Understanding dynamic sparse training capabilities in accommodating sparse data

Işıl Baysal Erez[1][0000−0002−0636−5372], Elena Mocanu[1][0000−0002−0856−579X], and Maurice van Keulen[1][0000−0003−2436−1372]

EEMCS, University of Twente, 7500 AE Enschede, The Netherlands
{i.baysalerez, e.mocanu, m.vankeulen}@utwente.nl

Abstract. Deep learning algorithms have became the state-of-the-art models for various tasks in a large area of applications. The most advanced deep learning models have many parameters, increasing costs, computational requirements, and memory footprints. Recently, the dynamic sparse training methods showed that it is possible to outperform the dense neural networks with sparse neural networks, while reducing the number of parameters (connections) quadratically. So far, all the proposed sparse training methods are tested on well-known benchmark datasets without data quality problems. However, in real-world data science applications, a lot of data quality challenges may appear, (e.g. missing data). Missing data can cause daunting challenges in determining the accuracy of models. Within this research, we intend to understand the interplay between dynamic sparse training methods and data sparsity for their mutual benefits.

Keywords: Deep learning · Dynamic Sparse Training · Sparse Data

1 Introduction

Enabled by the large amount of data that are continuously recorded, deep learning has been proven successful for various tasks (e.g., classification, regression, clustering, feature selection, and dimensionality reduction) in a large area of applications. However, the most advanced deep learning models have many parameters, increasing costs, computational requirements, and memory footprints. For example, Inception-V3 [14] a highly accurate object recognition network, requires 5.7 billion arithmetic operations and 27 million parameters to be evaluated, while GPT-3 [2] an experimental natural language processing network, requires 175 billion parameters (350 GiB assuming 16 bits per parameter). The latest language model, GPT-4, is claimed to have 100 trillion parameters — 500x the size of GPT-3. Training these huge models require extensive computing facilities leading to an undesirable impact on our otherwise scarce resources (i.e. energy consumption and CO2 footprint [12]). Thus, deep learning models, in general, must be trained in the cloud and then moved on to the hardware for exploitation. This limits their flexibility drastically.

2 Related work

A plethora of methods have been proposed, ranging from more traditional compression and pruning methods [8] [11] to the most recently introduced sparse training methods [9] [6], aiming to obtain faster and cheaper training and inference for neural networks. Model pruning assumes the training of large neural networks in the cloud. Then, it identifies the least meaningful connections based on various criteria, e.g., magnitude or information metrics. After that, those unimportant connections are removed in order to obtain models with sparse connectivity, which yield smaller memory footprints and faster computational times during inference. The sparse models obtained following the above procedures, also called dense-to-sparse models, can be as effective and even be exploited for inference purposes. Perhaps one of the most recent and popular dense-to-sparse method is The Lottery Ticket Hypothesis [7]. The solutions discussed above are limited by the initial need to train a very large neural network in the cloud. Recently, it has been shown that sparse neural networks trained from scratch (i.e., sparse training) can reach or even can outperform dense neural networks while using much fewer computational resources, and consequently, small memory footprint, while having high representational power. The basic idea of sparse-to-sparse models is to use sparse connectivity before training in neural networks [1, 9, 10, 4, 6, 15]. This starts to appear as the "de facto" solution to obtain sparse neural networks with a small number of parameters, which many times outperform even their very large dense counterparts. All in all, this makes the adaptive sparse connectivity concept very suitable for the training and exploitation of neural networks.

In this research, sparse data will be described as a subset of the data. It can be distinguished between two cases. First one includes some missingness mechanisms which have been traditionally divided into three main categories, such as Missing Completely at Random (MCAR) — in which the probability of missing a data depends neither of observed or unobserved data, Missing at Random (MAR) — in which the probability of missing a data depends just on observed data, and Missing not at Random (MNAR) — in which the probability of missing a data depends on observed data conditioned by observed measurements [13]. Second case includes, when all data are present, but the artificial neural network can not model them correctly. For example, data which are recognized to be easy-to-forget or hard-to-be-memorized.

3 Sparse data and sparse training models

We envision a general framework that accounts for mutual benefits between *sparse data* and *sparse models*. The main PhD research is split into two main parts. In the first part, we have started investigating the learning capabilities of dynamic sparse training methods to account for sparse data (e.g. missing data) in a given incomplete observability environment. In the second part, we aim to explore and extend the dynamic sparse training methods to accommodate special types of data recognized as being hard to perceive by the neural network model (e.g. easy-to-forget or hard-to-memorize) and the main well-known challenges related to it. The general flow of information, including the connection between both parts, can be seen in Figure 1.

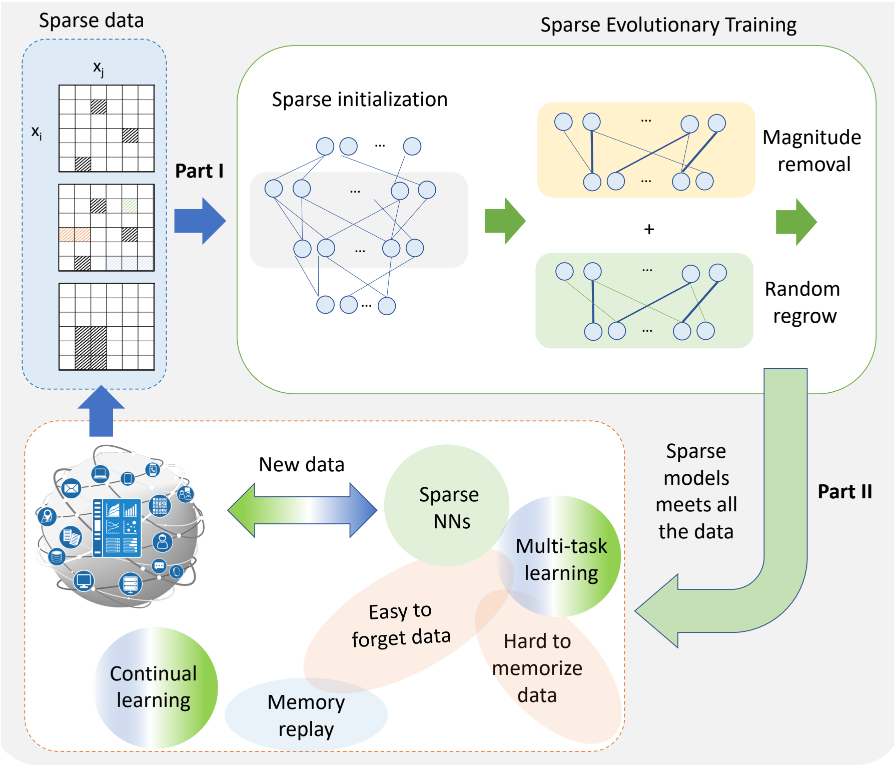

Fig. 1. Schematic representation of main research components and their interrelations. Sparse data includes all three missing data mechanisms: MCAR, MAR and MNAR. Black hash represents a missing data point, and orange and green shadows represents the conditional dependence to that specific missing point in case of missing at random. **Part I** consider the sparse data as input into a dynamic sparse model (i.e. Sparse Evolutionary Training [9]). **Part II** focus on understanding the possible difficulties to perceive the given data and their reflection on online learning tasks.

Part I: Incomplete data observability In real-world data science applications, quite often, the neural networks models should be able to accommodate various data quality aspects (e.g., missing data) and take advantage in a data-efficient way of all of them. In this first part, we want to go one step beyond the state-of-the-art and understand the interplay between "sparse data" and "sparse training models". This lead us to the following research questions:

Can sparse training benefit from a data-efficient solution? How should the "sparse training models" and "sparse data" interact? Can they possibly be simultaneously used to drive training efficiency to the next level?

Part II: Incomplete modeling In this second part, we aim to investigate the learning capabilities of sparse neural networks in a fully-observable environment, where all the data are present, but the neural network may not perceive them. Recently, it has been observed that deep neural networks can have samples that are either hard-to-memorize during training or easy-to-forget during pruning. Prior work has shown that using a specifically selected subset of data rather than the all training set, a dense-to-sparse model may overcome these issues and identify lottery tickets more effectively – leading to the general idea for this particular pruning models [16]. Based on our knowledge, the hard-to-memorize and easy-to-forget sample problems have not been analyzed in the context of any sparse-to-sparse models. Consequently, in the second part of this research, the following research questions will be investigated:

To what extent can we find a subset of data that allows us to overcome the catastrophic forgetting problem in sparse neural networks? Are the dynamic sparse training algorithms able to accommodate the hard-to-memorize and easy-to-forget samples?

4 Preliminary Results

We are considering the Wisconsin Breast Cancer [5] dataset, a very simple multivariate benchmark dataset with two classes, benign and malign, and 32 variables. We simulate the MCAR data using a set-aside masked values from the full data matrix. The obtained sparse data are then fed into a multi-layer perceptron (MLP) and trained using a sparse evolutionary training (SET) procedure. We compare the learning capabilities and computational performance of SET-MLP with a dense MLP and a static sparse MLP (SS-MLP).

For all three models, we use similar hyperparameters and a $30 \times 100 \times 100 \times 100 \times 2$ architecture. For the three hidden layers, we use a ReLU activation function, while for the output layer, we use Softmax. During training, we use a stochastic gradient descend optimizer and a cross-entropy loss function. Furthermore, we use a learning rate of 0.001, a momentum of 0.9, a batch size of 10, and a dropout rate of 0.1. The initial sparsity level for SET-MLP and Static Sparse MLP varies, while for SET-MLP the evolution of the weights is considered with a $\zeta = 0.1$ rate. All models have been run three times using a truly sparse implementation, building on top of the Curci et al. [3] code.

In Fig. 2 (left), we plot the accuracy of SET-MLP (solid lines) versus SS-MLP (dashed lines) and their dense MLP counterpart (black line) in the case of low (i.e. 25%) and moderate (i.e. 50% and 75%) model sparsities. We can observe that both resource-aware models, SET-MLP and SS-MLP, are often more accurate than the dense network while having a quadratically reduced number of parameters. Furthermore, as we increase the level of sparsity toward the high sparsity regime, as can be seen in Fig.2 (right), all three methods show

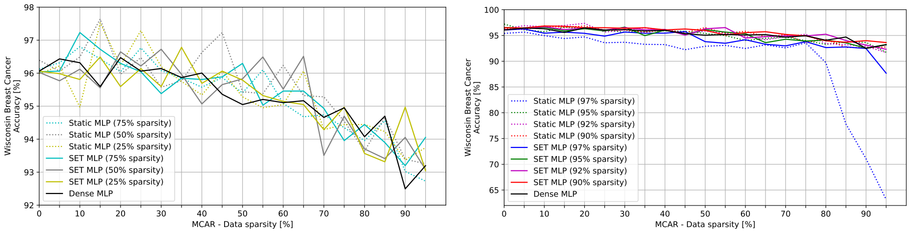

Fig. 2. Comparison over Wisconsin Breast Cancer dataset between a dense MLP, a static sparse MLP, and SET-MLP in (left) a low to moderate sparse regime and (right) an extreme sparse training regime.

good generalization capabilities and similar accuracy. Beyond a 90% reduction in both the data input used and model parameters, the huge gain in computational performance is reflected in a slight decrease in accuracy.

Our initial results using data in an incomplete observability environment show a great potential towards an impressive computational reduction and an increase in accuracy. These results are yet limited to one dynamic sparse training algorithm (i.e., Sparse Evolutionary Training [9] and one dataset where the missing completely at random features have been simulated (Breast Cancer Wisconsin) [5]. Further theoretical considerations and empirical results are currently under development to generalize these results and support these impactful claims.

5 Acknowledgments

This research has been partly funded by the Turkey's Ministry of National Education study-abroad program.

References

1. Guillaume Bellec, David Kappel, Wolfgang Maass, and Robert Legenstein. Deep rewiring: Training very sparse deep networks. *arXiv preprint arXiv:1711.05136*, 2017.
2. Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.
3. Selima Curci, Decebal Constantin Mocanu, and Mykola Pechenizkiyi. Truly sparse neural networks at scale, 2021.
4. Tim Dettmers and Luke Zettlemoyer. Sparse networks from scratch: Faster training without losing performance. *arXiv preprint arXiv:1907.04840*, 2019.
5. Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.
6. Utku Evci, Trevor Gale, Jacob Menick, Pablo Samuel Castro, and Erich Elsen. Rigging the lottery: Making all tickets winners. In *International Conference on Machine Learning*, pages 2943–2952. PMLR, 2020.

7. Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018.
8. Yann LeCun, John Denker, and Sara Solla. Optimal brain damage. *Advances in neural information processing systems*, 2, 1989.
9. Decebal Constantin Mocanu, Elena Mocanu, Peter Stone, Phuong H Nguyen, Madeleine Gibescu, and Antonio Liotta. Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science. *Nature communications*, 9(1):1–12, 2018.
10. Hesham Mostafa and Xin Wang. Parameter efficient training of deep convolutional neural networks by dynamic sparse reparameterization. In *International Conference on Machine Learning*, pages 4646–4655. PMLR, 2019.
11. Michael C Mozer and Paul Smolensky. Using relevance to reduce network size automatically. *Connection Science*, 1(1):3–16, 1989.
12. David Patterson, Joseph Gonzalez, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, and Jeff Dean. Carbon emissions and large neural network training. *arXiv preprint arXiv:2104.10350*, 2021.
13. Donald B Rubin. Inference and missing data. *Biometrika*, 63(3):581–592, 1976.
14. Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.
15. Geng Yuan, Xiaolong Ma, Wei Niu, Zhengang Li, Zhenglun Kong, Ning Liu, Yifan Gong, Zheng Zhan, Chaoyang He, Qing Jin, et al. Mest: Accurate and fast memory-economic sparse training framework on the edge. *Advances in Neural Information Processing Systems*, 34, 2021.
16. Zhenyu Zhang, Xuxi Chen, Tianlong Chen, and Zhangyang Wang. Efficient lottery ticket finding: Less data is more. In *International Conference on Machine Learning*, pages 12380–12390. PMLR, 2021.