Statistical Deep Model Pruning

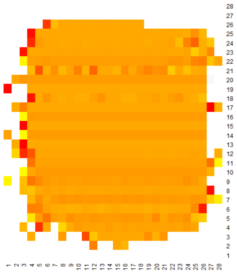
Filip Svoboda

Deep Learning remains severely resource intensive, to the extent that only very simple models can be run on wearable hardware. Consequently, energy efficiency, inference latency, and memory footprint are key in our effort to realize its full potential [5]. One of the most popular approaches to managing these resource constraints is pruning - a method that drops model weights based on an importance criterion specified for each weight or node[1].

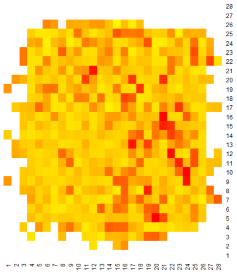
At present there are three main pruning importance criteria. First, Cun *et al.* [3] argued that the second derivative of the loss function with respect to the weights is a measure of their importance and thus should be used to set the pruning order. Second, Augasta & Kathirvalavakumar [1] observed that the frequency and scale of updates is related to the potential of a given weight to improve the loss, and thus the sum of the weight updates is a measure of its importance. Third, Han *et al.* [4] argued that the magnitude of a given weight carries all information necessary to judge its importance[2]. Neither of these methods, however, is supported by a theory, instead, they are based on partial sensitivity observations of the deep system. Therefore, the arguments that support them, could equally well support any function of any of their possible combinations. Moreover, there is no consensus on which criterion performs best, as their relative performance varies significantly between domains [2]. Consequently, we need a stronger theory to guide us.

In this paper, we develop the Local Asymptotic Theory for Deep Model Parameters under fairly weak assumptions, most of which are met trivially. We prove that the local estimate weights follow an asymptotically normal distribution with estimable variance. We define the importance criterion as an asymptotic-distribution-derived z-statistic. We then show that this statistic is closely related to the three current approaches. Specifically, the estimate of the weight distribution's variance is scaled inversely by the loss function derivative and thus enters in the z-statistic directly as in Cun *et al.* [3]. Second, the z-statistic is inversely proportional to the variance of the weight updates - whose sum was used in Augasta & Kathirvalavakumar [1]. Finally, the z-statistic is directly proportional to the weight magnitude as in Han *et al.* [4]. We then show that the three methods and our proposal follow the same dynamics. Therefore, the here-proposed method can be justified both on theoretical grounds as well as on the observational arguments of Cun *et al.* [3], Augasta & Kathirvalavakumar [1], and Han *et al.* [4].

As a short demonstration take a shallow model with the Sigmoid activation function. The data are MNIST non-binarized zero and one labels. What follows are the heath maps of the weight importance as judged by the proposed method and by the most widely used of the three alternatives - the Han *et al.* [4]. The maps are aligned with the underlying input pictures for easy interpretation. White space indicates weights that retained their original, initialized, state. Han *et al.* [4] would retain those that were randomly initialized large, but the specific choice of which would be retained is random and initialization depended. We therefore ignore them.



Han et al. (2015)



Statistical Pruning

The main distinguishing feature of our Statistical Pruning is its focus on retaining the central part of the picture. This makes sense and is desirable since in the domain (zeroes vs. ones) this is where most of the information is – zeroes have white space, while ones have black space there. The proposed method is therefore able to discover the inherent value in these weights. The Han *et al.* [4] method randomly prioritizes weights on the extreme edges and maintains a much flatter prioritization over the rest of the picture. Therefore, it is failing to capture the importance of the key weights in this domain. Similar lessons apply to the other two alternatives.

[1] Node pruning is essentially testing restrictions on all weights on a given node - that is the per-row restrictions.

[2] They scale all weights by the standard deviation of the weights - but since this factor is common, it is inconsequential. As a result the proposed method is equivalent to pruning by weights' absolute value.

References

[1] Augasta, M. G., & Kathirvalavakumar, T.: A novel pruning algorithm for optimizing feedforward neural network of classification problems, *Neural Processing Letters*, 34(3), (2011), 241.

[2] Augasta, M. G., & Kathirvalavakumar, T.: Pruning algorithms of neural networks — a comparative study, *Central European Journal of Computer Science*, 3(3), (2013), 105–115.

[3] Cun, Y. L., Denker, J. S., & Solla, S. A.: *Advances in Neural Information Processing Systems 2*, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1990.

[4] Han, S., Pool, J., Tran, J., & Dally, W.: Learning both weights and connections for efficient neural networks, 2015.

[5] Lane, N., Bhattacharya, S., Georgiev, P., Forlivesi, C., & Kawsar, F.: An early resource characterization of deep learning on wearables, smartphones and internet-of-things devices, 2015.