Pruning and The Lottery Ticket Hypothesis

Roberto Halpin Gregorio

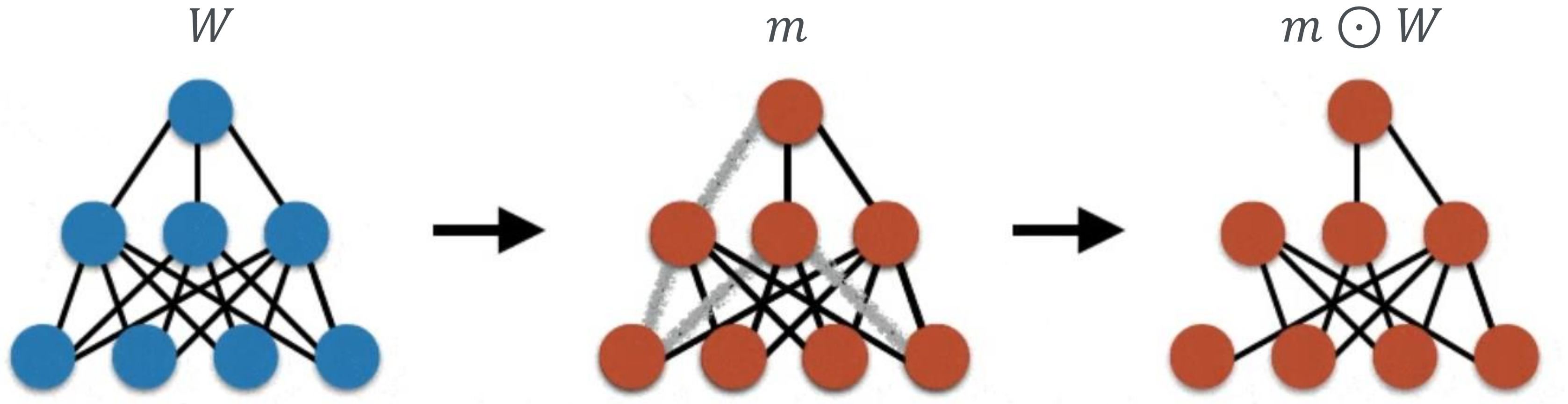
Quiz

1. Which structure is being removed/pruned in the Lottery Ticket Hypothesis paper?
   A. Individual weights
   B. Entire neurons
   C. Entire filters
   D. Entire channels

2. Which heuristic is used to prune in the Lottery Ticket Hypothesis paper?
   A. Magnitudes
   B. Gradients
   C. Activations

Network Pruning

- Most commonly it refers to setting a particular weight to 0 and freezing it for the course of any subsequent training.
- This can easily be done by element-wise multiplying the weights W with a binary pruning mask m.



Network Pruning

Algorithm 1 Pruning and Fine-Tuning

Input: N, the number of iterations of pruning, and
X, the dataset on which to train and fine-tune

$W \leftarrow initialize()$
$W \leftarrow trainToConvergence(f(X; W))$
$M \leftarrow 1^{|W|}$
for i in 1 to N **do**
    $M \leftarrow prune(M, score(W))$
    $W \leftarrow fineTune(f(X; M \odot W))$
end for
return M, W

Network Pruning

Algorithm 1 Pruning and Fine-Tuning

Input: N, the number of iterations of pruning, and
X, the dataset on which to train and fine-tune

$W \leftarrow initialize()$
$W \leftarrow trainToConvergence(f(X; W))$
$M \leftarrow 1^{|W|}$
for i in 1 to N **do**
  $M \leftarrow prune(M, score(W))$
  $W \leftarrow fineTune(f(X; M \odot W))$
end for
return M, W

What is Pruning?

- Remove superfluous structure

From [Frankle ICLR 2019 talk]

What is Pruning?

- Remove superfluous structure

Weights? Neurons? Filters? Channels?

From Frankle ICLR 2019 talk

What is Pruning?

- Remove superfluous structure

Magnitudes? Gradients? Activations?

From Frankle ICLR 2019 talk

What is Pruning?

- Remove superfluous structure

Magnitudes? Gradients? Activations?

- The pruned network can represent an equally accurate function.

From Frankle ICLR 2019 talk

Motivation

- It supports generalization by regularizing overparameterized functions.
- It reduces the memory constraints during inference time by identifying well-performing smaller networks which can fit in memory.
- It reduces energy costs, computations, storage and latency which can all support deployment on mobile devices.

The Big Questions

- Can we train (sparsely) pruned networks from scratch?

- Corollary: Do networks have to be so overparameterized to learn?

From Frankle ICLR 2019 talk

The Big Questions

- Can we train (sparsely) pruned networks from scratch? **Yes**

- Corollary: Do networks have to be so overparameterized to learn? **No**

From [Frankle ICLR 2019 talk]

Lottery Ticket Hypothesis

Weights pruned after training could have been pruned before training*.

*Need to use the same initializations.

Lottery Ticket Hypothesis

Weights pruned after training could have been pruned before training*.

*Need to use the same initializations.

The Lottery Ticket Hypothesis*: A randomly-initialized, dense neural network contains a subnetwork that is initialised such that — when trained in isolation — it can match the test accuracy of the original network after training for at most the same number of iterations. - Frankle & Carbin (2019, p.2)*

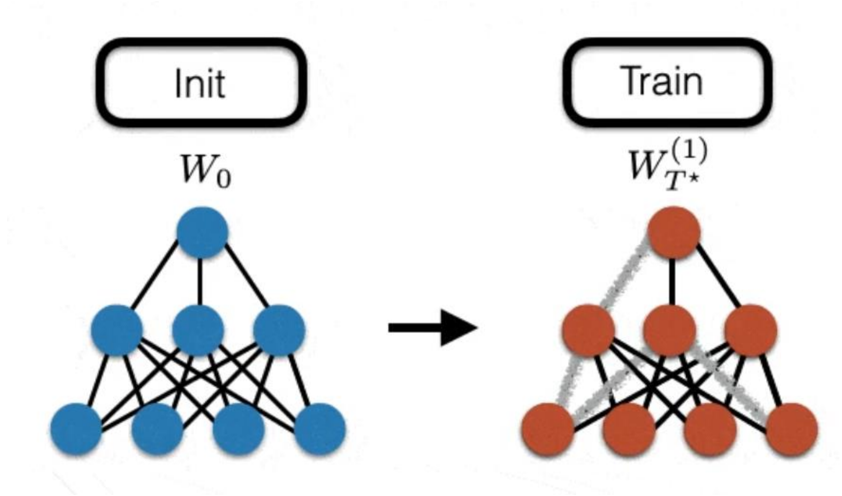
How to train pruned networks successfully from scratch?

- LTH paper observes success using a method called Iterative Magnitude Pruning (IMP).
  - Prune individual weights based on their magnitude
  - Reset each unpruned connection back to its initial value from before training

- **Definition** (Winning ticket).
  - When IMP produces a subnetwork of the original, untrained network that matches the accuracy of the original network, it is called a winning ticket.

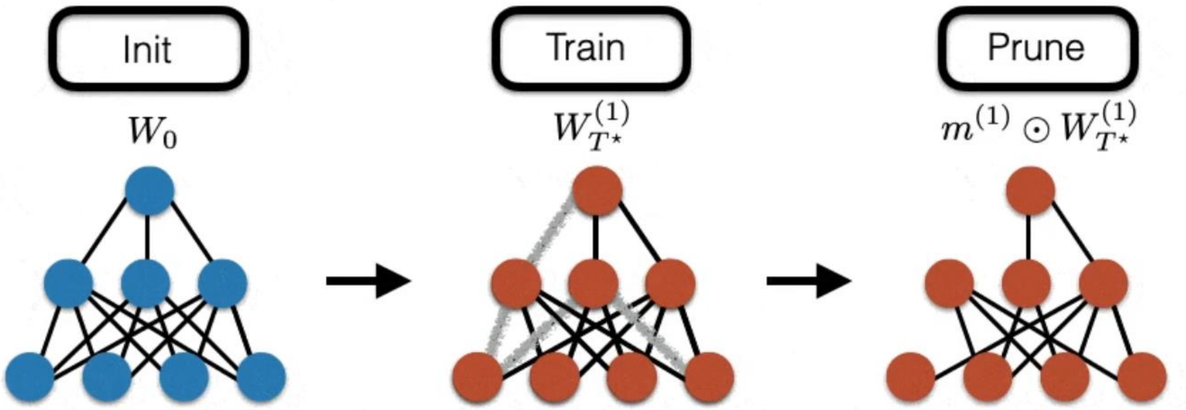
Iterative Magnitude Pruning (IMP)

- Starting from dense initialization W_0, train network until convergence: $W_{T^*}^{(1)}$.



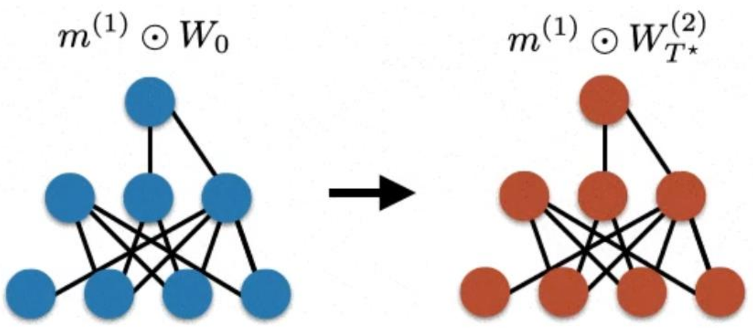
Iterative Magnitude Pruning (IMP)

- Starting from dense initialization W_0, train network until convergence: $W_{T^*}^{(1)}$.
- Determine the s percent smallest magnitude weights and create a binary mask $m^{(1)}$ that prunes these.



Iterative Magnitude Pruning (IMP)

- Starting from dense initialization W_0, train network until convergence: $W_{T^*}^{(1)}$.
- Determine the s percent smallest magnitude weights and create a binary mask $m^{(1)}$ that prunes these.
- Retrain the sparsified network with its **previous initial weight** $m^{(1)} \odot W_0$.



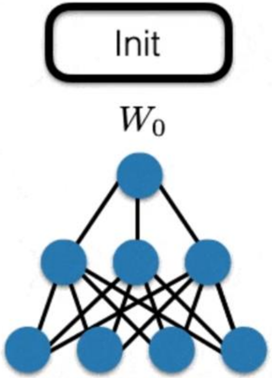
Iterative Magnitude Pruning (IMP)

- Starting from dense initialization W_0, train network until convergence: $W_{T^*}^{(1)}$.
- Determine the s percent smallest magnitude weights and create a binary mask $m^{(1)}$ that prunes these.
- Retrain the sparsified network with its **previous initial weight** $m^{(1)} \odot W_0$.
- After convergence, repeat the pruning process and reset to the initial weights with the newly found mask $m^{(2)}$.

Iterative Magnitude Pruning (IMP)

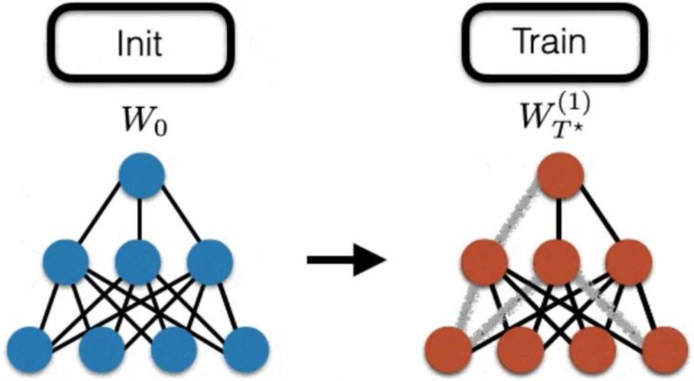
- Starting from dense initialization W_0, train network until convergence: $W_{T^*}^{(1)}$.
- Determine the s percent smallest magnitude weights and create a binary mask $m^{(1)}$ that prunes these.
- Retrain the sparsified network with its **previous initial weight** $m^{(1)} \odot W_0$.
- After convergence, repeat the pruning process and reset to the initial weights with the newly found mask $m^{(2)}$.
- Iterate this process until we reach the desired level of sparsity or the test accuracy drops significantly.

Searching for Tickets: Iterative Magnitude Pruning



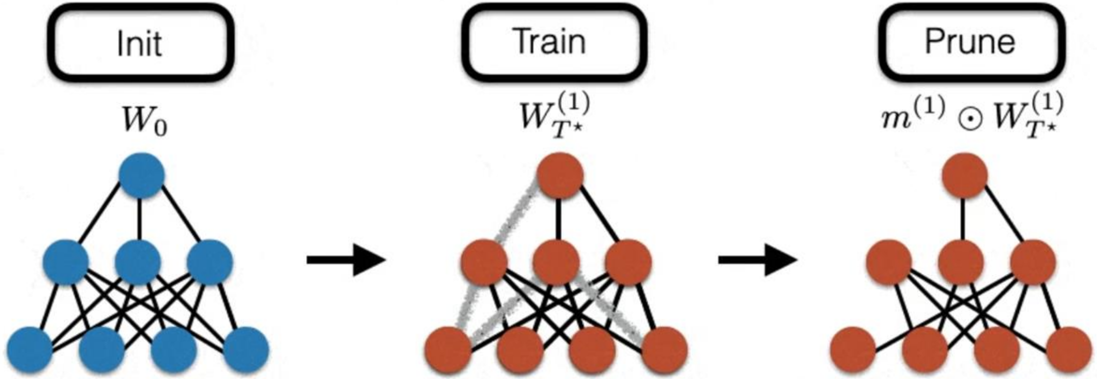
Frankle & Carbin, 2019
Viz: @RobertTLange

Searching for Tickets: Iterative Magnitude Pruning



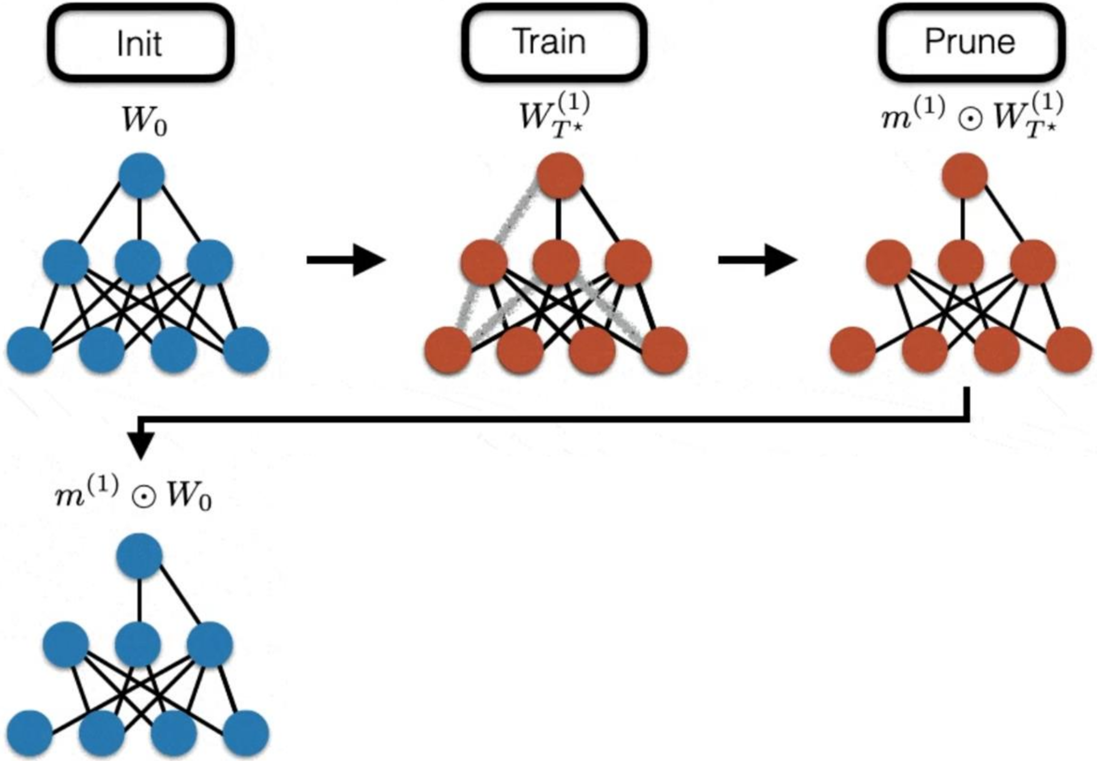
Frankle & Carbin, 2019
Viz: @RobertTLange

Searching for Tickets: Iterative Magnitude Pruning



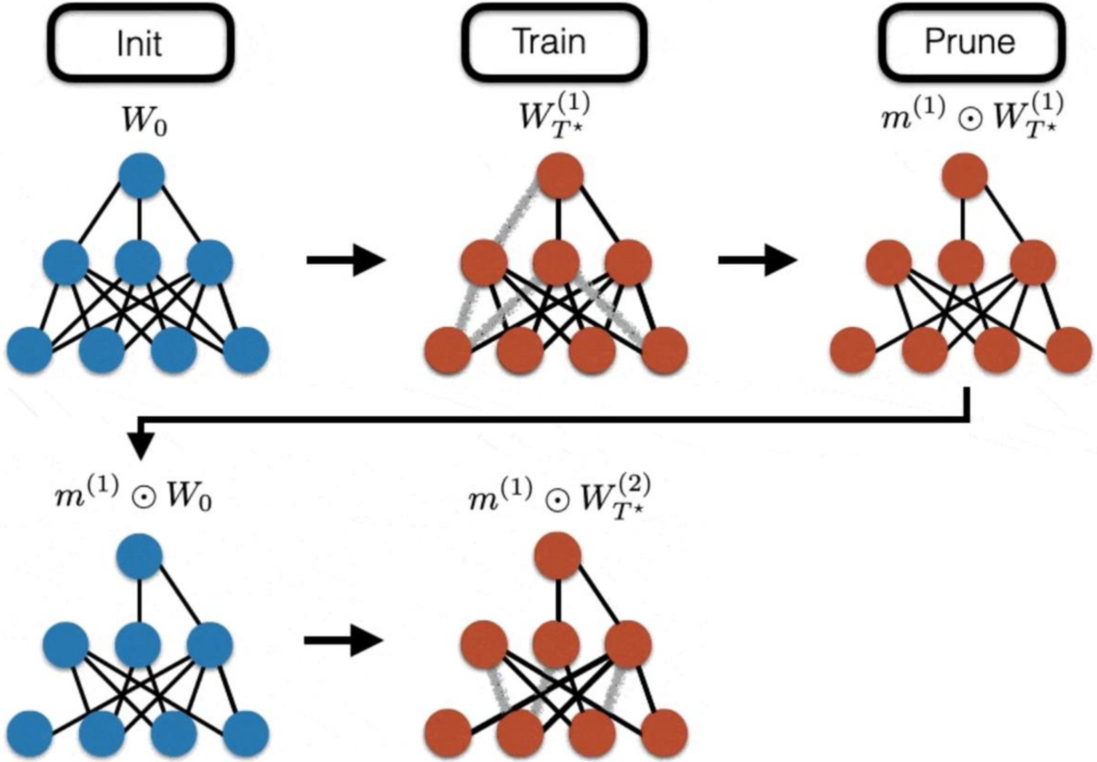
Frankle & Carbin, 2019
Viz: @RobertTLange

Searching for Tickets: Iterative Magnitude Pruning



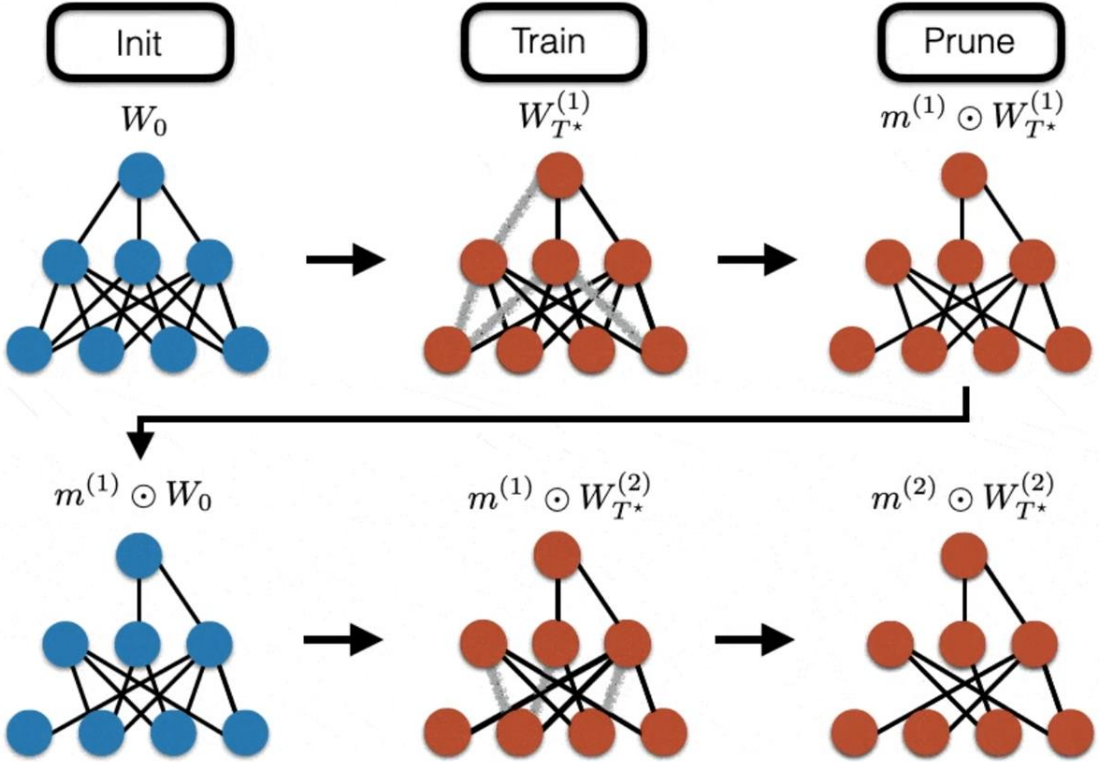
Frankle & Carbin, 2019
Viz: @RobertTLange

Searching for Tickets: Iterative Magnitude Pruning



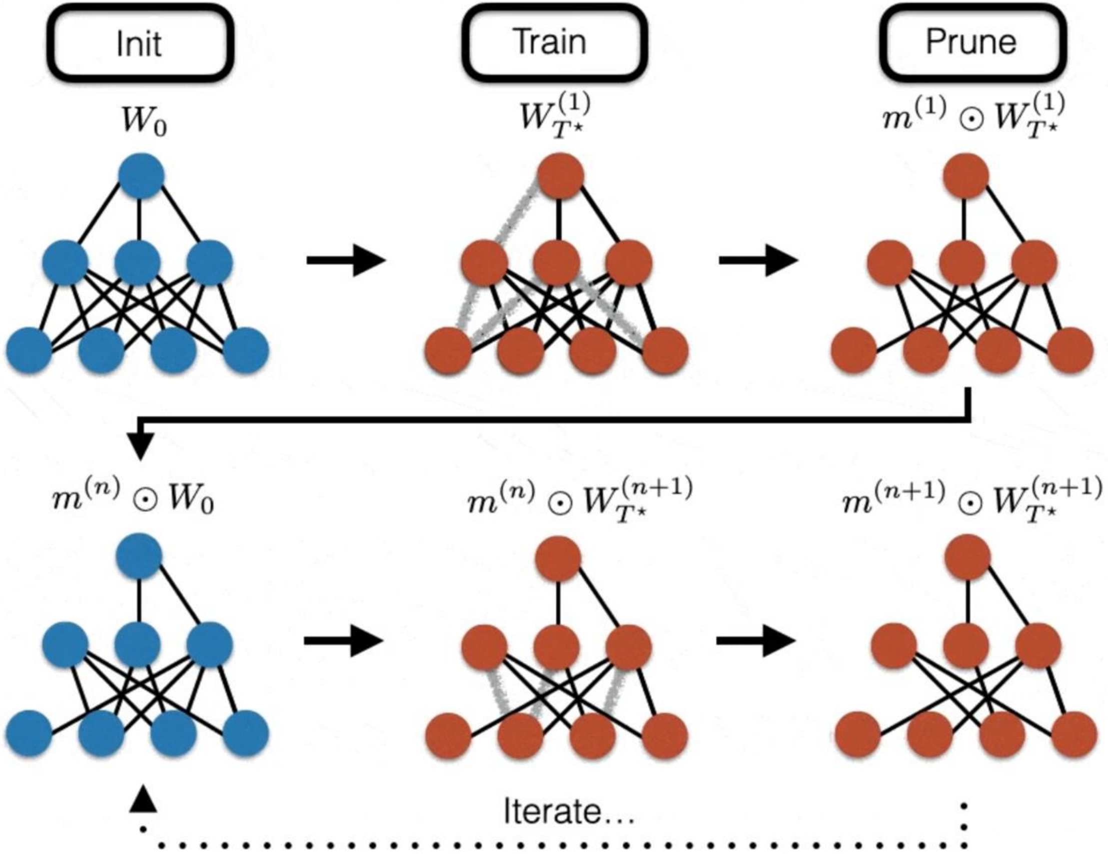
Frankle & Carbin, 2019
Viz: @RobertTLange

Searching for Tickets: Iterative Magnitude Pruning



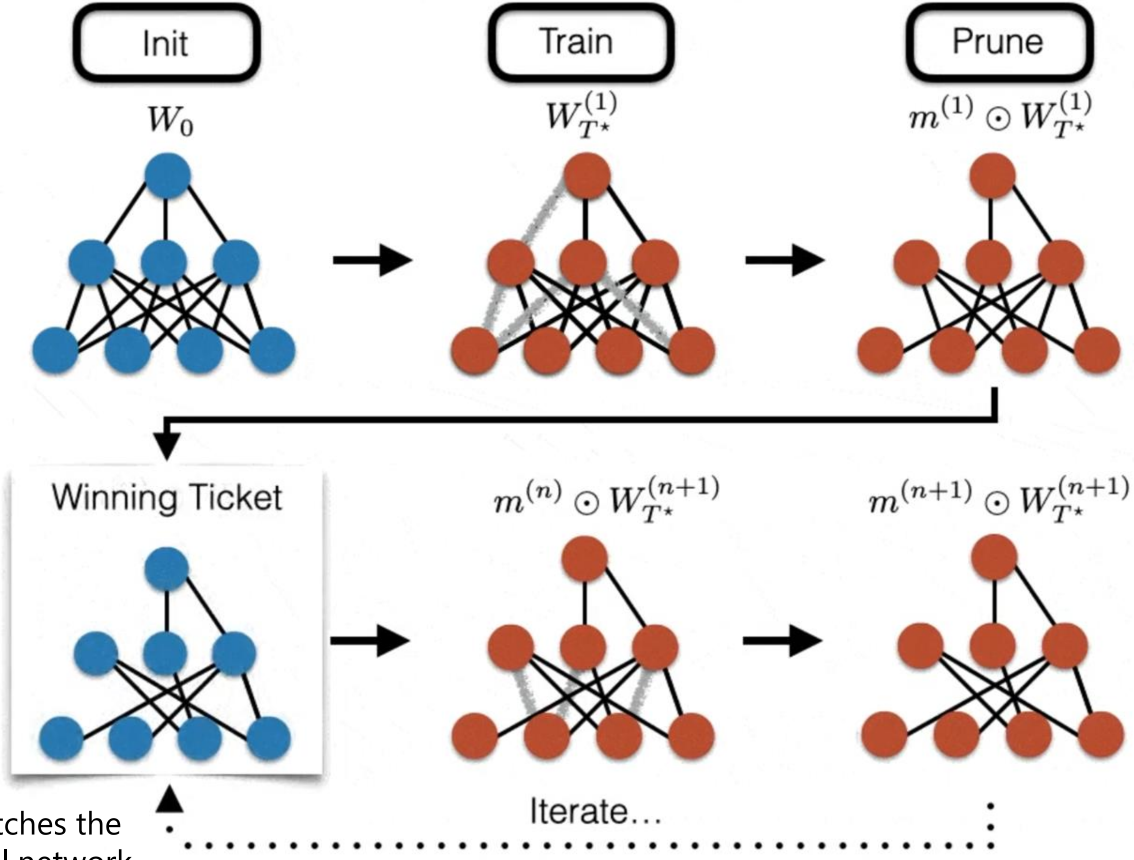
Frankle & Carbin, 2019
Viz: @RobertTLange

Searching for Tickets: Iterative Magnitude Pruning



Frankle & Carbin, 2019
Viz: @RobertTLange

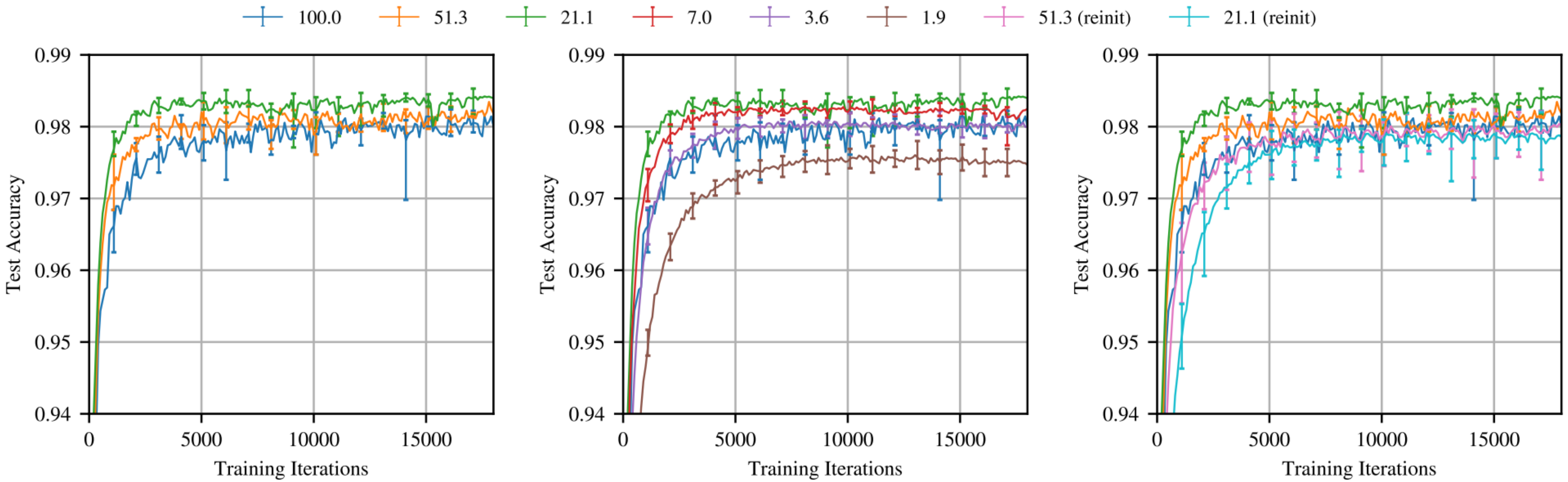
Searching for Tickets: Iterative Magnitude Pruning



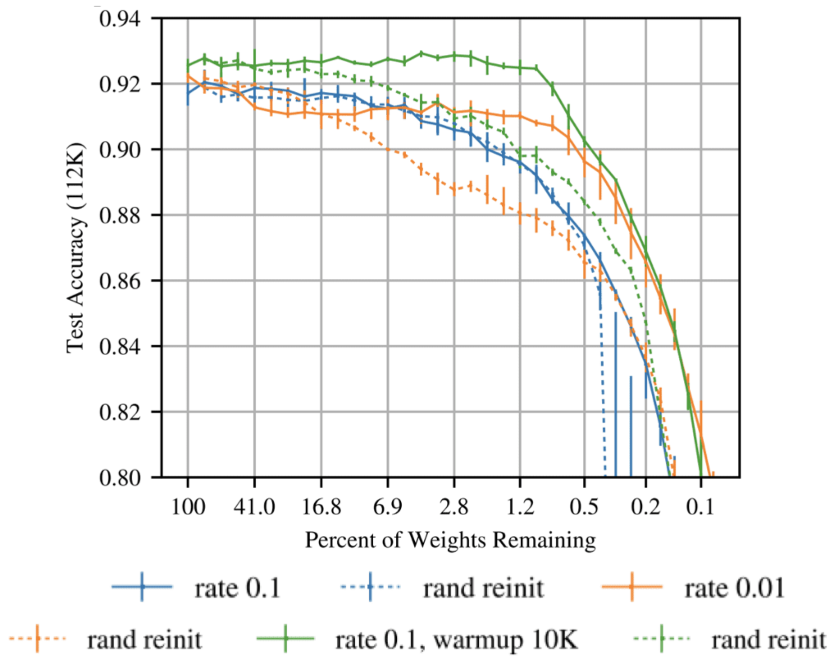
A subnetwork that matches the accuracy of the original network.

Frankle & Carbin, 2019
Viz: @RobertTLange

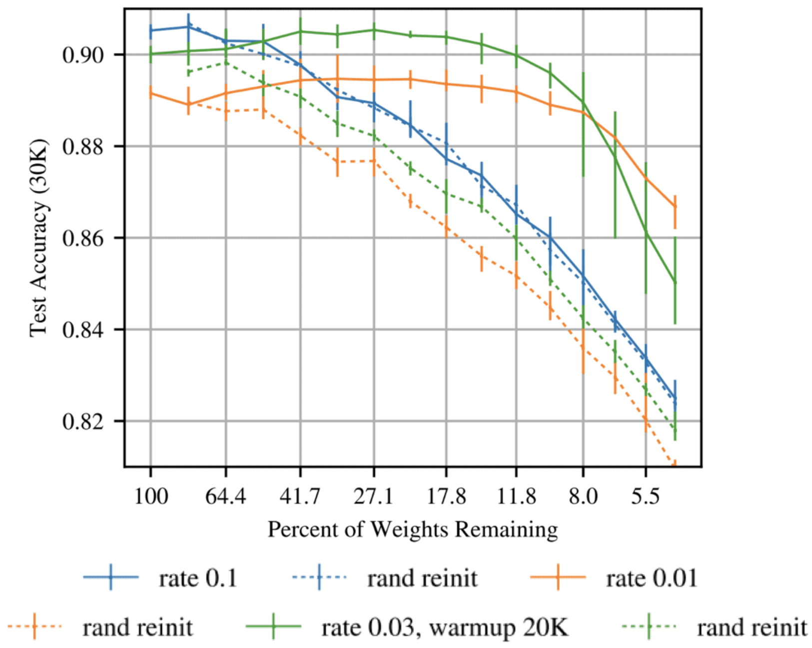
Lottery Tickets – LeNet on MNIST



Lottery Tickets
- VGG-19 on CIFAR-10 -



Lottery Tickets
- ResNet-20 on CIFAR-10 -



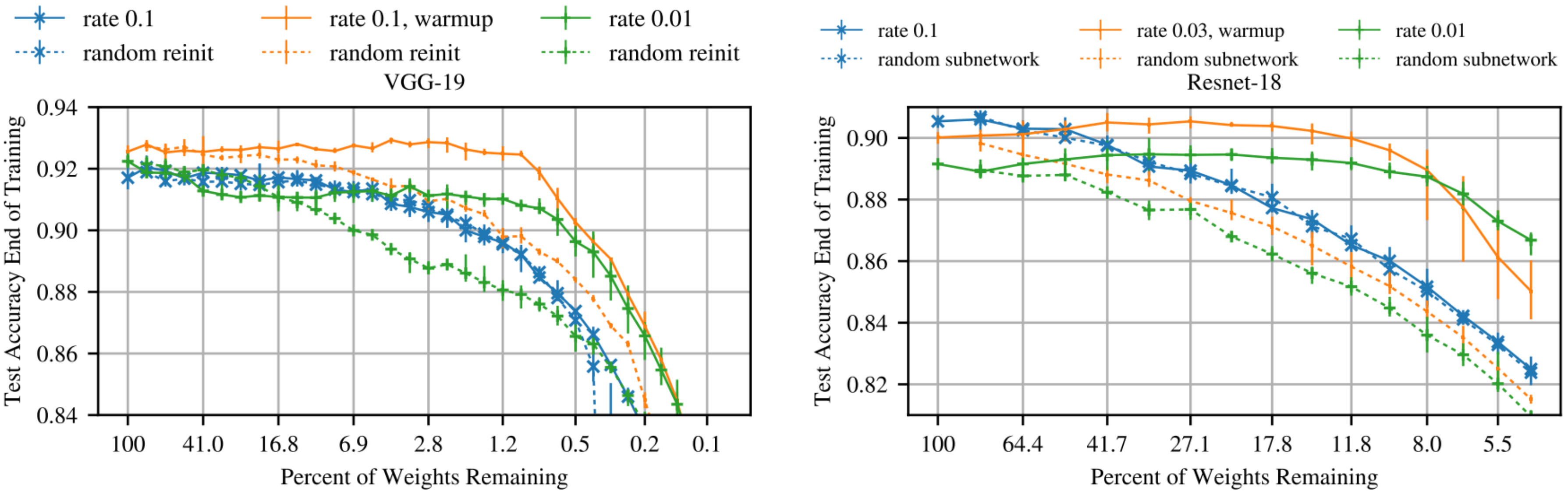
Implications

- Pruning neural networks early in training.
- Examine subnetworks to develop better architectures and initializations.
- Reuse subnetworks on new tasks.

Limitations of Iterative Magnitude Pruning

- In order to scale the LTH to competitive CIFAR-10 architectures , Frankle & Carbin (2019) had to tune learning rate schedules.

- Further work (Liu et al. 2018; Gale et al. 2019) show that without this adjustment it is not possible to obtain a pruned network that is on par with the original dense network.



On deeper networks for CIFAR10, IMP fails to find winning tickets unless the learning rate schedule is altered.

How severe is this limitation?

The Fix: Rewinding

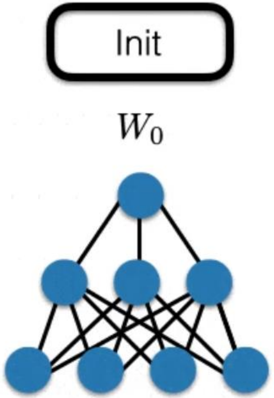
- Instead of rewinding to iteration 0 after pruning, we rewind to iteration k.

The Fix: Rewinding

- Instead of rewinding to iteration 0 after pruning, we rewind to iteration k.

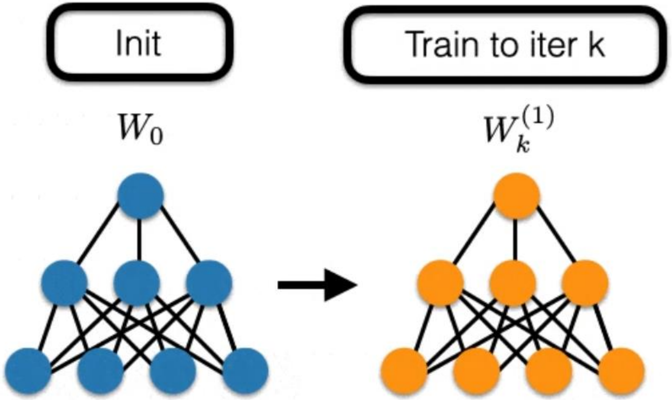
The Lottery Ticket Hypothesis with Rewinding: *Consider a dense, randomly-initialized neural network* $f(x; W_0)$ *that trains to accuracy* $a^\star$ *in* $T^\star$ *iterations. Let* W_t *be the weights at iteration* t *of training. There exist an iteration* k«$T^\star$ *and fixed pruning mask* $m \in \{0,1\}^{|W_0|}$ *(where* $|| \ m \ ||_1$ « $| \ W_0|$*) such that subnetwork* $m \odot W_k$ *trains to accuracy* $a \geq a^\star$ *in* $T \leq T^\star - k$ *iterations. - Frankle et al. (2019, p. 2)*

Iterative Magnitude Pruning with Rewinding



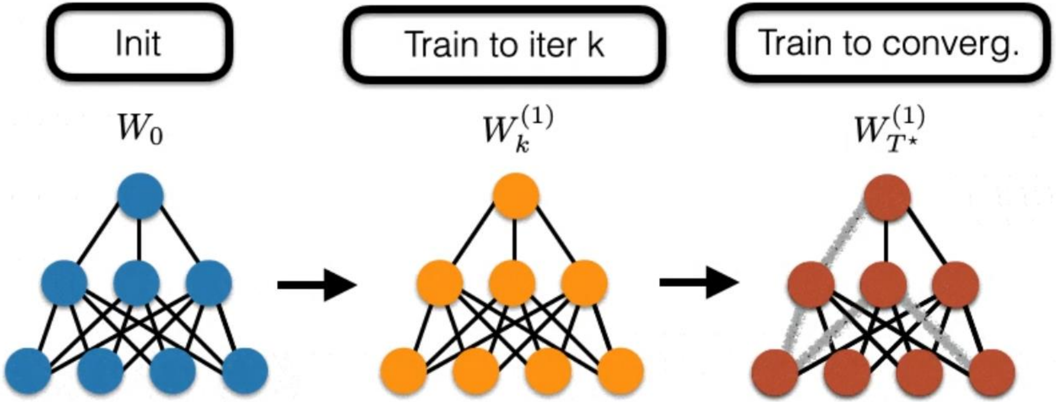
Frankle et al., 2019
Viz: @RobertTLange

Iterative Magnitude Pruning with Rewinding



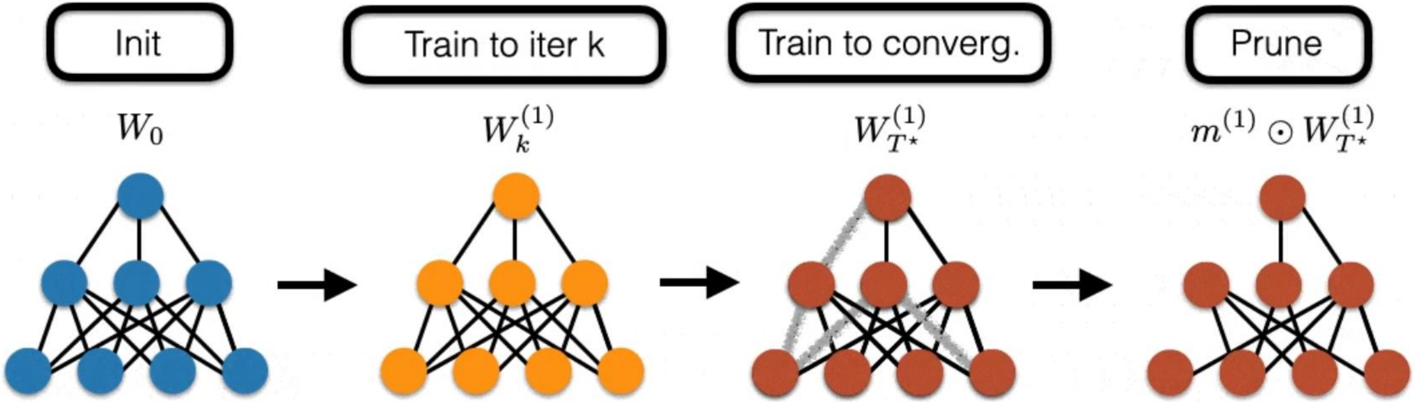
Frankle et al., 2019
Viz: @RobertTLange

Iterative Magnitude Pruning with Rewinding



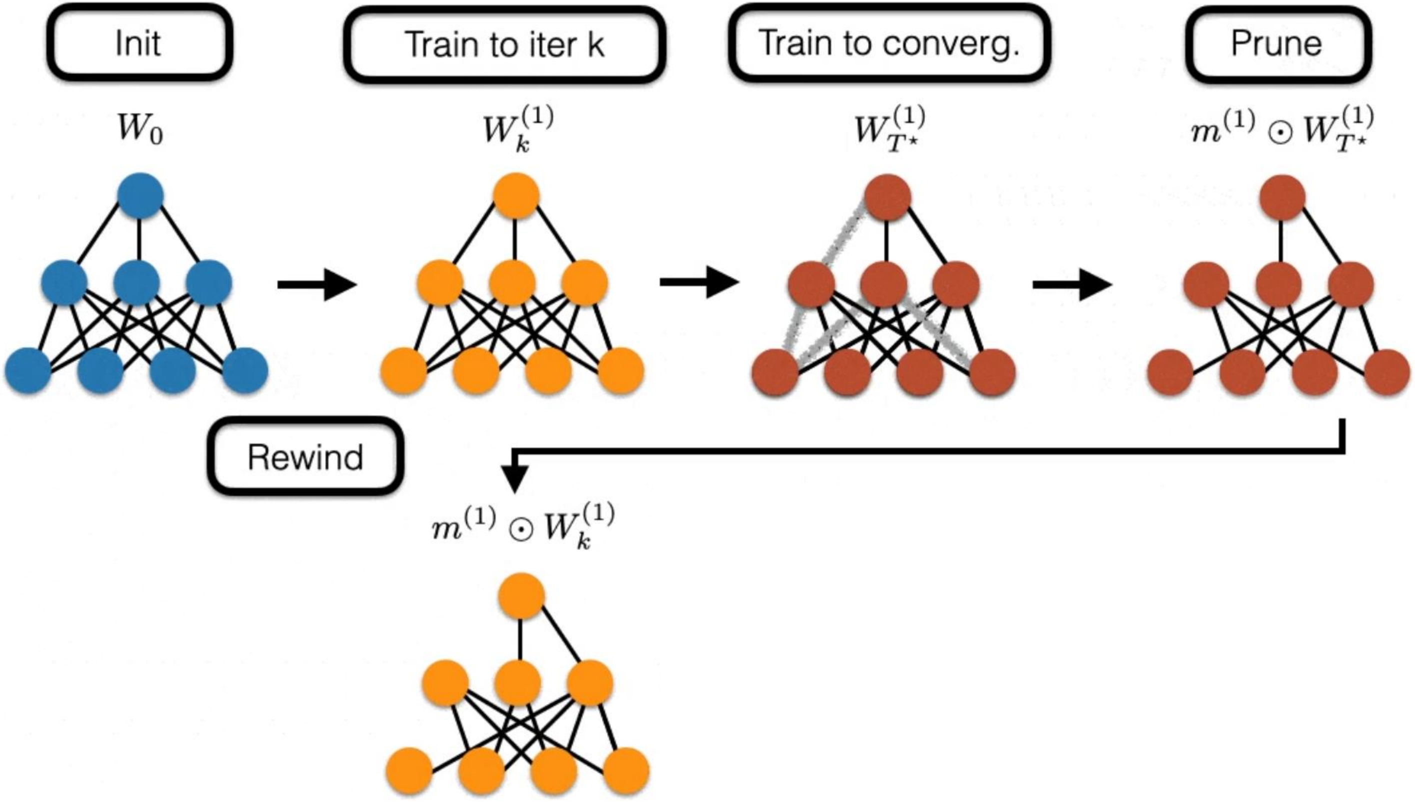
Frankle et al., 2019
Viz: @RobertTLange

Iterative Magnitude Pruning with Rewinding



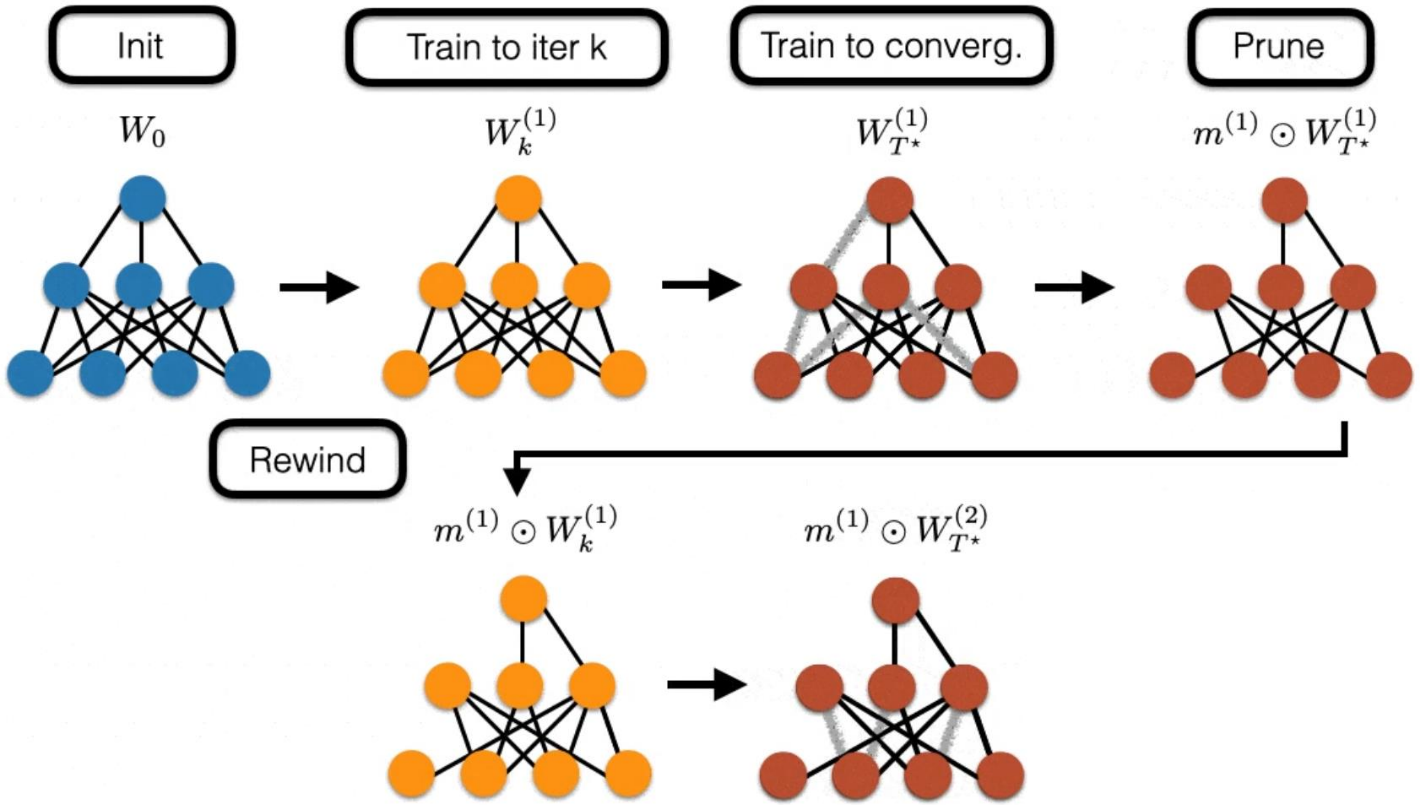
Frankle et al., 2019
Viz: @RobertTLange

Iterative Magnitude Pruning with Rewinding



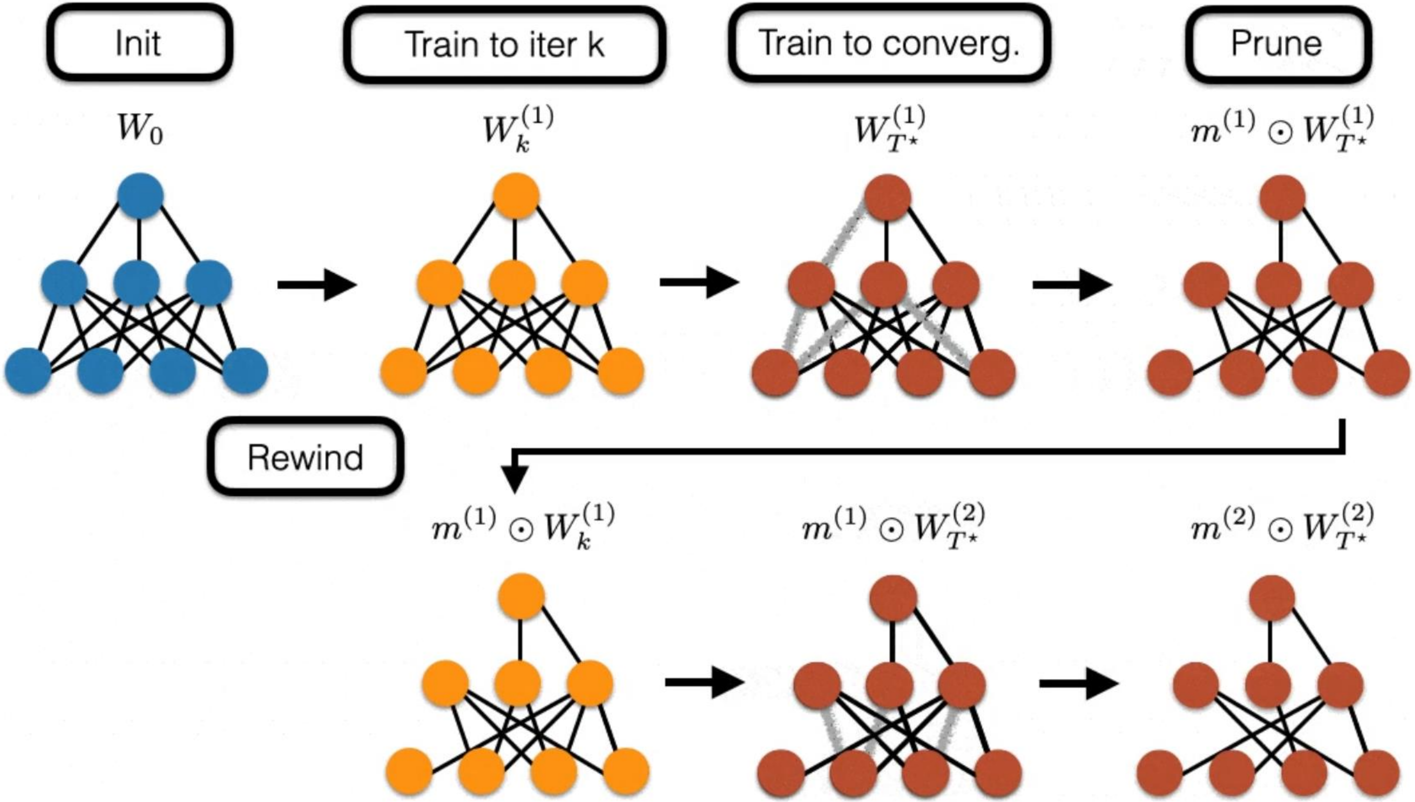
Frankle et al., 2019
Viz: @RobertTLange

Iterative Magnitude Pruning with Rewinding



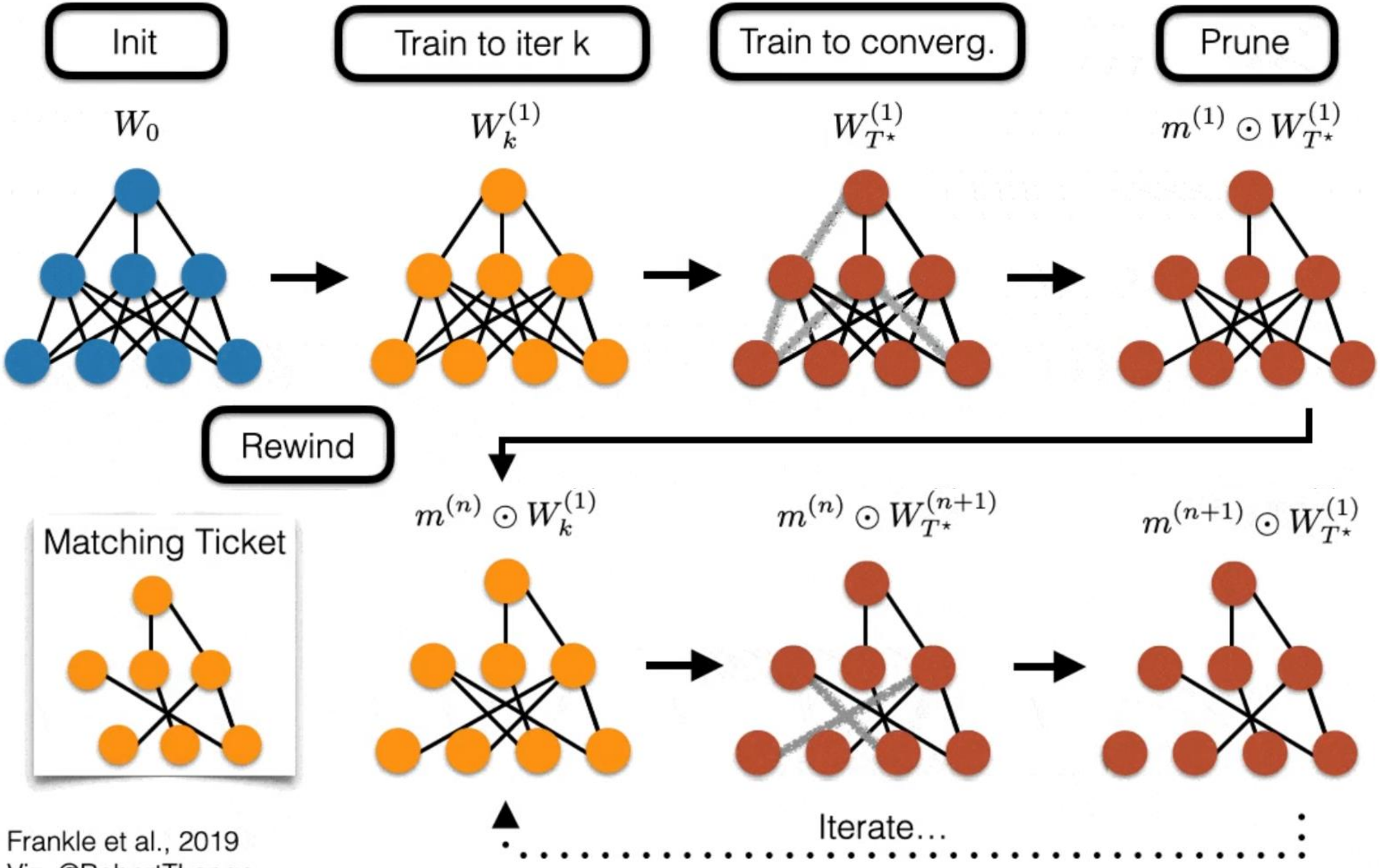
Frankle et al., 2019
Viz: @RobertTLange

Iterative Magnitude Pruning with Rewinding

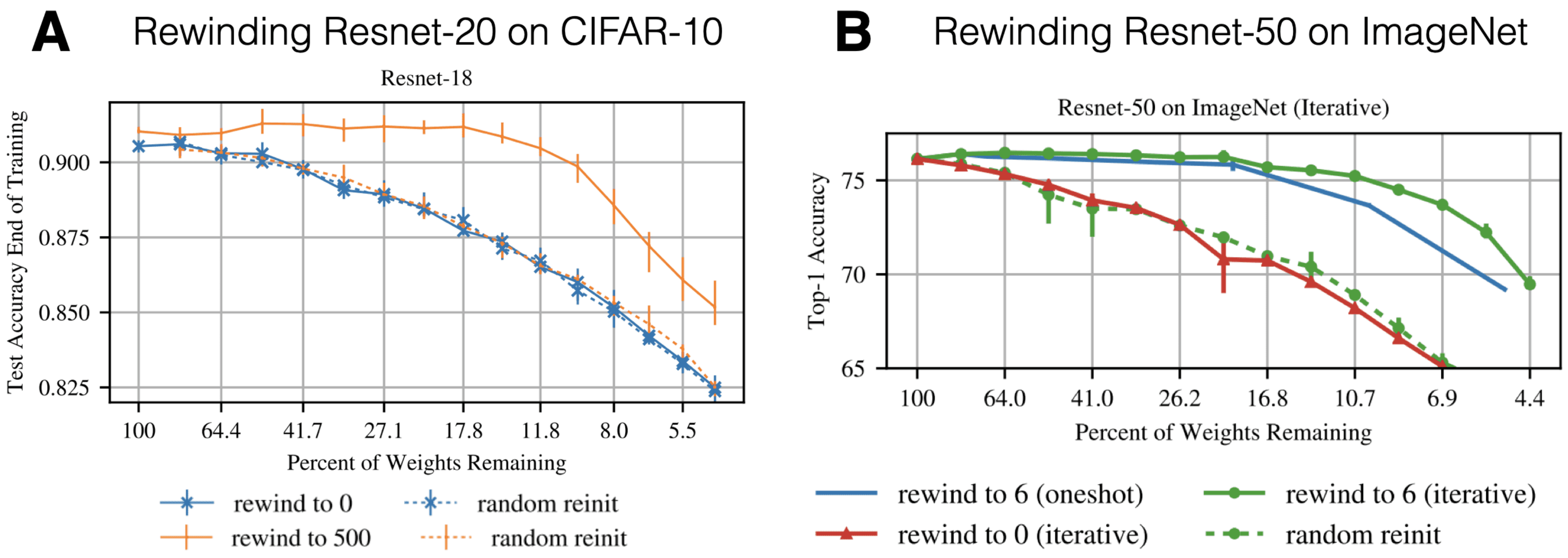


Frankle et al., 2019
Viz: @RobertTLange

Iterative Magnitude Pruning with Rewinding

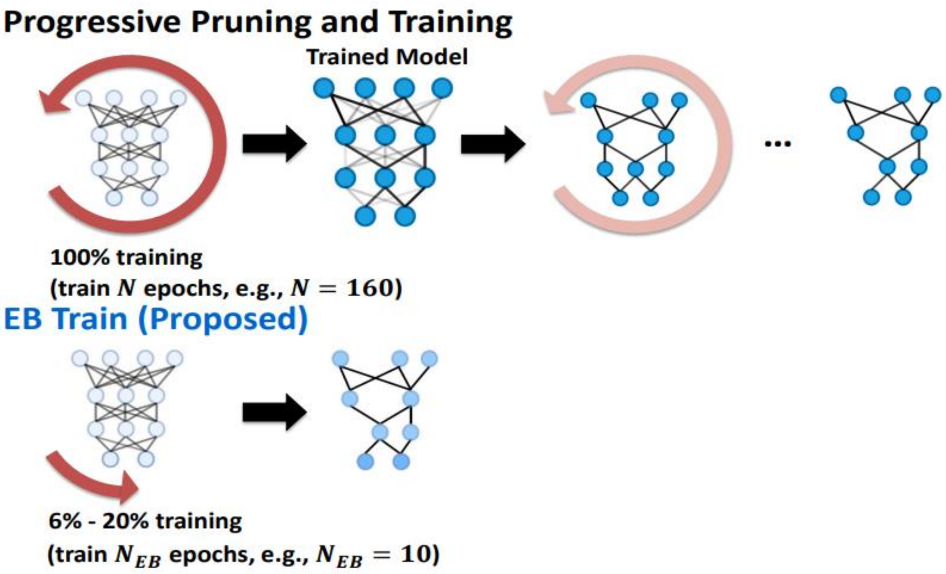


Frankle et al., 2019
Viz: @RobertTLange


A
Rewinding Resnet-20 on CIFAR-10
Resnet-18
Test Accuracy End of Training
0.900
0.875
0.850
0.825
100
64.4
41.7
27.1
17.8
11.8
8.0
5.5
Percent of Weights Remaining
rewind to 0
random reinit
rewind to 500
random reinit
B
Rewinding Resnet-50 on ImageNet
Resnet-50 on ImageNet (Iterative)
Top-1 Accuracy
75
70
65
100
64.0
41.0
26.2
16.8
10.7
6.9
4.4
Percent of Weights Remaining
rewind to 6 (oneshot)
rewind to 6 (iterative)
rewind to 0 (iterative)
random reinit

Early-Bird Tickets

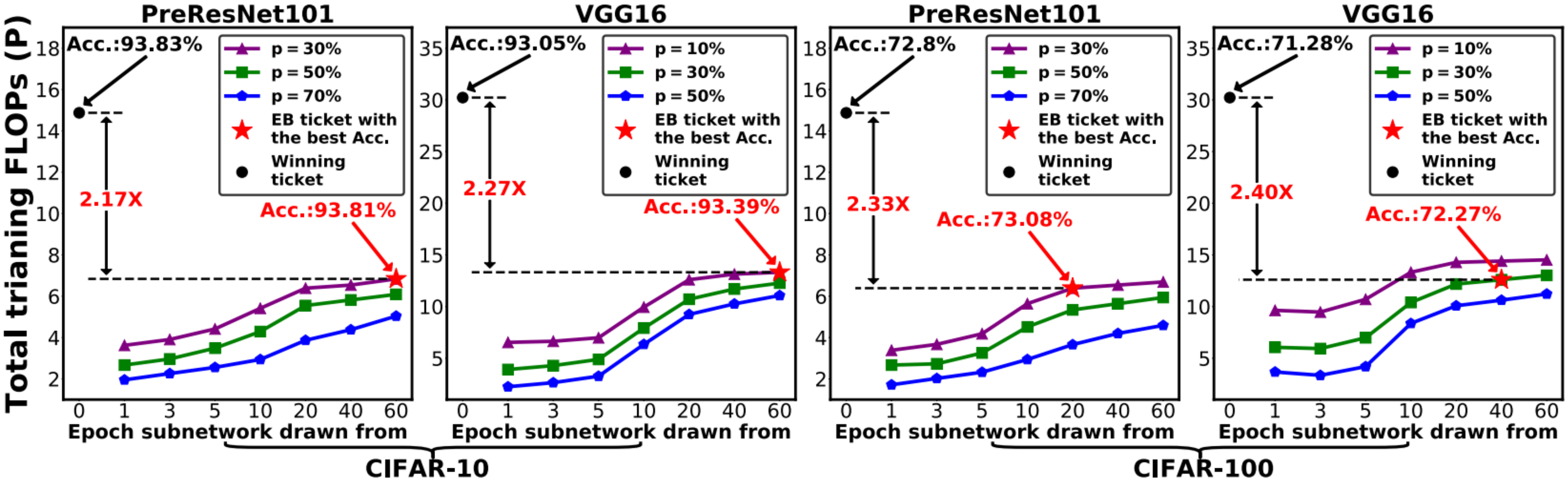
- You et al., 2020 identify winning tickets early on in training using a low-cost training scheme.
  - early stopping
  - low precision
  - large learning rates



Early-Bird Tickets

Algorithm 1: The Algorithm for Searching EB Tickets

```
Initialize the weights W, scaling factor r, pruning ratio p, and the FIFO queue Q with length l;
while t (epoch) < t_max do
    Update W and r using SGD training;
    Perform structured pruning based on r_t towards the target ratio p;
    Calculate the mask distance between the current and last subnetworks and add to Q.
    t = t + 1
    if Max(Q) < ϵ then
        t* = t
        Return f(x; m_{t*} ⊙ W) (EB ticket);
    end if
end while
```


PreResNet101
VGG16
PreResNet101
VGG16
Total trianing FLOPs (P)
Acc.:93.83%
Acc.:93.05%
Acc.:72.8%
Acc.:71.28%
p = 30%
p = 50%
p = 70%
p = 10%
p = 30%
p = 50%
EB ticket with the best Acc.
Winning ticket
2.17X
2.27X
2.33X
2.40X
Acc.:93.81%
Acc.:93.39%
Acc.:73.08%
Acc.:72.27%
Epoch subnetwork drawn from
CIFAR-10
CIFAR-100

Models	Methods	Pruning ratio	Top-1 Acc. (%)	Top-1 Acc. Improv. (%)	Top-5 Acc. (%)	Top-5 Acc. Improv. (%)	Total Training FLOPs (P)	Total Training Energy (MJ)
ResNet18 ImageNet	Unpruned	-	69.57	-	89.24	-	1259.13	98.14
	EB Train	10%	**69.84**	**+0.27**	**89.39**	**+0.15**	1177.15	95.71
		30%	68.28	-1.29	88.28	-0.96	**952.46**	**84.65**
ResNet50 ImageNet	Unpruned	-	75.99	-	92.98	-	2839.96	280.72
	EB Train	30%	**73.86**	**-1.73**	**91.52**	**-1.46**	2242.30	232.18
		50%	73.35	-2.24	91.36	-1.62	1718.78	188.18
		70%	70.16	-5.43	89.55	-3.43	**890.65**	**121.15**

Malach et al. *Proving the Lottery Ticket Hypothesis: Pruning is All You Need*

Fix a network $G(x) = W^{G(l)}\sigma\left(W^{G(l-1)}\sigma(\ldots W^{G(1)}x)\right)$, where $\sigma(x) = \max\{x, 0\}$ (ReLU) and $W^{G(i)}$ are the weights in the i-th layer.

$$W^{G(1)} \in \mathbb{R}^{d\times n}, \qquad W^{G(i)} \in \mathbb{R}^{n\times n}, \text{ for every } 1 < i < l, \qquad W^{G(1)} \in \mathbb{R}^{n\times 1}.$$

Malach et al. *Proving the Lottery Ticket Hypothesis: Pruning is All You Need*

Fix a network $G(x) = W^{G(l)}\sigma\left(W^{G(l-1)}\sigma(\ldots W^{G(1)}x)\right)$, where $\sigma(x) = \max\{x, 0\}$ (ReLU) and $W^{G(i)}$ are the weights in the i-th layer.

$$W^{G(1)} \in \mathbb{R}^{d\times n}, \qquad W^{G(i)} \in \mathbb{R}^{n\times n}, \text{ for every } 1 < i < l, \qquad W^{G(1)} \in \mathbb{R}^{n\times 1}.$$

Definition (subnetwork):

A **subnetwork** of G is any network of the form $\tilde{G}(x) = \widetilde{W}^{G(l)}\sigma\left(\widetilde{W}^{G(l-1)}\sigma(\ldots \widetilde{W}^{G(1)}x)\right)$, where $\widetilde{W}^{G(i)} = m_i \odot W^{G(i)}$ for some $m_i \in \{0,1\}^{n_{in}\times n_{out}}$.

(n_{in}, n_{out} denote the input/output dimension of each layer, respectively)

Malach et al. *Proving the Lottery Ticket Hypothesis: Pruning is All You Need*

- Given a target network of depth l with bounded weights
- A random network of depth $2l$ and polynomial width contains with high probability a subnetwork that approximates the target network.

Malach et al. *Proving the Lottery Ticket Hypothesis: Pruning is All You Need*

- Given a target network of depth l with bounded weights
- A random network of depth $2l$ and polynomial width contains with high probability a subnetwork that approximates the target network.

Theorem 2.1. *Fix some $\epsilon, \delta \in (0,1)$. Let F be some target network of depth l such that for every $i \in [l]$ we have $\|W^{F(i)}\|_2 \leq 1, \|W^{F(i)}\|_{\max} \leq \frac{1}{\sqrt{n_{in}}}$ (where $n_{in} = d$ for $i = 1$ and $n_{in} = n$ for $i > 1$). Let G be a network of width $\mathrm{poly}(d, n, l, \frac{1}{\epsilon}, \log \frac{1}{\delta})$ and depth $2l$, where we initialize $W^{G(i)}$ from $U([-1,1])$. Then, w.p at least $1-\delta$ there exists a weight-subnetwork $\widetilde{G}$ of G such that:*

$$\sup_{x \in \mathcal{X}} \left| \widetilde{G}(x) - F(x) \right| \leq \epsilon$$

Discussion and Open Questions

- What does this actually tell us about these highly non-linear systems (deep nets) we are trying to understand?
- What additional theoretical support would help augment the Lottery Ticket Hypothesis?
- What could be the main ingredients that determine whether an initialization is a winning ticket or not?
- What is special about large weights? Do other alternative rewinding strategies preserve winning tickets? And why set weights to zero?
- Any interesting additional experiments to try?
- Can we exploit lottery tickets?

References

- Frankle, J., & Carbin, M. (2019). The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks. arxiv.org/abs/1803.03635.
- Frankle, J., Dziugaite, G., Roy, D., & Carbin, M. (2020). Stabilizing the Lottery Ticket Hypothesis. arxiv.org/abs/1903.01611.
- You, H., et al. (2020). Drawing Early-Bird Tickets: Towards More Efficient Training of Deep Networks. arxiv.org/abs/1909.11957.
- Lange, R. (2020). The Lottery Ticket Hypothesis: A Survey. roberttlange.github.io/year-archive/posts/2020/06/lottery-ticket-hypothesis/.
- Malach, E., Yehudai, G., Shalev-Shwartz, S., & Shamir, O. (2020). Proving the Lottery Ticket Hypothesis: Pruning is All You Need. arxiv.org/abs/2002.00585.