Winning Weights: A Review of The Lottery Ticket Hypothesis

Søren Winkel Holm

June 20, 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.

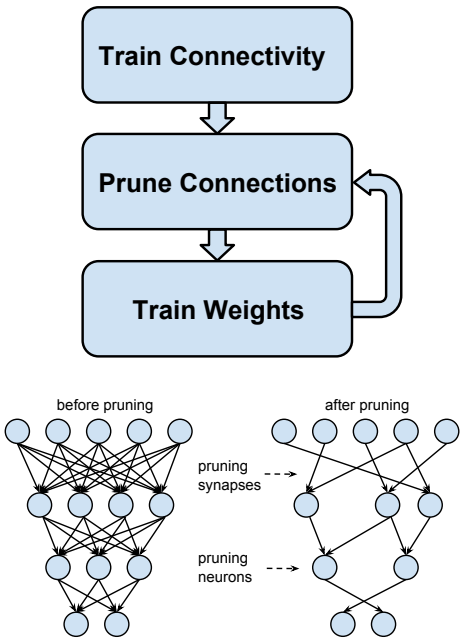

Figure 1: The original iterative pruning method where connectivity training corresponds to standard full training of a dense DNN [Han+15, Fig. 2 and 3].

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. These ideas and methods will here be reviewed.

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 [LDS89]. 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 [Lan20]. 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]. The procedure is stopped when a specified compression level of performance drop is reached [Han+15] as shown on Figure 1. The pruned parametrizations $w^{(p)}$ can be represented using a mask m, $w^{(p)} = m \odot w$ and (1) is thus dubbed the masking criterion. Pruning might be structured locally by assigning layer-specific thresholds and target compression levels or by fixing parts of the DNN [Han+15]. The resulting sparse DNN $f(x; m \odot w)$ is called a subnetwork of the full, trained $f(x; w)$

Simple pruning approaches have empirically been shown to work well across network types and learning tasks with compression rates of $\sim \times 10$ resulting in accuracy drops $\sim 1\ \%$ [Bla+20, Fig. 7]. These results do not arise when training from the start with such pruned networks [Li+16, Chap. 4], [Han+15, Chap. 3.3]. LTH gives an explanation for this effect by postulating the existence of a *winning ticket* for a randomly initialized, dense DNN $f(x; w^{(0)}), w^{(0)} \sim \mathcal{D}_w$. A winning ticket is a subnetwork $f(x; m \odot w^{(0)})$ that can be trained by itself and reach same generalization error as the full network in the same number of epochs or less. The name thus implies the existence of an initialization lottery where specific combinations of connection masks and weight prior realisations allow learning. In this context, standard pruning techniques find winning tickets by first learning the entire, dense w and then after training finding m.

LTH implies that m can be computed from a full training after which the weights can be *rewinded* to $w^{(0)}$ at which point the training $m \odot w^{(0)}$ should result in a performant network.

State of the Art

Evidence for ticket existence

LTH was presented in Frankle and Carbin [FC19] in 2019 where empirical evidence for hypothesis was presented on MNIST and CIFAR-10 by rewinding weights to initialization and comparing to random initialization. Using iterative pruning, winning tickets were found for all tried DNN's and these were found to learn faster than full networks, but for deep networks such as VGG-19, finding the tickets was sensitive to learning rate setup and required warmup steps [FC19, Chap. 4]. In follow-up work, robustness is added to this iterative search for winning tickets i [**Frankle2020LinearMC**]

HOW WINNING [**Renda2020ComparingRA**]

Optimal ticket identification

Open Problems

- Sparsity not optimal

References

[Bla+20] Davis Blalock, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John Guttag. "What is the State of Neural Network Pruning?" In: *Proceedings of Machine Learning and Systems*. Ed. by I. Dhillon, D. Papailiopoulos, and V. Sze. Vol. 2. 2020, pp. 129–146. URL: https://proceedings.mlsys.org/paper/2020/file/d2ddea18f00665ce8623e36bd4e3c7c5-Paper.pdf.

[FC19] Jonathan Frankle and Michael Carbin. "The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks". In: *International Conference on Learning Representations*. 2019. URL: https://openreview.net/forum?id=rJl-b3RcF7.

[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.

[Lan20] Robert Tjarko Lange. "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).

[LDS89] Yann LeCun, John S. Denker, and Sara A. Solla. "Optimal Brain Damage". In: *NIPS*. 1989.

[Li+16] Hao Li et al. "Pruning Filters for Efficient ConvNets". In: *CoRR* abs/1608.08710 (2016). arXiv: 1608.08710. URL: http://arxiv.org/abs/1608.08710.

[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.