Model Compression Beyond Size Reduction

Anonymous authors
Paper under double-blind review

Abstract

With the current set-up, the success of Deep Neural Network models is highly tied to their size. Although this property might help them improve their performance, it makes them difficult to train, deploy them on resource-constrained machines, and iterate on experiments. There is also a growing concern about their environmental and economic impacts. Model Compression is a set of techniques that are applied to reduce the size of models without a significant loss in performance. Their use is increasing as models grow with time. However, these techniques alter the behavior of the network beyond reducing its size. This paper aims to draw attention to the matter by highlighting present works with regard to Explaniability, Neural Architecture Search, and Fairness before finalizing with a suggestion for future research directions.

1 Introduction

Breakthrough advances in Artificial Intelligence algorithms are consequences of Neural Networks which are the foundations for Deep Learning which is a family of Machine Learning algorithms behind successful Artificial Intelligence innovation in tasks like Voice Recognition, Image ClassificationKrizhevsky et al. (2012), Human Language understanding Devlin et al. (2019), etc. Deep Learning algorithms are based on the universal approximation theoremHornik et al. (1989), which guarantees that Neural Networks, grouped in some way, can compute any continuous function, and the hierarchical representation of information inspired by the human brainBengio et al. (2009). These ideas incentivized over-parameterization of networks as the exact number of parameters a specific Deep Learning model needs for a certain problem remained a hyper-parameter, a choice of the designer. Therefore, in order to increase a model's representation capacity and thus performance, the design of models is growing larger and larger as the tasks they are being applied to grow in complexity Bianco et al. (2018).

Model Compression is a set of techniques for reducing the size of large models without a significant loss in performance. Its necessity is increasing in parallel with the advancement and complexity of large models. Even though their recent attempts to train smaller networks from the beginning Frankle & Carbin (2019), mostly, Model Compression is applied after training a bigger model as the model needs much fewer parameters for inference than for learning.

There are numerous Model Compression methods in the literature that can be classified into four major parts: Pruning, Knowledge Distillation (KD), Quantization, and other methods. Pruning is removing an unwanted structure from a trained network. The way to determine what part of the network is unwanted, how to, and when to remove creates different variants of Pruning. Knowledge Distillation (KD) is a mechanism for transferring the knowledge of a bigger network onto a smaller network. Quantization is reducing the number of bits of model parts require to be represented. It is similar to approximation. The rest of the Model Compression methods other than Pruning, Knowledge Distillation (KD), and Quantization are can be organized in one section and are referred to as Other methods. Weight decomposition methods deserve to be to have their own section, but this arrangement makes the paper simpler for the reader.

2 Model Compression Beyond Size Reduction

The purpose of Model Compression is to reduce the size of networks, but the size reduction can alter the behavior of the network and this change can benefit or harm performance. Where it benefits, it will be an extra advantage. In fact, in some cases, it can be done for the purpose of the extra advantage LeCun et al. (1989).

Basically, all decisions that were made about the network before compressing it can be questioned after the compression, but mentioned here are four of them for there is a lack of enough work in the area.

2.1 Reducing Overfitting

Ideally, a model is expected to generalize and not overfit. Extreme Pruning does damage generalizationHan et al. (2015) but a certain level of pruning also helps parameters learn representations independently and serve as a means to introduce noise in the network which serves as a regularizer which in turn reduces overfitting. Overfitting is when a model learns the detail and noise in the training data to the extent that it negatively impacts the performance. The most famous application of compression, especially pruning, is generalization. In fact, in the early days of neural networks, the purpose of pruning was to reduce overfitting LeCun et al. (1989). Recently, dropouts, randomly dropping out weights from the network with a certain probability, Srivastava et al. (2014) enabled models to go deeper than usual and reignited the consideration of compression as a cure for overfitting. Dramatic size reduction with KD Romero et al. (2015) impacts generalization negatively.

2.2 Explainability

Explainability or interpretability is an effort to try to understand the decisions of neural networks. This issue is getting more and more attention due to high stake applications of neural networks. In Frosst & Hinton (2017), the researchers distilled the knowledge of a high-performing neural network into a decision tree model which is inherently explainable. A later work Liu et al. (2018) generalized the application of Knowledge Distillation for the purpose of interpretability by formulating the problem as a multi-output regression problem. The result is a Decision tree that performs better than one trained on the data directly but not better than the original neural network. Thus, trading a little bit of accuracy for interpretability. But this is a double-edged phenomenon because compressing a trained model impacts explanations as it can be seen Joseph et al. (2020) and Park et al. (2020) where the authors suggested explanation preserving methods. There is a lack of detailed work between compression and Explainability. For example, what model compression does to Mechanstic Interpretability, an effort trying to understand what happens inside Neural Networks, remained a mysteryOlah (2022).

2.3 Neural Architecture Search

Neural Architecture Search is a relatively recent approach in the AI community. It is an attempt to find an optimal architecture in an educated way. This is because existing novel architectures are almost human choices and could have been different. The relationship of Neural Architecture Search with model compression is also intuitive and well recognized in literature Cheng et al. (2017). The task of optimal compression can be seen as a search in the space of sub-architectures. For example, the task of Pruning can be taken as a search in the space of architectures that are subnetworks of the original network. The works in Yang et al. (2020) and Ashok et al. (2018) directly demonstrate this concept.

2.4 Algorithmic Fairness

Algorithmic fairness has become increasingly important due to the increasing impact of AI on our society. Particularly, as they are being adopted in various fields of high social importance or automated decision-making, the question of how fair the algorithm is is critical now more than ever. Since Model Compression is becoming a default component of Machine Learning deployment, its impact on fairness has to be examined. A recent work aimed at this problem reports that model compression, Pruning, and weight decomposition, seem to exacerbate the bias in a networkStoychev & Gunes (2022).

Table 1: Summary of Model Compression and its consequences

NETWORK BEHAVIOUR	DESCRIPTION
Overfitting	All except for slight Pruning LeCun et al. (1989)damage it
Explainability(attribution)	Pruning and KD damage attribution Joseph et al. (2020)
Explainability	KD improves it Frosst & Hinton (2017), Liu et al. (2018)
Neural Architecture Search	Helps Pruning and KDCheng et al. (2017),Ashok et al. (2018).
Bias	All exacerbate existing biasStoychev & Gunes (2022)
Security(Adversarial attacks)	KD helps, Pruning and Quantization don't Zhao et al. (2018)
Security(Membership Inference Attack)	Pruning can help Wang et al. (2020).

2.5 Security

There are many security issues Neural Networks models focus on two areas: the privacy of the training data and the robustness of the model. These include Gradient Leakage Attacks, where an attacker can gain access to private training data from a model's gradient, Adversarial Samples, where an attacker misleads a trained classifier with carefully designed inputsGoodfellow et al. (2014), and Membership Inference Attacks, where an attacker learns about the training data by making repeated inferencesWang et al. (2020).

In general, both Quantization nor Pruning do not help mitigate Adversarial attacks but both of them can be made helpful at the cost of accuracyZhao et al. (2018). KD is extensively used as a defense mechanism for Adversarial attacks Papernot et al. (2015). Pruning can also be used to prevent Membership Inference Attacks Wang et al. (2020).

3 Open research questions

Despite efforts to train smaller networks from scratch Frankle & Carbin (2019), most Model Compression methods are applied after a big model is trained. Thus, the compression ought to have an effect on other aspects of the model. For example, as pointed out in Cheng et al. (2017), there are still remaining works that can bridge the concept of a model's size (compression) and its Explainability. Some applications of Explainability are mentioned in the pruning section of this writing. But, there are only a few works, save for Calvi et al. (2019), that even acknowledge the existence of a bridge between them. Thus, formalized future work in this direction can be fruitful.

Model Compression is going to become even more important with the advent of Large Language Models and to make them useful practically, a study of their consequences will be an important research direction.

Basically, one can raise different behaviors of a network and ask how compression impacts it. Presented here are only a limited number of them because there is still much work to be done in the area. A comprehensive survey on the effect of Model Compression beyond size reduction can be extremely helpful to build Model Compression without ramifications and beyond.

References

Anubhav Ashok, Nicholas Rhinehart, Fares N. Beainy, and Kris M. Kitani. N2n learning: Network to network compression via policy gradient reinforcement learning. *ArXiv*, abs/1709.06030, 2018.

Yoshua Bengio et al. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009.

Simone Bianco, Remi Cadene, Luigi Celona, and Paolo Napoletano. Benchmark analysis of representative deep neural network architectures. *IEEE Access*, 6:64270–64277, 2018. ISSN 2169-3536. doi: 10.1109/ACCESS.2018.2877890.

Giuseppe Giovanni Calvi, Ahmad Moniri, Mahmoud Mahfouz, Qibin Zhao, and Danilo P. Mandic. Compression and interpretability of deep neural networks via tucker tensor layer: From first principles to tensor valued back-propagation. *arXiv: Learning*, 2019.

Yu Cheng, Duo Wang, Pan Zhou, and Tao Zhang. A survey of model compression and acceleration for deep neural networks. *ArXiv*, abs/1710.09282, 2017.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv*, abs/1810.04805, 2019.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv: Learning*, 2019.

Nicholas Frosst and Geoffrey E. Hinton. Distilling a neural network into a soft decision tree. *ArXiv*, abs/1711.09784, 2017.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572, 2014.

Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28, 2015.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

Vinu Joseph, Shoaib Ahmed Siddiqui, Aditya Bhaskara, Ganesh Gopalakrishnan, Saurav Muralidharan, Michael Garland, Sheraz Ahmed, and Andreas R. Dengel. Going beyond classification accuracy metrics in model compression. 2020.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60:84 – 90, 2012.

Yann LeCun, John S. Denker, and Sara A. Solla. Optimal brain damage. In *NIPS*, 1989.

Xuan Liu, Xiaoguang Wang, and Stan Matwin. Improving the interpretability of deep neural networks with knowledge distillation. *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, pp. 905–912, 2018.

Chris Olah. Mechanistic Interpretability, Variables, and the Importance of Interpretable Bases. https://transformer-circuits.pub/2022/mech-interp-essay/index.html, jun 22 2022. [Online; accessed 2022-12-14].

Nicolas Papernot, Patrick Mcdaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. *2016 IEEE Symposium on Security and Privacy (SP)*, pp. 582–597, 2015.

Geondo Park, June Yong Yang, Sung Ju Hwang, and Eunho Yang. Attribution preservation in network compression for reliable network interpretation. *Advances in Neural Information Processing Systems*, 33:5093–5104, 2020.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *CoRR*, abs/1412.6550, 2015.

Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15: 1929–1958, 2014.

Samuil Stoychev and Hatice Gunes. The effect of model compression on fairness in facial expression recognition. *ArXiv*, abs/2201.01709, 2022.

Yijue Wang, Chenghong Wang, Zigeng Wang, Shangli Zhou, Hang Liu, Jinbo Bi, Caiwen Ding, and Sanguthevar Rajasekaran. Against membership inference attack: Pruning is all you need. In *International Joint Conference on Artificial Intelligence*, 2020.

Zhaohui Yang, Yunhe Wang, Xinghao Chen, Boxin Shi, Chao Xu, Chunjing Xu, Qi Tian, and Chang Xu. Cars: Continuous evolution for efficient neural architecture search. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1826–1835, 2020.

Yiren Zhao, Ilia Shumailov, Robert D. Mullins, and Ross Anderson. To compress or not to compress: Understanding the interactions between adversarial attacks and neural network compression. *ArXiv*, abs/1810.00208, 2018.