Monte Carlo Winning Tickets[*]

Rafał Grzeszczuk[0000−0002−0736−9500] and Marcin Kurdziel[0000−0003−2022−7424]

AGH University of Science and Technology,
al. A. Mickiewicza 30, 30-059 Krakow, Poland
{grzeszcz,kurdziel}@agh.edu.pl

Abstract. Recent research on sparse neural networks demonstrates that densely-connected models contain sparse subnetworks that are trainable from a random initialization. Existence of these so called *winning tickets* suggests that we may possibly forego extensive training-and-pruning procedures, and train sparse neural networks from scratch. Unfortunately, winning tickets are data-derived models. That is, while they can be trained from scratch, their architecture is discovered via iterative pruning. In this work we propose Monte Carlo Winning Tickets (MCTWs) – random, sparse neural architectures that resemble winning tickets with respect to certain statistics over weights and activations. We show that MCTWs can match performance of standard winning tickets. This opens a route to constructing random but trainable sparse neural networks.

Keywords: Lottery Tickets hypothesis · Neural network initialization · Sampling.

1 Introduction

Contemporary neural network architectures tend to employ a large number of trainable parameters. In computer vision, for example, convolutional nets frequently have from over a million (as is the case in deep residual models) to tens of millions of parameters (e.g. in certain DenseNet architectures [6]). Natural language processing applications employ even larger models, with the current record held by a transformer network with 175 *billion* parameters [2]. However, available empirical evidence suggests that such large numbers of parameters are *not* a necessary prerequisite for strong performance in learned tasks. On the contrary: even though final performance in training usually increases with the number of parameters, a trained network can often be pruned of unimportant weights with virtually no performance loss. The fraction of parameters that can be pruned from a trained model is significant: performance of a dense network can often be matched by a pruned model with a tenth of the original number of weights [5]. This may, in turn, translate to major computational gains in a dedicated inference hardware, especially in energy-limited applications.

[*] Research presented in this paper was supported by the funds assigned to AGH University of Science and Technology by the Polish Ministry of Science and Higher Education. This research was supported in part by PL-Grid Infrastructure.

An immediate question arising from the empirical evidence for strong performance of pruned networks is whether we can avoid training dense models altogether. Surprisingly, at least up until recently the answer was no. Frankle and Carbin [3] show that randomly sampled sparse networks cannot be trained to match the performance of pruned networks. Previously, Han et al. [5] observed that pruned networks trained from scratch (i.e. from a random initialization) also do not converge to good solutions. Crucially, however, Frankle and Carbin also demonstrated that randomly initialized dense networks *do* contain sparse subnetworks that train well – what they call *winning tickets*. To find such subnetworks they start from dense models, which are iteratively trained, pruned and rewound to original parameter values. Each iteration remove only a small fraction of trainable weights. Ultimately, after a number of pruning iterations they uncover sparse subnetworks that train well starting from original initializations.

Winning tickets, by their construction, are data-derived subnetworks – the connectivity retained in the sparse model arises from pruning networks trained on the given data set. Furthermore, the specific initial parameter values – which can be seen as indirectly chosen via data-dependent pruning – also play an important role. In particular, randomly reinitialized winning tickets do not train as well as their counterparts with original initialization. Frankle and Carbin therefore suggest that winning tickets can possibly be seen as networks whose structures are adapted to the solved learning task. If so, then winning ticket-like networks could be uncovered only by training large, dense models, usually across many pruning iterations – a procedure with large computational cost.

In this work we explore an alternative hypothesis. Specifically, we investigate sparse neural networks with random architectures. However, we do not sample these architectures from a uniform distribution, but from a distribution that approximates certain statistics over weights and activations in an untrained winning ticket. In other words, we investigate networks that resemble untrained winning tickets with respect to certain statistics, but are otherwise randomly sampled. Our goal is to see whether such random architectures – which we call *Monte Carlo Winning Tickets* (MCWTs) – could achieve performance level close to the original, data-derived winning tickets. The main outcome from our experiments is that this may indeed be the case: we demonstrate Monte Carlo winning tickets with performance close to the iteratively pruned winning tickets. The main implication from this finding is that sparse, trainable neural networks can possibly be constructed without expensive retraining and pruning of dense models.

2 Monte Carlo Winning Tickets

We hypothesise that a sparse trainable neural networks can be constructed by sampling architectures (more precisely: network connections) from a distribution that replicates certain statistics over weights and activations in an untrained winning ticket. In this work we focus on two such statistics: magnitudes of weights and *connectivity*, which we describe below in more details.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_11

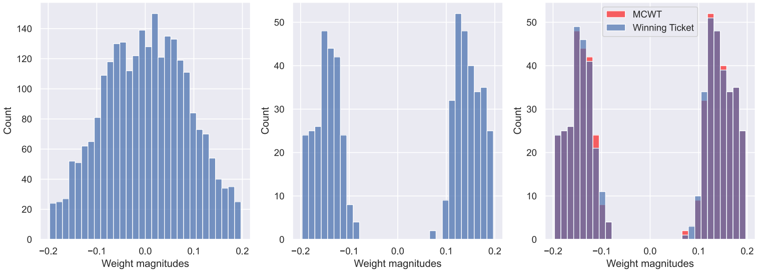


Fig. 1. Weight magnitudes in an untrained dense layer (left) and a winning ticket (center). Right: match between weight magnitudes in a winning ticket and an MCWT.

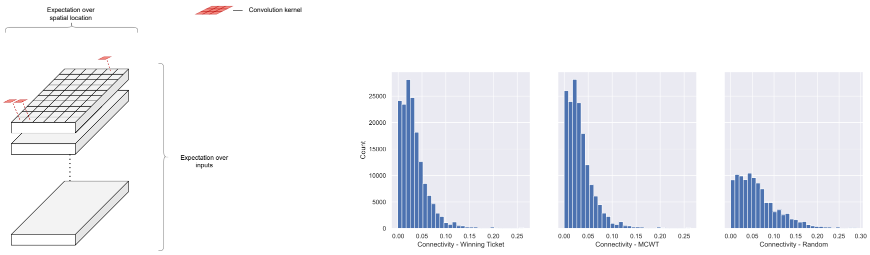
Let Φ_N be a set of parameters (i.e. weights) in neural network N, e.g. a densely connected convolutional network with certain number of layers and channels. Further, let $w^{kl}_{cij} \in \Phi_N$ be the (c, i, j) element of the k-th kernel in the l-th convolutional layer, where c enumerates input channels and (i, j) are indices over the kernel width and height, respectively. Similarly, for a fully connected layer l we will write $w^l_{ij} \in \Phi_N$ to denote the weight between the i-th neuron in layer $l - 1$ and j-th neuron in layer l. We will also write $w \in \Phi_N$ to denote a network parameter irrespective of its location in the architecture. Consider an untrained network N. If N is densely connected, or is a sparse network sampled from a uniform distribution over all possible subsets of parameters, the distribution $p\,(w \mid w \in \Phi_N)$ will simply match the density from which we sample initial parameter values. However, this will generally not be the case if N is a sparse network chosen in some non-uniform way from all possible subsets of parameters. In particular, if N is an untrained winning ticket, the parameters $w \in \Phi_N$ will be chosen by iterative training, pruning and rewinding to initial values. Thus, the conditional density $p\,(w \mid w \in \Phi_N)$ may be quite different from the density used to sample initial parameter values. Indeed, in Fig. 1 (center) we report an empirical distribution of initial parameter values retained by a winning ticket in a fully connected layer. Under uniform pruning we would expect a Gaussian distribution, which was used to initialize the layer (Fig. 1, left). However, winning tickets appears to preferentially prune weights with small initial values. We observed similar tendency in weights retained in convolutional layers.

Our first goal is to construct a randomly sampled sparse network S (Monte Carlo winning ticket) which replicates empirical distribution of weights in a genuine winning ticket. To approximately replicate this distribution, we begin with a densely connected model and sample pruning masks for its parameters from Bernoulli distributions parametrized by probabilities that depend on magnitudes of weights. That is, let $z_w \in \{0, 1\}$ be an indicator variable such that $z_w = 1$ if w is retained in S, and $z_w = 0$ otherwise. Then:

$$z_w \sim \text{Bernoulli}\left(f\left(|w|\right)\right). \tag{1}$$

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_11

To fit the acceptance probability $f(|w|)$ we fit a simple logistic regression model $f(|w|) = \sigma(u|w|)$, with parameter u, to the set $\{w \mid w \in N\}$ of weights retained in an untrained (genuine) winning ticket N. We fit two distinct models for probabilities of retaining weights: one for fully connected layers and one for convolutional layers. This sampling procedure allows us to construct sparse, random networks in which magnitudes of weights resemble those in untrained winning tickets (Fig. 1, right).



(a) (b)

Fig. 2. (a) Connectivity measure in a sparse neural network. (b) Connectivity in a standard winning ticket (left), Monte Carlo winning ticket (center) and a uniformly sampled sparse network (right).

Sampling that is driven purely by weight magnitudes disregards any connectivity structure in the sampled network. To remedy this, we introduce a simple connectivity measure illustrated in Fig. 2a. Let $\mathbf{x} \sim D$ be an input observation chosen randomly from the train set D. Further, let $a_c^{l-1}(m, n; \mathbf{x})$ be an input activation map (channel) connected to the parameter w_{cij}^{kl} and evaluated for input $\mathbf{x}$. Indices m, n enumerate spatial locations in the input channel. We define the connectivity of w_{cij}^{kl} as an expected input to that parameter in an untrained network:

$$g\left(w_{cij}^{kl}\right) = \mathbb{E}_{\substack{x \sim D \\ m,n}}\left[a_c^{l-1}(m, n; \mathbf{x})\right]. \tag{2}$$

Note that expectation is taken with respect to input examples and spatial locations in the channel connected to w_{cij}^{kl}. In practice, we approximate this expectation by averaging across all inputs in the training set. A similar connectivity measure can be defined for fully connected layers:

$$g\left(w_{ij}^{l}\right) = \mathbb{E}_{x \sim D}\left[a_{l-1}(i; \mathbf{x})\right], \tag{3}$$

where $a_{l-1}(i; \mathbf{x})$ is the i-th input to the fully connected layer l in an untrained network evaluated for $\mathbf{x}$.

Connectivity measures introduced above express the expected input to a given network parameter, which in turn reflect contributions to that input from parameters in proceeding layers. In Fig. 2b we compare distributions of connectivity measure in an untrained standard winning ticket (left) and a uniformly sampled sparse network (right). Note a substantial difference between the two distributions. We leverage this discrepancy to incorporate connectivity information into the sampling model for Monte Carlo winning tickets. That is, we extend the magnitude-base sampling (Eq. 1) to account for $g\left(w\right)$:

$$\begin{aligned} & z_w \sim \text{Bernoulli}\left(f\left(|w|, g\left(w\right)\right)\right), \\ & f\left(|w|, g\left(w\right)\right) = \sigma\left(u_1|w| + u_2 g\left(w\right)\right). \end{aligned} \tag{4}$$

Again, we use a simple logistic regression model for the acceptance probability (with parameters $u_1,\ u_2$), and fit it to the magnitudes of weights and connectivities observed in a genuine winning ticket. The resultant connectivity distribution in a Monte Carlo winning ticket is pictures in Fig. 2b (center). To estimate connectivity in a layer l we must known which parameters from the proceeding layers are retained. Thus, we sample Monte Carlo wining tickets in a greedy layer-wise way, by moving from the network input towards the last layer. In the first layer we account only for weight magnitudes. In subsequent layers we also estimate connectivity values.

3 Results

Experiments in this work follow the setup used in Frankle and Carbin [3]. Specifically, we evaluate MCWTs on *Conv2*, *Conv4* and *Conv6* architectures used therein, and use hyper-parameter values reported in that work. In each case, we prune 80% of weights in every network layer using either random pruning, iterative pruning described by Frankle and Carbin [3] (i.e. winning tickets) or Monte Carlo method described in the previous section.

We use CIFAR-10 dataset [7] to evaluate MCWTs. It consists of 60,000 images from 10 classes. Of these images, 50,000 are in the train set and the remaining 10,000 in the test set. All images are 32 × 32 pixels in size. To follow the experimental setup from [3], we use only the 50,000 train samples. Specifically, we train networks on 45,000 examples and evaluate on the remaining 5,000. Even though we train for up to 300-500 epochs, MCWTs learn faster, achieving near-final performance after 10-20 epochs.

Results from our experiments are reported in Table 1. The main finding here is that Monte Carlo winning tickets allow for low-computational-cost discovery of sparse trainable neural network architectures. Specifically, we observe similar performance for original winning tickets and our Monte Carlo winning tickets. These results are slightly above unpruned networks and significantly outperform baseline, i.e. random pruning. In Monte Carlo winning tickets most of the improvement comes from fitting distributions of initial weights, with connectivity playing a less important role. For *Conv4* and *Conv6* models our approach shows modest improvement over original winning tickets, while for smaller *Conv2* architecture it exhibit slightly worse, but still similar performance. Note, however, that it is often impossible to exactly replicate experimental conditions in deep learning, due to missing data on some hyper-parameters or differences in software

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_11

Experiment	**Conv2**	**Conv4**	**Conv6**
Unpruned network	67.20%	74.62%	77.40%
Randomly pruned network	66.90%	73.20%	73.14%
Original Winning Ticket	**70.42**%	74.92%	78.04%
Monte Carlo Winning Ticket - w/o Connectivity	69.47%	**76.39**%	78.10%
Monte Carlo Winning Ticket - Connectivity	68.40%	76.21%	**78.17**%

Table 1. Experimental results for networks with 80% pruning. To facilitate comparison, we reproduced the original Winning Ticket results in our code. We also compare against random pruning and unpruned networks.

versions. As a result, we observed higher performance for unpruned networks and slightly better results for the original winning tickets, than those reported in [3].

4 Related work

The starting point for the research presented in this work was the Lottery Tickets Hypothesis formulated by Frankle and Carbin [3]. From a practical perspective it presents an iterative pruning mechanism for discovering sparse, trainable neural networks. The main disadvantage of this approach is the necessity to train the network several times, each time increasing the number of pruned parameters by a certain, architecture-dependant factor. Zhou et al. [9] conduct further research on this subject and introduce the concept of *supermasks*. They show that choosing the pruning masks with a specific, carefully designed criteria can lead to significantly better-than-chance performance of the randomly-initialized sparse network. Our research shows that for good results, we can simply sample the connections in the sparse model in such a way that they resemble winning tickets with respect to certain statistics over magnitudes of weights and network connectivity. Frankle et al. [4] further develop their technique for finding winning tickets, thereby enabling its use on more complex datasets, such as ImageNet. Most importantly they show that it is easier to start pruning networks that are trained for a few epochs, rather than rewinding them all the way to the initial parameter values. This suggest an avenue for further research on Monte Carlo Winning tickets, namely investigation of performance of models sampled using our approach on ImageNet-scale datasets. Concurrently to work on Monte Carlo winning tickets, Blalock et al. [1] suggested alternative ways to assess neural network pruning. However, their findings cannot be applied directly to our work, because we compare two networks with the same architecture and pruning ratio. Finally, Xie et al. [8] investigated randomly wired neural networks constructed using random graph models. They obtained competitive performance in computer vision tasks.

5 Conclusions

In this work we presented Monte Carlo winning tickets. These sparse neural networks are constructed by sampling connections from a probability distribution that replicates statistics over weights and connectivity in standard winning tickets. We demonstrated that Monte Carlo winning tickets are trainable from scratch, i.e. they train to a performance level matching standard winning tickets and densely connected model. Typically, this level of performance used to be achieved by training and then pruning a densely connected model. Thus, Monte Carlo winning tickets open an avenue to lower the computational cost of deploying sparse neural nets.

Bibliography

[1] Blalock, D., Ortiz, J.J.G., Frankle, J., Guttag, J.: What is the state of neural network pruning? arXiv preprint arXiv:2003.03033 (2020)

[2] Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners. In: Advances in Neural Information Processing Systems 33 (2020)

[3] Frankle, J., Carbin, M.: The lottery ticket hypothesis: Finding sparse, trainable neural networks. In: 7th International Conference on Learning Representations, ICLR 2019 (2019), https://arxiv.org/pdf/1803.03635.pdf

[4] Frankle, J., Dziugaite, G.K., Roy, D.M., Carbin, M.: Stabilizing the lottery ticket hypothesis. arXiv preprint arXiv:1903.01611 (2019)

[5] Han, S., Pool, J., Tran, J., Dally, W.J.: Learning both weights and connections for efficient neural networks. In: Advances in Neural Information Processing Systems 28. pp. 1135–1143 (2015)

[6] Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2261–2269. IEEE Computer Society (2017)

[7] Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)

[8] Xie, S., Kirillov, A., Girshick, R.B., He, K.: Exploring randomly wired neural networks for image recognition. In: 2019 IEEE/CVF International Conference on Computer Vision, ICCV. pp. 1284–1293. IEEE (2019)

[9] Zhou, H., Lan, J., Liu, R., Yosinski, J.: Deconstructing lottery tickets: Zeros, signs, and the supermask. arXiv preprint arXiv:1905.01067 (2019)