Enhancing Network Pruning Method Evaluation and Integration

Yulai Zhao

Princeton University

Abstract

- This study extends the comparative analysis of various neural network pruning techniques—specifically SNIP [2], GraSP [4], SynFlow [3], random pruning, and magnitude based pruning—by integrating modern evaluation metrics and introducing an iterative pruning method, Iterative Magnitude Pruning (IMP) [1].
- Our objectives are to enhance the understanding and efficiency of these techniques for more effective neural network model development.
- We implemented IMP to evaluate its theoretical effectiveness and incorporated additional metrics such as CPU memory usage, GPU allocation, and cache memory tracking.
- Our comparative analysis across different compression levels reveals that iterative pruning methods like IMP tend to outperform one-shot approaches. Furthermore, initial results suggest that each one-shot pruning method presents distinct advantages and limitations. This comprehensive assessment aids in identifying optimal pruning strategies for various network architectures and applications..

Iterative Magnitude Pruning

- Iterative Magnitude Pruning (IMP) [1] is a neural network pruning technique that employs an iterative process to re move weights based on their magnitudes. It seeks to identify a sparse but capable subnetwork that, when trained from the beginning, could match or surpass the performance of the unpruned network. IMP is inspired by the Lottery Ticket Hypothesis, which suggests that effective subnetworks can exist within randomly initialized networks.
- IMP differs significantly from traditional one-shot pruning methods. While one-shot pruning involves removing a pre determined percentage of weights based solely on a single pass or criterion (such as weight magnitude), IMP applies a more nuanced approach. It uses multiple iterations of pruning followed by training, where each cycle aims to eliminate a fixed percentage of the smallest weights and then retrain the network to regain performance. This methodical reduction

Iterative Magnitude Pruning (Cont'd)

and training process allows IMP to refine the network's structure iteratively, enhancing its ability to maintain or improve performance despite increased sparsity.

- This cyclic nature of IMP is crucial for its success. It allows the network to adapt gradually to the loss of weight, which can prevent the significant performance degradation often ob served with one-shot pruning methods after aggressive weight removal. By continuously adjusting and retraining, IMP can discover more efficient and robust network configurations capable of achieving similar or even superior performance compared to the original unpruned model.

Main Results

Compression	Rand	Mag	SNIP	GraSP	SynFlow
0.05	88.08	88.69	88.20	79.03	88.05
0.1	87.51	89.47	88.17	72.34	88.16
0.2	88.16	89.36	87.87	78.87	88.50
0.5	86.80	89.95	88.55	80.86	87.14
1	10.00	88.88	87.77	81.46	87.83
2	10.00	42.61	81.55	82.72	10.00

Compression	IMP (2 Iters)	IMP (3 Iters)	IMP (4 Iters)
0.05	88.56	88.65	89.81
0.1	89.15	88.65	89.06
0.2	88.84	89.20	89.69
0.5	89.14	89.37	88.94
1	89.52	89.57	89.80
2	81.60	75.37	19.34

Table 1: We present the top-1 testing accuracy of a VGG16 model on CIFAR-10 across various compression ratios, comparing one-shot pruning methods (Rand, Mag, SNIP, GraSP, SynFlow) with Iterative Magnitude Pruning (IMP) over 2, 3, and 4 iterations. IMP consistently outperforms other methods, especially at higher compression levels, demonstrating its superior ability to maintain accuracy while reducing model size.

Compression	Rand	Mag	SNIP	GraSP	SynFlow
0.05	0.653	0.630	0.704	0.598	0.617
0.1	0.705	0.676	0.606	0.621	0.620
0.2	0.631	0.637	0.600	0.618	0.682
0.5	0.693	0.639	0.602	0.618	0.603
1	0.606	0.604	0.606	0.613	0.610
2	0.621	0.602	0.668	0.627	0.618

Compression	IMP (2 Iters)	IMP (3 Iters)	IMP (4 Iters)
0.05	0.743	0.700	0.689
0.1	0.646	0.634	0.653
0.2	0.638	0.655	0.666
0.5	0.740	0.672	0.641
1	0.649	0.698	0.641
2	0.638	0.653	0.667

Table 2: Inference time in seconds for a VGG16 model on CIFAR-10 across varying compression ratios using different pruning methods, including one-shot (Rand, Mag, SNIP, GraSP, SynFlow) and iterative (IMP) approaches. IMP consistently shows optimized inference times, particularly at higher iterations, which highlights its efficiency in streamlining network operations post-pruning.

Compression	Rand	Mag	SNIP	GraSP	SynFlow
0.05	0.8916	0.9477	0.9377	0.8201	0.9488
0.1	0.7945	0.8991	0.9268	0.7295	0.9026
0.2	0.6310	0.8151	0.7812	0.5711	0.8217
0.5	0.3165	0.5634	0.4622	0.3781	0.6423
1	0.1009	0.2283	0.1979	0.1751	0.4571
2	0.0108	0.0322	0.0411	0.0586	0.1845

Compression	IMP (2 Iters)	IMP (3 Iters)	IMP (4 Iters)
0.05	0.9290	0.9256	0.9229
0.1	0.8655	0.8612	0.8560
0.2	0.7564	0.7505	0.7414
0.5	0.5189	0.5125	0.5115
1	0.2134	0.2193	0.2280
2	0.0286	0.0245	0.0239

Table 3: Sparsity ratios of a VGG16 model using various pruning techniques at different compression levels. The ratios, calculated as the fraction of remaining FLOPs relative to the original model's 313,478,154 FLOPs, indicate the efficiency of each method in reducing computational complexity. IMP (Iterative Magnitude Pruning) shows a consistent and significant increase in sparsity with more iterations, highlighting its effectiveness in achieving higher computational savings.

Compression	Rand	Mag	SNIP	GraSP	SynFlow
0.05	10.0	9.8	16.0	16.1	9.5
0.1	10.1	10.0	13.0	13.0	9.5
0.2	10.0	10.0	9.8	13.0	9.5
0.5	10.0	10.1	9.8	9.8	9.5
1	10.0	9.8	9.8	9.5	9.5
2	10.0	9.8	9.8	9.5	9.5

Compression	IMP (2 Iters)	IMP (3 Iters)	IMP (4 Iters)
0.05	9.8	9.8	9.8
0.1	9.8	9.8	9.8
0.2	9.8	9.8	9.4
0.5	9.4	9.4	9.4
1	9.4	9.4	9.4
2	9.4	9.4	9.4

Table 4: Maximum CPU memory consumption (in GBs) required for inferencing on the test set using variously pruned VGG16 models at different compression levels. The data demonstrates how iterative pruning (IMP) maintains consistent and lower memory usage across iterations compared to one-shot methods

Main Results (Cont'd)

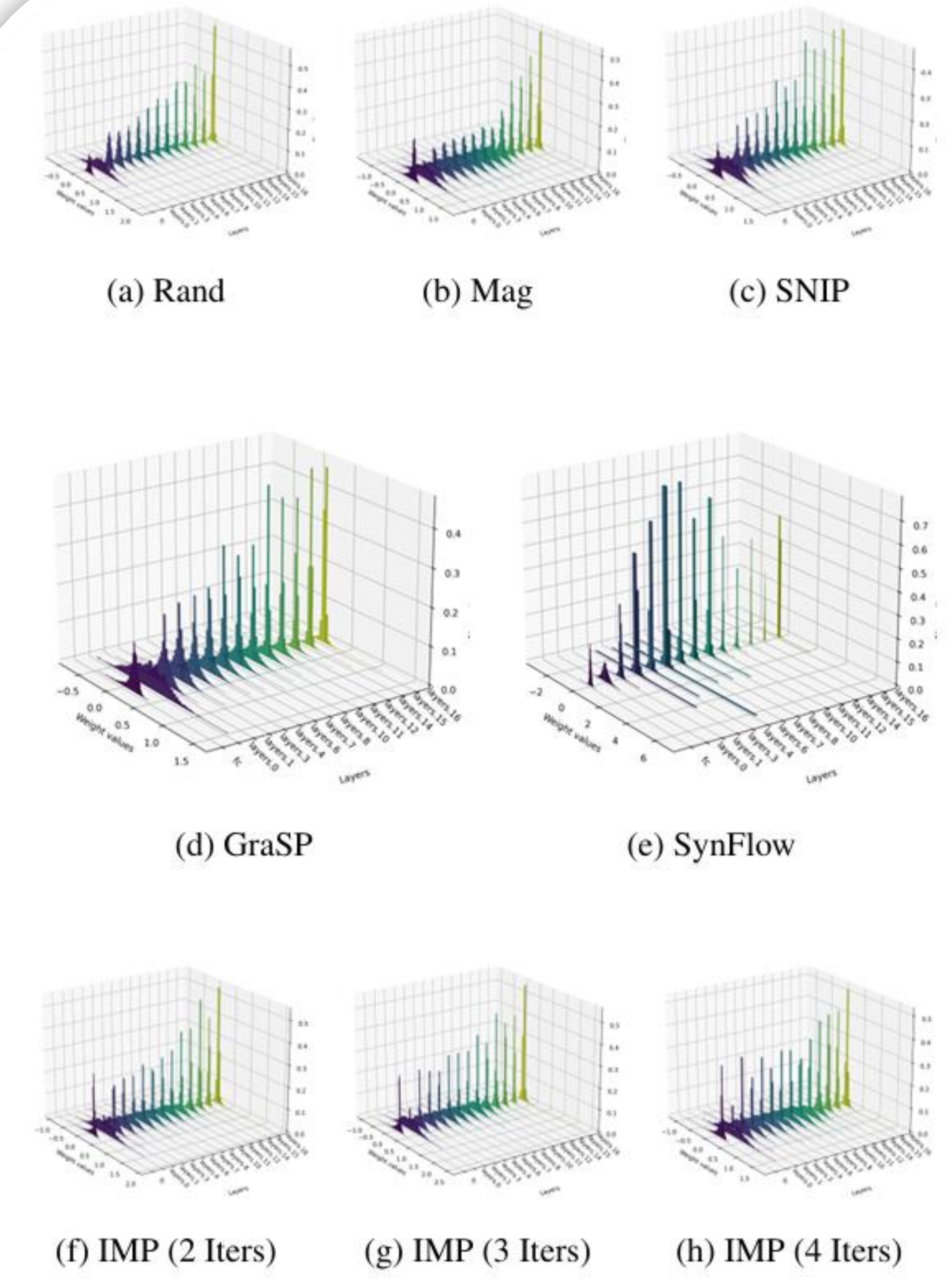

Figure 1: Comparative visualization of weight distributions across various pruning techniques for a VGG16 model trained on CIFAR-10, targeting 50% compression. The plots demonstrate distinct sparsity patterns: Random (Rand) and Magnitude-based (Mag) pruning show less strategic weight removal than structured approaches like SNIP, GraSP, and SynFlow, which more aggressively prune the later layers. Iterative Magnitude Pruning (IMP) over multiple iterations (2 to 4 Iters) refines sparsity, progressively concentrating weight reductions in the final layers. This suggests an adaptive focus on maintaining early layer density for feature extraction while optimizing later layers for decision-making efficiency.

Main Results (Cont'd)

Compression	Rand	Mag	SNIP	GraSP	SynFlow
0.05	0.375	0.376	0.437	0.378	0.374
0.1	0.375	0.376	0.437	0.378	0.374
0.2	0.375	0.376	0.437	0.378	0.374
0.5	0.375	0.376	0.437	0.378	0.374
1	0.375	0.376	0.437	0.378	0.374
2	0.375	0.376	0.437	0.378	0.374

Compression	IMP (2 Iters)	IMP (3 Iters)	IMP (4 Iters)
0.05	0.494	0.495	0.497
0.1	0.499	0.499	0.498
0.2	0.494	0.499	0.494
0.5	0.497	0.499	0.497
1	0.497	0.499	0.497
2	0.494	0.496	0.498

Table 5: Maximum GPU memory allocation in GBs for inferencing on the CIFAR-10 test set using VGG16 models pruned via various techniques at different compression ratios. Compared to one-shot pruning methods, IMP demonstrates higher memory efficiency, particularly at finer iterations.

Compression	Rand	Mag	SNIP	GraSP	SynFlow
0.05	1.124	0.937	1.097	1.267	1.114
0.1	1.124	0.937	1.097	1.267	1.114
0.2	1.124	0.937	1.097	1.267	1.114
0.5	1.124	0.937	1.097	1.267	1.114
1	1.124	0.937	1.097	1.267	1.114
2	1.124	0.937	1.097	1.267	1.114

Compression	IMP (2 Iters)	IMP (3 Iters)	IMP (4 Iters)
0.05	1.277	1.277	1.277
0.1	1.277	1.277	1.277
0.2	1.277	1.140	1.277
0.5	1.140	1.277	1.277
1	1.141	1.141	1.277
2	1.277	1.277	1.277

Table 6: Maximum GPU memory cached for inferencing on the CIFAR-10 test set with VGG16 models pruned at various compression ratios.

Conclusion

- Our comparative study underscores the superiority of iterative pruning over traditional one-shot methods.
- IMP retains model accuracy, reduces inference times, and optimizes memory consumption across both CPU and GPU, making it a robust solution for enhancing the operational efficiency of deep neural networks.
- This study highlights the potential of iterative pruning techniques in advancing the SOTA model compression, offering significant benefits for real-world applications where efficiency and performance are critical.

References

[1] FRANKLE, J., DZIUGAITE, G. K., ROY, D., AND CARBIN, M. Linear mode connectivity and the lottery ticket hypothesis. In International Conference on Machine Learning (2020), PMLR, pp. 3259–3269.

[2] LEE, N., AJANTHAN, T., AND TORR, P. H. Snip: Single-shot network pruning based on connection sensitivity. arXiv preprint arXiv:1810.02340 (2018). [3] TANAKA, H., KUNIN, D., YAMINS, D. L., AND GANGULI, S. Pruning neural networks without any data by iteratively conserving synaptic f low. Advances in neural information processing systems 33 (2020), 6377–6389.

[4] WANG, C., ZHANG, G., AND GROSSE, R. Picking winning tickets before training by preserving gradient flow. In International Conference on Learning Representations (2019).