적 경사하강법 (stochastic gradient descent, SGD)은 대규모 데이터에 대한 심층학습을 가능케 한 결정적인 기법으로, 지금까지 영상, 음성 등 모든 양식의 최첨단 심층학습 모델에 사용되고 있다. 이에 따라, SGD의 특성을 분석하기 위한 다양한 이론적인 연구가 진행되고 있다. 특히 심층학습 모델의 확률적 경사소음 (stochastic gradient noise, SGN)을 확률 과정으로 모델링하며, SGN의 실험적인 분포를 토대로 SGD에 대응되는 확률과정의 특성을 분석하는 방향이 있다. 본 논문에서는 그래프 신경망 (graph neural network, GNN)에서 SGN의 통계적인 특성을 분석하고자 한다. GNN은 일반적인 신경망 (neural network, NN)과 달리, 노드의 차수 (degree) 분포와 같이 그래프 데이터가 가지는 고유한 특성이 존재하기 때문에, 이와 같은 특성이 SGN의 양상에 영향을 미칠 것으로 예상한다. 본 논문에서는 흔히 쓰이는 Cora 벤치마크에서 실험을 통해 GNN 위에서 SGN이 어떤 분포를 따르는지 파악한다. 또한, 다양한 분석을 통해 이를 직관적으로 설명하고, 이를 토대로 향후 GNN 연구 및 실무를 돕기 위한 통찰을 제공한다.

1. Introduction

1.1. Stochastic Optimization

Most of the key tasks in machine/deep learning can be expressed as the following optimization problem:

$$\min_w F(w) \triangleq \min_w \frac{1}{n}\sum_{i=1}^{n} f^{(i)}(w) \qquad (1)$$

where $f^{(i)}$ is the individual loss contributed by each data point $\mathrm{x_i}$. Such optimization is usually solved via some gradient-based method, and one such (first-order) algorithm is gradient descent (GD). But in recent years we are seeing an explosion in the number of datas (so-called "big-data era"), and thus a naïve GD is too computationally inefficient. Thus, its stochastic version, called the stochastic gradient descent (SGD), is widely used. (see [1], for instance, for a comprehensive review)

1.2. Modeling SGN

Other than its computational efficiency, it has been empirically confirmed that the noises of SGD actually contribute towards better generalization capability of the resulting model. From a theoretical perspective, this is still a mystery, and numerous methodologies have been employed to shed light on this rather surprising phenomenon.

One method of analysis is regarding SGD as discretization of some continuous stochastic differential equation (SDE), driven by some stochastic process. Such continuous analysis allows one to utilize tools from a rich literature of statistical physics, probability theory, and more to elucidate the unreasonable efficiency and efficacy of SGD. To see this, we first rewrite SGD as:

$$\mathrm{w_{t+1}} = \mathrm{w_t} - \eta\nabla\widetilde{\mathrm{f_t}}(\mathrm{w_t}) = \mathrm{w_t} - \eta\nabla\mathrm{F}(\mathrm{w_t}) + \eta\mathrm{U_t} \qquad (2)$$

where $\nabla\widetilde{\mathrm{f_t}}(\mathrm{w}) = (1/|\Omega_t|)\sum_{i\in\Omega_t}\nabla f^{(i)}(w)$ is the stochastic gradient at iteration t, and $\mathrm{U_t}(\mathrm{w}) = \nabla\mathrm{F}(\mathrm{w}) - \nabla\widetilde{\mathrm{f_k}}(\mathrm{w})$ is the stochastic gradient noise (SGN). Note that each fixed iteration t, $\mathrm{U_t}$ is a random vector of zero mean.

Depending on the empirical observation and/or modeling assumption, one can either choose to model $\mathrm{U_t}$ as normal or heavy-tailed. The distinction comes from whether we assume that the (co)variance of SGNs is finite or infinite. Depending on the modeling assumption, the behavior of the SDE differs greatly, and so does the theoretical analysis of the SGD [2]; especially the explanation of why SGD favors flat minima becomes vastly difference. [2,6] argued for the heavy-tailed theory of SGN, invoking several results from Levy-driven dynamical systems in statistical physics literatures [3]. From

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2019-0-00075, Artificial Intelligence Graduate School Program(KAIST), 10%) and the Institute of Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No. 2022-0-00871, Development of AI Autonomy and Knowledge Enhancement for AI Agent Collaboration, 90%).

a rather different perspective, [4] argued for *multivariate*, state-dependent Gaussian theory of SGN, invoking diffusion theory from statistical physics.

However, their justifications are too simple and heuristic. Indeed, all the papers mentioned above only provided qualitative observations, namely the histograms of SGNs. Especially, [5] criticized the heavy-tailed theory in that 1. the alpha estimator used is true *only when* the distribution is known to be stable alpha distributed, and 2. the assumption that SGNs are coordinate-wise i.i.d is invalid, as the training procedure somehow entangles the individual coordinates with one another. However, even in [5], the authors did not perform rigorous statistical tests other than the normality tests, and it isn't clear whether the normality test deployed considers the multivariate nature of the SGNs. Also, [4] did not provide any results, not even qualitative, for the structure of SGN in mid-training phase; they only considered the initial iteration.

Recently, [6] provided the first precise statistical analysis into SGN of several old/modern deep architectures, and the authors found that the SGNs are actually best described using lognormal distribution. Arguably, their methodology is the most scientific way of exploring the suitable assumptions for SGNs, and thus we also employ it in this work.

1.3. (Stochastic) Optimization of GNN

Despite the abundance of literature in analyzing/modeling SGD as SDE, not much work has been done that analyzes the stochastic optimization trajectory of GNN training, let alone even deterministic optimization algorithms. [7,8] considers the convergence analysis of gradient descent for GNNs; they do not consider any stochastic variants.

We are the *first* to consider the nature of stochastic optimization of GNNs. Precisely, we provide a preliminary answer to the following question: *What are the statistical properties of SGNs stochastic training of GNNs?*

2. Problem Settings

2.1. Semi-Supervised Node Classification

We follow discussions of [9]. Given a graph $\mathrm{G} = (\mathrm{V}, \mathrm{E})$, each $\mathrm{v} \in \mathrm{V}$ is given a feature $x_v \in \mathbb{R}^d$. Let $\mathrm{S} \subseteq \mathrm{V}$ be the labeled vertices, which consists our training set: $\{(x_v, y_v)\}_{v \in S}$. The goal of semi-supervised *node classification* is to predict the labels of the remaining nodes in $\mathrm{V} \backslash \mathrm{S}$.

2.2. Graph Neural Networks (GNNs)

We consider two types of GNN; graph convolutional network (GCN; [9]) and graph isomorphism network (GIN; [10]), both of which is one of the most popular GNN architectures. Their main difference lies in the aggregator function that they use. GCN makes use of the (normalized) average aggregator while GIN makes use of the sum aggregator.

2.3. Node Sampling and SGD for GNNs

To enable efficient computations, sampling is employed. There are two main methods of sampling: node batching and neighborhood sampling. For clarity, we only consider uniformly random node batching and always make use of the entire neighbors. Indeed, it is an interesting and important future direction to see the effect of neighborhood sampling and/or non-uniform sampling (e.g. importance sampling), as such techniques are known to induce variance reduction and even faster convergence [11,12].

3. Experimental Settings

3.1. Downstream task, Architectures

We consider the node classification task on the Cora dataset. Cross entropy loss is used, and we use GCN/GIN with three message passing layers and ReLU nonlinearity.

3.2. Statistical tests

Every certain epoch, we sample 1000 random batches, compute and store the norms of the corresponding SGN vectors. For simplicity, we show here the results for only initial and final phases of training.

We employ QQ-plots that compares the collected SGNs to some predefined distribution; the considered distributions are normal, lognormal, Pareto (power law), exponential, and stretched exponential (Weibull). We use Python's powerlaw package [13] to ensure the best possible fit for the last four distributions. Using other statistical tests such as empirical mean residual life, or even developing suitable multivariate distribution testing is left for future work.

4. Results

The results for GCN and GIN are shown in 그림 1,2, respectively. Each 그림 has two rows, one for the initial phase and one for the final phase. Several observations can be made; firstly, for all considered situations, normal and Pareto (power law) distributions do not have good fit, while lognormal distribution has somewhat good fit overall. This is in line with the observations made in [6] for image classification tasks.

Overall, there are more than one distribution that seems to provide good fit for each situation (ex. initial phase of GIN seems to be well-fitted by chi-squared, lognormal, and Weibull). One can also observe that the chi-squared fitting establishes a distinction between the SGN of GCN and GIN; the latter very well fitted, while former is not! Lastly, if chi-squared is the best distribution, it suggests that the SGN

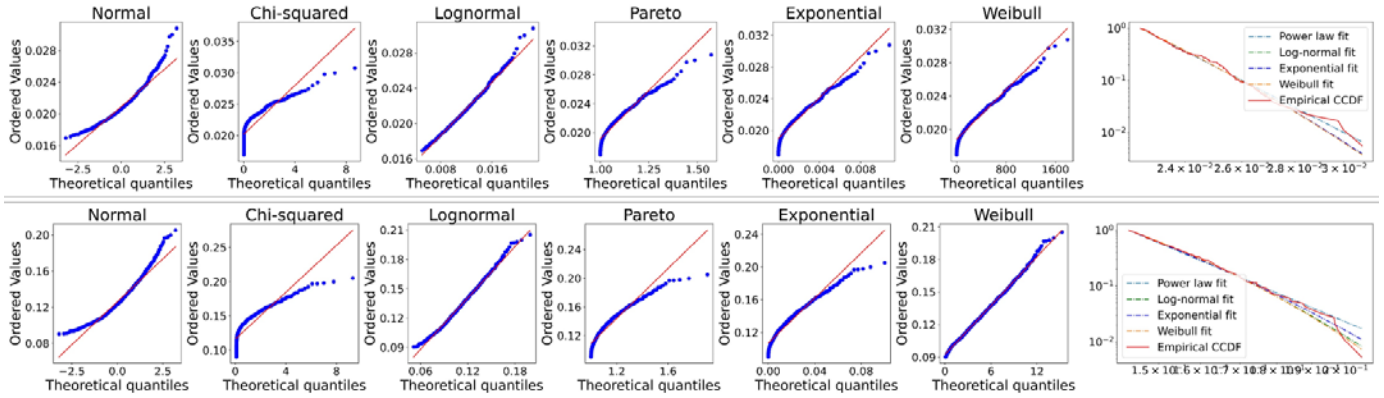

그림 1 QQ-plots of SGN of GCN training. Top row is for the initial phase and bottom row is for the final phase.

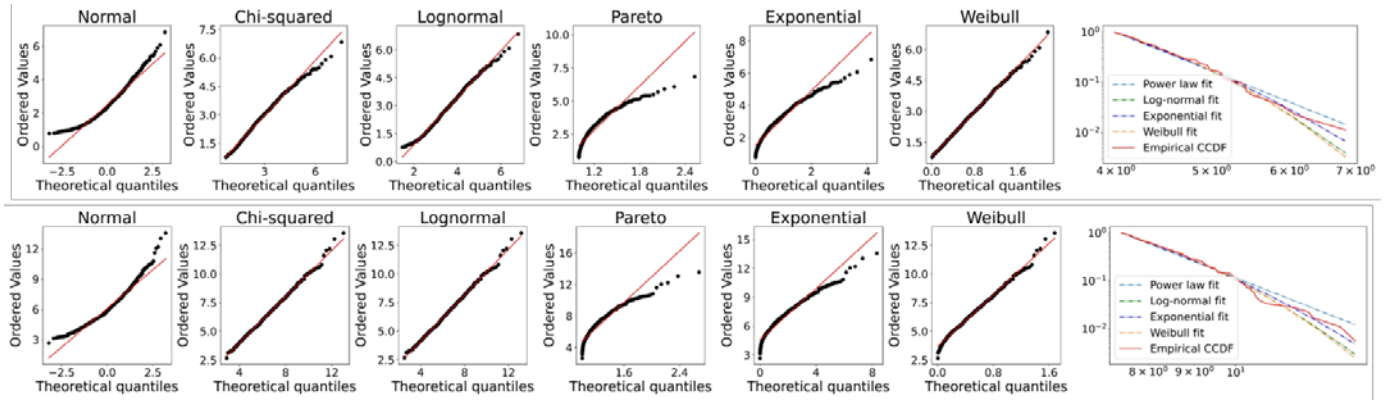

그림 2 QQ-plots of SGN of GIN training. Top row is for the initial phase and bottom row is for the final phase.

for GCN should be modeled as multivariate Gaussian.

5. Conclusion and Future Works

In this paper we provide preliminary statistical analysis of SGN of GNNs. From theoretical perspective, the most interesting question would be to see whether specific graph properties can be analytically incorporated into the dynamics of SGD for simple GNNs with simple losses such as MSE.

6. References

[1] L. Bottou et al., “Optimization methods for large-scale machine learning.” *SIAM Review* 60(2): 223-311, 2018.

[2] U. Şimşekli et al., “A Tail-Index Analysis of Stochastic Gradient Noise in Deep Neural networks.” In ICML 2019.

[3] P. Imkeller et al., “First Exit Times of Non-linear Dynamical Systems in R^d Perturbed by Multifractal Lévy Noise.” *Journal of Statistical Physics* 141: 94-119, 2010.

[4] Z. Xie et al., “A Diffusion Theory For Deep Learning Dynamics: Stochastic Gradient Descent Exponentially Favors Flat Minima.” In ICLR 2021.

[5] A. Panigrahi et al., “Non-Gaussianity of Stochastic Gradient Noise.” In NeurIPS 2019 Science meets Engineering of Deep Learning (SEDL) Workshop

[6] X. Wang et al., “Eliminating Sharp Minima from SGD with Truncated Heavy-tailed Noise.” In ICLR 2022.

[7] P. Awasthi et al., “A Convergence Analysis of Gradient Descent on Graph Neural Networks.” In NeurIPS 2021.

[8] K. Xu et al., “Optimization of Graph Neural Networks: Implicit Acceleration by Skip Connections and More Depth.” In ICML 2021.

[9] T. Kipf & M. Welling, “Semi-Supervised Classification with Graph Convolutional Networks.” In ICLR 2017.

[10] K. Xu et al., “How Powerful are Graph Neural Networks?” In ICLR 2019.

[11] W. Hamilton et al., “Inductive Representation Learning on Large Graphs.” In NeurIPS 2017.

[12] J. Chen et al., “FastGCN: Fast Learning with Graph Convolutional Networks via Importance Sampling.” In ICLR 2018.

[13] J. Alstott et al., “powerlaw: A Python Package for Analysis of Heavy-Tailed Distributions.” *PLOS ONE* 9(1): e85777, 2014.