SURROGATE-BASED NEURAL NETWORK STRUCTURAL PRUNING

Maksim Ivanov
Department of Intelligent Systems
MIPT
ivanov.mo@phystech.edu

Oleg Bakhteev
Department of Intelligent Systems
MIPT
bakhteev@phystech.edu

ABSTRACT

The paper investigates the problem of structural pruning of models. Structural pruning is the process of removing groups of unimportant [TODO: irrelevant?] weights from a neural network, for example, filters in CNN or skip-connections. Proper pruning strategy leads to improvement of both generalizing ability and inference performance. Main difficulty of structural pruning is that when one layer of the network is removed, its dependent layers should also be removed. The proposed method is based on the deep learning computation graph analyzing and estimation of information flow transferred through it. The method enables estimation of the importance of operations in a computation graph in a zero-shot mode, i.e., using only a single pass of a subset of data through the analyzed model.p The basic idea [TODO]. To demonstrate the performance of the proposed method we conduct multiple experiments on synthetic data, CIFAR-10 and Wikitext dataset [TODO].

1 Introduction

Training neural networks requires increasingly large amounts of computing power [Sevilla et al., 2022]. Training models with novel architectures can be a challenging task due to constraints on the computational budget [Thompson et al., 2020]. In addition, edge computing applications require neural network inference to be performed on portable devices [Banjanović-Mehmedović and Husaković, 2023]. Consequently, there is a growing need to improve both the generalization ability and the inference performance of neural networks. One of the approaches to address the issues mentioned above is model pruning.

By pruning, we mean the task of reducing the size of a network by removing parameters [Blalock et al., 2020]. Pruning is generally classified into unstructured and structured. In the first case, individual unimportant weights are set to zero, while in the second case, entire groups of unimportant weights are removed from the neural network. Several methods for unstructured pruning have been proposed [Lecun et al., 1989, Hassibi and Stork, 1992, Zeng and Urtasun, 2019, Han et al., 2015]. All of them rely on weight removal according to some importance criterion, for example, weight magnitude [Han et al., 2015, Lubana and Dick, 2020], norm [He et al., 2018], or loss change [Molchanov et al., 2016, Liu et al., 2021].

Although unstructured pruning helps reduce the computational resources required for inference [Laurent et al., 2020], pruning models with novel architectures still remains a challenge, especially when dependent layers must be removed simultaneously. This issue is investigated in Fang et al. [2023], where structural pruning is employed by constructing a Dependency Graph to explicitly model inter-layer dependencies and comprehensively group coupled parameters for pruning. Other works review the structural pruning problem and propose methods that either do not use dataset information [Tanaka et al., 2020] or rely only on a small number of samples [Sun et al., 2023]. The goal of many investigations is to perform pruning in a zero-shot setting, i.e., without fine-tuning [Chen et al., 2021]. However, many methods are designed for specific architectures, which poses a significant obstacle to applying such algorithms across diverse problems [Sun et al., 2023, Wang et al., 2019].

In this paper, we investigate the problem of structural pruning. We consider the computation graph as a directed graph in which the vertices correspond to the layers. The proposed method is based on analyzing the deep learning computation graph and estimating the information flow transferred through it. Our goal is to perform structural pruning on arbitrary neural network architectures, without restricting the method to a specific one. The method enables estimation of the importance of operations in a computation graph in a zero-shot setting, i.e., using only a single forward pass of a subset of data through the analyzed model. [TODO: sota ?]

The computational experiment is performed on synthetic data, CIFAR-10 and Wikitext dataset [TODO:].

2 Problem statement

In this work, we address the problem of structural pruning in deep neural networks, formulated in terms of their computation graphs. We define the **computation graph** of a neural network as a directed graph $G = (V, E)$, where each vertex $v_i \in V$ corresponds to a layer (or a computational module) of the network, and each directed edge $e_{ij} \in E$ represents the data flow from the output of layer v_i to the input of layer v_j. Formally,

$$e_{ij} \in E \quad \Leftrightarrow \quad \mathbf{h}_j = \mathbf{f_j}(\mathbf{h}_i),$$

where $\mathbf{h}_i$ denotes the activation produced by vertex v_i, and $\mathbf{f_j}$ is the transformation implemented by vertex v_j. [TODO] In this formulation, removing a vertex corresponds to eliminating the entire layer from the network, whereas removing an edge corresponds to cutting a data dependency between two layers (e.g., in residual or multi-branch architectures).

[TODO] We consider three possible structural pruning scenarios:

1. Edge removal — deleting certain connections (e_{ij}) in E to reduce memory consumption and computational cost, at the risk of accuracy degradation.
2. Edge removal with fine-tuning — pruning followed by re-optimization of the remaining parameters to recover lost performance.
3. Transfer learning.

Let the training set be

$$\mathcal{D} = \{(\mathbf{x}_n, y_n)\}_{n=1}^N, \quad \mathbf{x}_n \in \mathbb{R}^d, \quad y_n \in \mathcal{Y},$$

and let $\mathbf{w} \in \mathbb{R}^k$ be the vector of all model parameters. The model is trained by minimizing a loss function $\mathcal{L}(\mathcal{D} \mid \mathbf{w})$.

After pruning, we obtain a sparse parameter vector $\mathbf{w}' \in \mathbb{R}^k$, in which certain groups of parameters are set to zero corresponding to removed edges or vertices. The challenge of structural pruning lies in estimating the importance of parameter groups in terms of their contribution to the network's information flow and final accuracy.

We investigate two general approaches:

1. Loss-based pruning — directly measuring the post-pruning loss difference and selecting $\mathbf{w}'$ to minimize the performance drop:

$$\min_{\mathbf{w}' \in \mathbb{R}^k} \; |\mathcal{L}(\mathcal{D} \mid \mathbf{w}') - \mathcal{L}(\mathcal{D} \mid \mathbf{w})|$$

.

2. Taylor approximation of the loss function — estimating the impact of pruning using a second-order Taylor expansion around $\mathbf{w}$:

$$\mathcal{L}(\mathcal{D} \mid \mathbf{w}') \approx \mathcal{L}(\mathcal{D} \mid \mathbf{w}) + \mathbf{g}^\top \Delta\mathbf{w} + \frac{1}{2}\Delta\mathbf{w}^\top \mathbf{H}\, \Delta\mathbf{w},$$

where $\mathbf{g} = \frac{\partial \mathcal{L}}{\partial \mathbf{w}}$ is the gradient and $\mathbf{H}$ is the Hessian matrix.

The problem now is to choose an estimation strategy for parameter group importance that leads to effective pruning, i.e., maximizes the reduction in model size and computational complexity while keeping the loss increase within acceptable bounds.

References

Jaime Sevilla, Lennart Heim, Anson Ho, Tamay Besiroglu, Marius Hobbhahn, and Pablo Villalobos. Compute trends across three eras of machine learning. *2022 International Joint Conference on Neural Networks (IJCNN), Padua, Italy, 2022, pp. 1-8*, pages 1–8, July 2022. doi:10.1109/ijcnn55064.2022.9891914.

Neil C. Thompson, Kristjan Greenewald, Keeheon Lee, and Gabriel F. Manso. The computational limits of deep learning. July 2020. doi:10.48550/ARXIV.2007.05558.

Lejla Banjanović-Mehmedović and Anel Husaković. Edge ai: Reshaping the future of edge computing with artificial intelligence. In *BASIC TECHNOLOGIES AND MODELS FOR IMPLEMENTATION OF INDUSTRY 4.0*, Basic 4.0, pages 133–160. Academy of Sciences and Arts of Bosnia and Herzegovina, October 2023. doi:10.5644/pi2023.209.07.

Davis Blalock, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John Guttag. What is the state of neural network pruning? March 2020. doi:10.48550/ARXIV.2003.03033.

Yann Lecun, John Denker, and Sara Solla. Optimal brain damage. volume 2, pages 598–605, 01 1989.

Babak Hassibi and David Stork. Second order derivatives for network pruning: Optimal brain surgeon. In S. Hanson, J. Cowan, and C. Giles, editors, *Advances in Neural Information Processing Systems*, volume 5. Morgan-Kaufmann, 1992. URL `https://proceedings.neurips.cc/paper_files/paper/1992/file/303ed4c69846ab36c2904d3ba8573050-Paper.pdf`.

Wenyuan Zeng and Raquel Urtasun. MLPrune: Multi-layer pruning for automated neural network compression, 2019. URL `https://openreview.net/forum?id=r1g5b2RcKm`.

Song Han, Jeff Pool, John Tran, and William J. Dally. Learning both weights and connections for efficient neural networks. June 2015. doi:10.48550/ARXIV.1506.02626.

Ekdeep Singh Lubana and Robert P. Dick. A gradient flow framework for analyzing network pruning. September 2020. doi:10.48550/ARXIV.2009.11839.

Yang He, Guoliang Kang, Xuanyi Dong, Yanwei Fu, and Yi Yang. Soft filter pruning for accelerating deep convolutional neural networks. August 2018. doi:10.48550/ARXIV.1808.06866.

Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. Pruning convolutional neural networks for resource efficient inference. November 2016. doi:10.48550/ARXIV.1611.06440.

Liyang Liu, Shilong Zhang, Zhanghui Kuang, Aojun Zhou, Jing-Hao Xue, Xinjiang Wang, Yimin Chen, Wenming Yang, Qingmin Liao, and Wayne Zhang. Group fisher pruning for practical network compression. August 2021. doi:10.48550/ARXIV.2108.00708.

César Laurent, Camille Ballas, Thomas George, Nicolas Ballas, and Pascal Vincent. Revisiting loss modelling for unstructured pruning. June 2020. doi:10.48550/ARXIV.2006.12279.

Gongfan Fang, Xinyin Ma, Mingli Song, Michael Bi Mi, and Xinchao Wang. Depgraph: Towards any structural pruning. January 2023. doi:10.48550/ARXIV.2301.12900.

Hidenori Tanaka, Daniel Kunin, Daniel L. K. Yamins, and Surya Ganguli. Pruning neural networks without any data by iteratively conserving synaptic flow. *Advances in Neural Information Processing Systems 2020*, June 2020. doi:10.48550/ARXIV.2006.05467.

Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. A simple and effective pruning approach for large language models. June 2023. doi:10.48550/ARXIV.2306.11695.

Tianyi Chen, Bo Ji, Tianyu Ding, Biyi Fang, Guanyi Wang, Zhihui Zhu, Luming Liang, Yixin Shi, Sheng Yi, and Xiao Tu. Only train once: A one-shot neural network training and pruning framework. July 2021. doi:10.48550/ARXIV.2107.07467.

Wei-Ting Wang, Han-Lin Li, Wei-Shiang Lin, Cheng-Ming Chiang, and Yi-Min Tsai. Architecture-aware network pruning for vision quality applications. August 2019. doi:10.48550/ARXIV.1908.02125.