Winning Weights: A Review of The Lottery Ticket Hypothesis

Søren Winkel Holm

June 17, 2022

Introduction

While Deep Learning (DL) is celebrated as a universal technological step forward, making complex modelling available to many industries requiring less use of domain-specialised engineers, the computational cost of the training procedure of Deep Artificial Neural Network (DNN)'s makes the technology unavailable for low-resource actors. The price of large language modelling projects such as Google's T5 is estimated in the region of $ 10 M and such tech companies have a leading role in research. While the processing of big data is naturally computationally intensive, much of training cost is caused by a large number of epochs being required before convergence. The Lottery Ticket Hypothesis (LTH) points to DNN learning dynamics that are keeping this number high.

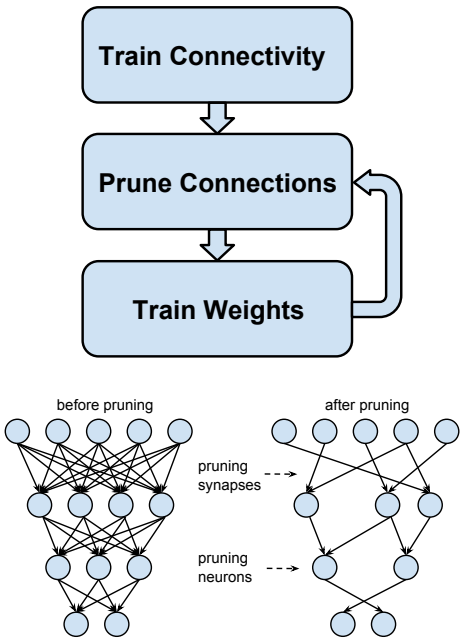

Figure 1: The original iterative pruning method where connectivity training corresponds to standard full training of a dense DNN [Han+15, Fig. 2 and 3].

To help availability, rich actors often share the trained model parametrizations but even in this case, application might be widely inaccessible because of com-

putational costs of inference using these large parametrizations. This problem has been attacked using model pruning, reducing trained model size while retaining performance, but the expensive training of the full DNN has generally been required. LTH explains why an initial full training is generally required and opens up for researching how to train efficient parametrizations from scratch.

Fundamental Concepts

DNN pruning refers to disabling particular model connections $w_i \leftarrow 0$ possibly to improve generalization, reducing memory constraints in inference and lowering inference computation [Han+15]. Pruning during training is related to regularization e.g. using dropout, while pruning after fully training a dense network parametrization $q(w) = \tilde{w}$ is often motivated by computational cost, and might require some retraining to limit the decrease in accuracy [RE20]. Disabling unnecessary weights is a way to learn the connectivity of a DNN and can be performed iteratively based on magnitude such as

$$\forall i \text{ s. t. } |w_i^{(t)}| < k \text{ let } w_i^{(t+1)} \leftarrow 0, \quad (1)$$

where each iteration is followed by retraining, k is a threshold set to e.g. $k = s\sqrt{\text{Var}[w]}, s = \frac{1}{2}$ [Han+15; Zmo+19] and the procedure is stopped when a specified compression level of performance drop is reached [Han+15] as shown on Figure 1. Pruning might be performed locally by assigning layer separate thresholds, having layer-specific target compression levels or fixing part of the DNN [Han+15].

State of the Art

Open Problems

References

[Han+15] Song Han, Jeff Pool, John Tran, and William J. Dally. "Learning Both Weights and Connections for Efficient Neural Networks". In: *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*. NIPS'15. Montreal, Canada: MIT Press, 2015, pp. 1135–1143.

[RE20] Carey Radebaugh and Ulfar Erlingsson. "The Lottery Ticket Hypothesis: A Survey". In: *Rob's Homepage* (June 27, 2020). URL: `https://roberttlange.github.io/posts/2020/06/lottery-ticket-hypothesis` (visited on 06/10/2022).

[Zmo+19] Neta Zmora et al. "Neural Network Distiller: A Python Package For DNN Compression Research". In: (Oct. 2019). URL: `https://arxiv.org/abs/1910.12232`.