Strong Lottery Ticket Hypothesis with ε-Perturbation

Zheyang Xiong*, Fangshuo Liao*, Anastasios Kyrillidis

April 28, 2023

The Lottery Ticket Hypothesis

A randomly initialized, dense neural network contains a subnetwork that is **initialized** *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)

The Lottery Ticket Hypothesis

- How to find lottery ticket:
  - Train a randomly initialized NN.

The Lottery Ticket Hypothesis

- How to find lottery ticket:
  - Train a randomly initialized NN.
  - Prune weights with small magnitude.

The Lottery Ticket Hypothesis

- How to find lottery ticket:
  - Train a randomly initialized NN.
  - Prune weights with small magnitude.
  - Rewind remaining weights to initialization & Re-train.

The Lottery Ticket Hypothesis

- How to find lottery ticket:
  - Train a randomly initialized NN.
  - Prune weights with small magnitude.
  - Rewind remaining weights to initialization & Re-train.
- If instead of rewinding, we randomly initialize again, the performance is worse.

The Lottery Ticket Hypothesis

- How to find lottery ticket:
  - Train a randomly initialized NN.
  - Prune weights with small magnitude.
  - Rewind remaining weights to initialization & Re-train.
- If instead of rewinding, we randomly initialize again, the performance is worse.
- **Initialization is important**

The Strong Lottery Ticket Hypothesis

- The approximation process is
  - We have a target neural network $\mathcal{F}$

The Strong Lottery Ticket Hypothesis

- The approximation process is
  - We have a target neural network $\mathcal{F}$
  - We have a randomly initialized neural network $\mathcal{G}$ (much larger)

The Strong Lottery Ticket Hypothesis

- The approximation process is
  - We have a target neural network $\mathcal{F}$
  - We have a randomly initialized neural network $\mathcal{G}$ (much larger)
  - We can approximate $\mathcal{F}$ by pruning $\mathcal{G}$

$$\eta = \min_{\mathcal{M}} \sup_{\mathbf{x}} \|\mathcal{F}(x) - (\mathcal{M} \circ \mathcal{G})(x)\|$$

The Strong Lottery Ticket Hypothesis

- The approximation process is
  - We have a target neural network $\mathcal{F}$
  - We have a randomly initialized neural network $\mathcal{G}$ (much larger)
  - We can approximate $\mathcal{F}$ by pruning $\mathcal{G}$

$$\eta = \min_{\mathcal{M}} \sup_{\mathbf{x}} \|\mathcal{F}(x) - (\mathcal{M} \circ \mathcal{G})(x)\|$$

- No need for training!

The Strong Lottery Ticket Hypothesis

- The approximation process is
  - We have a target neural network $\mathcal{F}$
  - We have a randomly initialized neural network $\mathcal{G}$ (much larger)
  - We can approximate $\mathcal{F}$ by pruning $\mathcal{G}$

$$\eta = \min_{\mathcal{M}} \sup_{\mathbf{x}} \|\mathcal{F}(x) - (\mathcal{M} \circ \mathcal{G})(x)\|$$

- No need for training!
- Easier to prove!

The Strong Lottery Ticket Hypothesis

- The approximation process is
  - We have a target neural network $\mathcal{F}$
  - We have a randomly initialized neural network $\mathcal{G}$ (much larger)
  - We can approximate $\mathcal{F}$ by pruning $\mathcal{G}$

$$\eta = \min_{\mathcal{M}} \sup_{\mathbf{x}} \|\mathcal{F}(x) - (\mathcal{M} \circ \mathcal{G})(x)\|$$

- No need for training!
- Easier to prove!
- But the over-parameterization will be larger.

The Goal

We want to understand the LTH using ideas from the Strong LTH.

Strong LTH with ε-Perturbation

- Treating the weight change during training as general perturbation around initialization

Strong LTH with ε-Perturbation

- Treating the weight change during training as general perturbation around initialization
- Example, consider NN $\mathcal{G}_{\boldsymbol{W}}$

$$\mathcal{G}_{\boldsymbol{W}} \xrightarrow{\text{Perturb}} \mathcal{G}_{\boldsymbol{W}+\Delta\boldsymbol{W}}$$

Strong LTH with ε-Perturbation

- Treating the weight change during training as general perturbation around initialization
- Example, consider NN $\mathcal{G}_{\boldsymbol{W}}$

$$\mathcal{G}_{\boldsymbol{W}} \xrightarrow{\text{Perturb}} \mathcal{G}_{\boldsymbol{W}+\Delta\boldsymbol{W}}$$

- Require $\|\Delta \boldsymbol{W}\|_\infty \leq \varepsilon$, and we can study how varying ε affects the approximation error η

$$\eta = \min_{\Delta\boldsymbol{W},\mathcal{M}} \sup_{\mathbf{x}} \|\mathcal{F}(\mathbf{x}) - (\mathcal{M} \circ \mathcal{G}_{\boldsymbol{W}+\Delta\boldsymbol{W}})(\mathbf{x})\|$$

How much Over-parameterization Does Strong LTH Need?

Theorem

Assume $\mathcal{F}$ has L layers, and the width of the ℓth layer is d_ℓ for all $\ell \in [L]$. Then if $\mathcal{G}$ has $2L$ layers, and the width of the $(2\ell - 1)$th layer is d'_ℓ, the width of the 2ℓth layer is d_ℓ. As long as

$$d'_\ell = O\left(d_{\ell-1} \log\left(\hat{\eta}^{-1} d_\ell d_{\ell-1} L\right)\right)$$

then with high probability, we have

$$\min_{\mathcal{M}} \sup_{\mathbf{x}} \left\|\mathcal{F} - \left(\mathcal{M} \circ \mathcal{G}\right)(x)\right\| \leq \hat{\eta}$$

How much Over-parameterization Does Strong LTH Need?

For short:

- if $\mathcal{F}$ has n parameters in total and L layers
- then we need $\mathcal{G}$ to have $\left(n \log\left(\hat{\eta}^{-1} nL\right)\right)$ parameters and $2L$ layers

Guaranteed Approximation Using Subset-Sum

The Subset-Sum Problem:

Guaranteed Approximation Using Subset-Sum

The Subset-Sum Problem:

- Given n randomly generated candidate values $\{x_i\}_{i=1}^{n}$ and a target value z

Guaranteed Approximation Using Subset-Sum

The Subset-Sum Problem:

- Given n randomly generated candidate values $\{x_i\}_{i=1}^n$ and a target value z
- Find the best mask $\boldsymbol{\delta} \in \{0,1\}^n$ to minimize $|\sum_{i=1}^n \delta_i x_i - z|$.

Guaranteed Approximation Using Subset-Sum

The Subset-Sum Problem:

- Given n randomly generated candidate values $\{x_i\}_{i=1}^n$ and a target value z
- Find the best mask $\boldsymbol{\delta} \in \{0, 1\}^n$ to minimize $|\sum_{i=1}^n \delta_i x_i - z|$.

Theoretical Guarantee (Lueker, 1996)

Guaranteed Approximation Using Subset-Sum

The Subset-Sum Problem:

- Given n randomly generated candidate values $\{x_i\}_{i=1}^n$ and a target value z
- Find the best mask $\boldsymbol{\delta} \in \{0,1\}^n$ to minimize $|\sum_{i=1}^n \delta_i x_i - z|$.

Theoretical Guarantee (Lueker, 1996)

- Let $\eta = \min_\delta |\sum_{i=1}^n \delta_i x_i - z|$

Guaranteed Approximation Using Subset-Sum

The Subset-Sum Problem:

- Given n randomly generated candidate values $\{x_i\}_{i=1}^n$ and a target value z
- Find the best mask $\boldsymbol{\delta} \in \{0,1\}^n$ to minimize $|\sum_{i=1}^n \delta_i x_i - z|$.

Theoretical Guarantee (Lueker, 1996)

- Let $\eta = \min_\delta |\sum_{i=1}^n \delta_i x_i - z|$
- If $n = O\left(\log \eta^{-1}\right)$, then w.h.p over $\{x_i\}_{i=1}^n$, all z has an η-approximation

ε-Perturbed Subset-Sum

The ε-Perturbed Subset-Sum Problem:

ε-Perturbed Subset-Sum

The ε-Perturbed Subset-Sum Problem:

- Still have n random candidates $\{x_i\}_{i=1}^n$ and a target z

ε-Perturbed Subset-Sum

The ε-Perturbed Subset-Sum Problem:

- Still have n random candidates $\{x_i\}_{i=1}^n$ and a target z
- Find the best $\mathbf{y} \in [-\epsilon, \epsilon]^n$ and $\boldsymbol{\delta} \in \{0,1\}^n$ to minimize $|\sum_{i=1}^n \delta_i(x_i + y_i) - z|$

ε-Perturbed Subset-Sum

The ε-Perturbed Subset-Sum Problem:

- Still have n random candidates $\{x_i\}_{i=1}^n$ and a target z
- Find the best $\mathbf{y} \in [-\epsilon, \epsilon]^n$ and $\boldsymbol{\delta} \in \{0,1\}^n$ to minimize $|\sum_{i=1}^n \delta_i(x_i + y_i) - z|$
- Additional freedom of choosing $\mathbf{y}$

ε-Perturbed Subset-Sum

The ε-Perturbed Subset-Sum Problem:

- Still have n random candidates $\{x_i\}_{i=1}^n$ and a target z
- Find the best $\mathbf{y} \in [-\epsilon, \epsilon]^n$ and $\boldsymbol{\delta} \in \{0,1\}^n$ to minimize $|\sum_{i=1}^n \delta_i(x_i + y_i) - z|$
- Additional freedom of choosing $\mathbf{y}$

Theoretical Guarantee

ε-Perturbed Subset-Sum

The ε-Perturbed Subset-Sum Problem:

- Still have n random candidates $\{x_i\}_{i=1}^n$ and a target z
- Find the best $\mathbf{y} \in [-\epsilon, \epsilon]^n$ and $\boldsymbol{\delta} \in \{0,1\}^n$ to minimize $|\sum_{i=1}^n \delta_i(x_i + y_i) - z|$
- Additional freedom of choosing $\mathbf{y}$

Theoretical Guarantee

- Let $\eta = \min_\delta |\sum_{i=1}^n \delta_i(x_i + y_i) - z|$

ε-Perturbed Subset-Sum

The ε-Perturbed Subset-Sum Problem:

- Still have n random candidates $\{x_i\}_{i=1}^n$ and a target z
- Find the best $\mathbf{y} \in [-\epsilon, \epsilon]^n$ and $\boldsymbol{\delta} \in \{0,1\}^n$ to minimize $|\sum_{i=1}^n \delta_i(x_i + y_i) - z|$
- Additional freedom of choosing $\mathbf{y}$

Theoretical Guarantee

- Let $\eta = \min_\delta |\sum_{i=1}^n \delta_i(x_i + y_i) - z|$
- If $n = O\left(\frac{\log \eta^{-1}}{\log(1+\epsilon)+1}\right)$, then w.h.p over $\{x_i\}_{i=1}^n$, all z has an η-approximation

Approach of Approximation

width $= 2n$

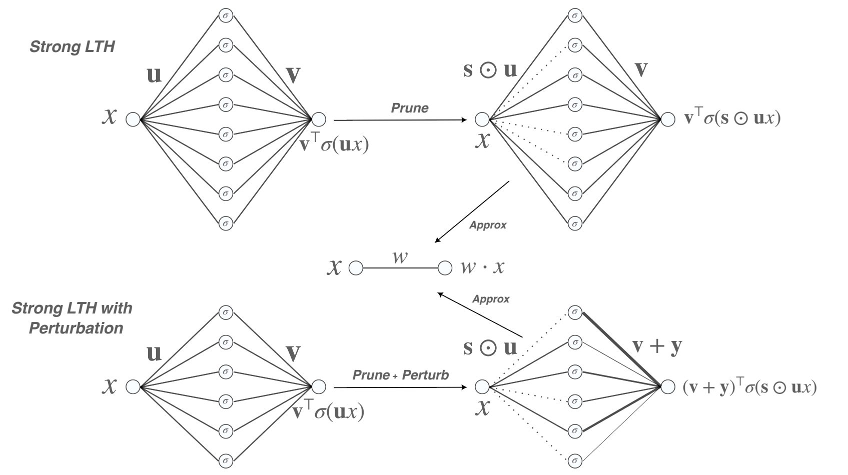
$u_i = 1$ if $i \leq n$;

$u_i = -1$ if $i > n$

$\mathbf{v} \sim \mathtt{Unif}[-1, 1]^{2n}$

$\mathbf{s} \in \{0, 1\}^{2n}$

$\sigma(\cdot) = \max\{0, \cdot\}$



ε-Perturbed Strong LTH

Theorem

Assume $\mathcal{F}$ has L layers, and the width of the ℓth layer is d_ℓ for all $\ell \in [L]$. Then if $\mathcal{G}$ has $2L$ layers, and the width of the $(2\ell-1)$th layer is d'_ℓ, the width of the 2ℓth layer is d_ℓ. As long as

$$d'_\ell = O\left(d_{\ell-1}\frac{\log\left(\hat{\eta}^{-1}d_\ell d_{\ell-1}L\right)}{\log\left(1+\epsilon\right)+1}\right)$$

then with high probability, we have

$$\min_{\mathcal{M},\Delta\mathbf{W}}\sup_{\mathbf{x}}\left\|\mathcal{F}-\left(\mathcal{M}\circ\mathcal{G}_{\mathbf{W}+\Delta\mathbf{W}}\right)(x)\right\| \le \hat{\eta}$$

ε-Perturbed Strong LTH

For short:

- if $\mathcal{F}$ has n parameters in total and L layers
- then we need $\mathcal{G}$ to have $\left(n\frac{\log\left(\hat{\eta}^{-1}nL\right)}{\log(1+\epsilon)+1}\right)$ parameters and $2L$ layers

Question

How to find a good ε perturbation?

GD finds Good ε-Perturbation

- Run projected GD under $\|\Delta \boldsymbol{W}\|_{\max} \leq \varepsilon$ $\longrightarrow$
- Finding best pruning with `Edge-Popup`. $\longrightarrow$
- Finding best (sparsity, accuracy) pair. $\longrightarrow$

Algorithm 1 PGD+StrongLTH

Input: Perturbation scale ε, neural network loss $\mathcal{L}$, initial weight $\mathbf{W}_0$, learning rate $\{\alpha_t\}_{t=0}^{T-1}$

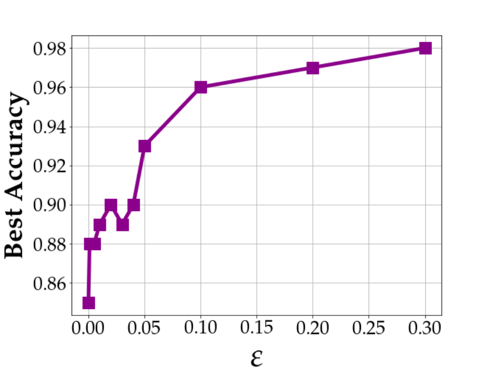
$\Delta\mathbf{W} \leftarrow 0$
for $t \in \{0, \ldots, T-1\}$ **do**
  $\hat{\mathbf{W}} \leftarrow \Delta\mathbf{W} - \alpha_t \nabla\mathcal{L}(\mathbf{W}_t)$
  $\Delta\mathbf{W} \leftarrow \mathtt{sign}(\hat{\mathbf{W}}) \cdot \min\{\mathtt{abs}(\hat{\mathbf{W}}), \varepsilon\}$
  $\mathbf{W}_{t+1} \leftarrow \mathbf{W}_0 + \Delta\mathbf{W}$
end for
$\ell^* \leftarrow \infty\,, \mathcal{M}^* \leftarrow \text{None}$
for pruning level $s \in \{0.1, 0.2, \ldots, 0.9\}$ **do**
  $\ell, \mathcal{M} \leftarrow \mathtt{Edge\text{-}Popup}(\mathcal{L}, \mathbf{W}_T, s)$
  if $\ell \leq \ell^*$ **then**
    $\ell^* \leftarrow \ell\,, \mathcal{M}^* \leftarrow \mathcal{M}$
  end if
end for
return Optimal loss ℓ^*, mask $\mathbf{M}^*$ and sparsity level s

Results

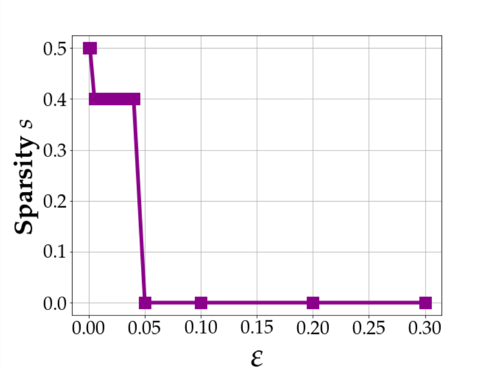
Sparsity s	Perturbation Scale ε										
	0	10^{-3}	$5 \cdot 10^{-3}$	10^{-2}	$2 \cdot 10^{-2}$	$3 \cdot 10^{-2}$	$4 \cdot 10^{-2}$	$5 \cdot 10^{-2}$	10^{-1}	$2 \cdot 10^{-1}$	$4 \cdot 10^{-1}$
0	0.12	0.14	0.25	0.42	0.68	0.84	**0.90**	**0.93**	**0.96**	**0.97**	**0.98**
0.1	0.49	0.48	0.65	0.70	0.78	0.82	0.87	0.87	0.94	0.97	0.98
0.2	0.75	0.76	0.77	0.79	0.84	0.86	0.88	0.87	0.93	0.96	0.97
0.3	0.83	0.82	0.82	0.82	0.88	0.88	0.86	0.90	0.92	0.94	0.93
0.4	0.82	0.86	**0.88**	**0.89**	**0.90**	**0.89**	0.90	0.90	0.88	0.91	0.86
0.5	**0.85**	**0.88**	0.86	0.89	0.87	0.88	0.89	0.89	0.90	0.89	0.76
0.6	0.83	0.87	0.87	0.83	0.86	0.88	0.87	0.88	0.87	0.85	0.54
0.7	0.81	0.85	0.84	0.83	0.86	0.82	0.81	0.81	0.79	0.74	0.29
0.8	0.73	0.71	0.71	0.75	0.77	0.75	0.73	0.68	0.77	0.55	0.17

Red: Strong LTH; Blue: SGD without Pruning; Orange: SGD dominates pruning.

Results



Best Accuracy
0.98
0.96
0.94
0.92
0.90
0.88
0.86
0.00 0.05 0.10 0.15 0.20 0.25 0.30
ε



Sparsity s
0.5
0.4
0.3
0.2
0.1
0.0
0.00 0.05 0.10 0.15 0.20 0.25 0.30
ε

Next Steps: Does GD Approximates Single Vector $\mathbf{w}$?

Given a set of input data points $\{\mathbf{x}_i\}_{i=1}^m$, whether solving the optimization problem of $\min_{\mathbf{U}} \sum_{i=1}^m \left\|\mathbf{1}^\top \mathbf{U}\mathbf{x}_i - \mathbf{w}^\top \mathbf{x}_i\right\|_2^2$ using gradient descent

$$\mathbf{U}_{t+1} = \mathbf{U}_t - \alpha \frac{\partial}{\partial \mathbf{U}} \sum_{i=1}^m \left\|\mathbf{1}^\top \mathbf{U}_t \mathbf{x}_i - \mathbf{w}^\top \mathbf{x}_i\right\|_2^2$$

will satisfy the descending property

$$\left\|\mathbf{w} - \left(\mathbf{U}_{t+1} \odot \mathcal{M}_{t+1}\right)^\top \mathbf{1}\right\|_2 < \left\|\mathbf{w} - \left(\mathbf{U}_t \odot \mathcal{M}_t\right)^\top \mathbf{1}\right\|_2,$$

where $\mathcal{M}_t$ is the optimal mask in iteration t: $\mathcal{M}_t = \operatorname{argmin}_{\mathcal{M}} \left\|\mathbf{w} - \left(\mathbf{U}_t \odot \mathcal{M}\right)^\top \mathbf{1}\right\|_2$