JULY 2023

TEXAS
The University of Texas at Austin

Instant Soup: Cheap Pruning Ensembles in A Single Pass Can Draw Lottery Tickets from Large Models

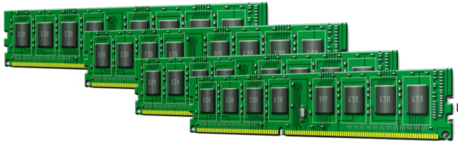
Ajay Jaiswal · Shiwei Liu · Tianlong Chen · Ying Ding · Zhangyang "Atlas" Wang

Ajay Jaiswal

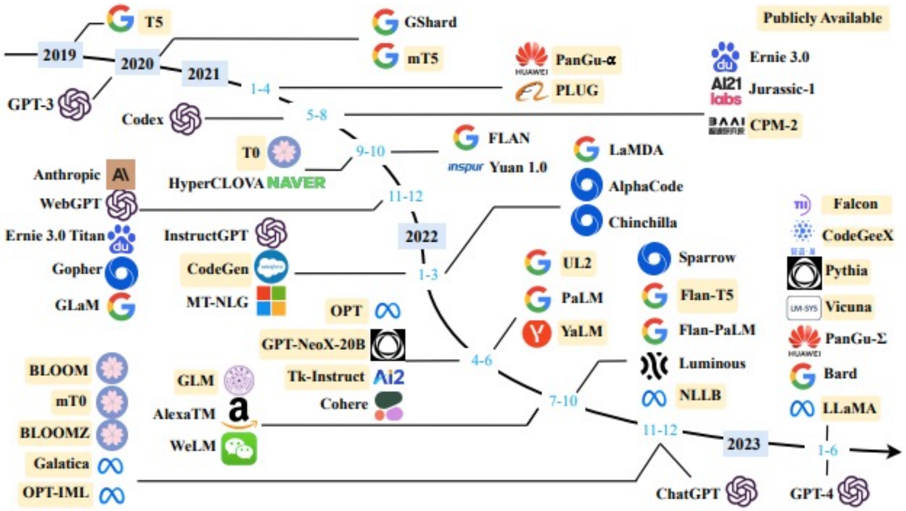
VITA Group, The University of Texas at Austin

https://vita-group.github.io

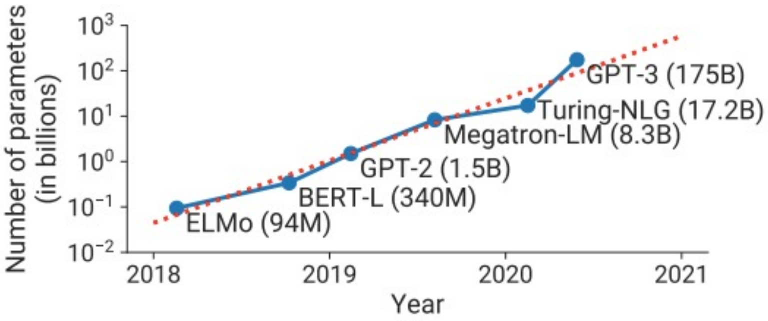
Today's AI is HUGE(R)



7



Source: https://arxiv.org/pdf/2303.18223.pdf



Source: https://arxiv.org/pdf/2104.04473.pdf

Sparse Neural Networks (SNNs) have received voluminous attention predominantly due to growing computational and memory in large-scale models….

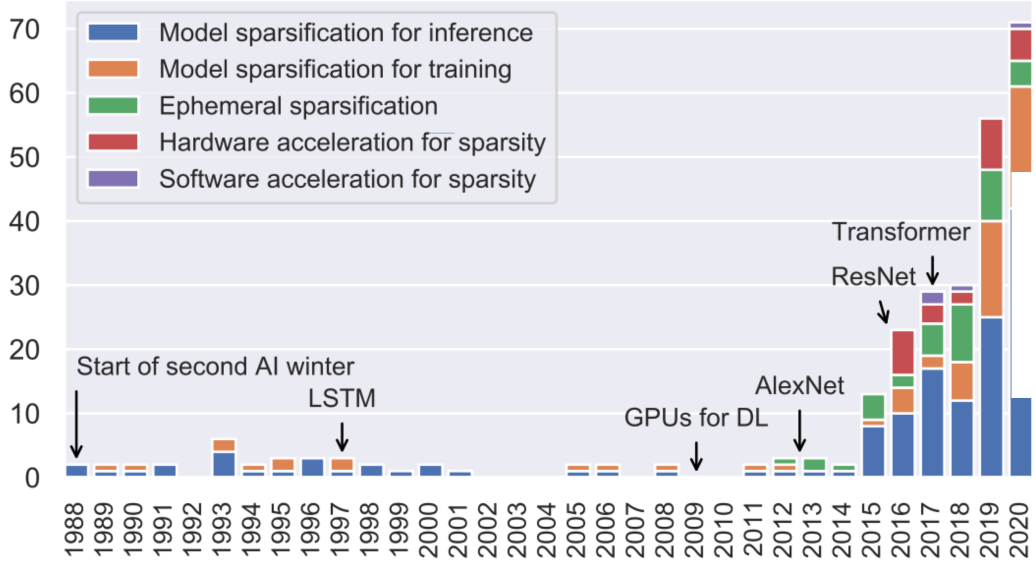


Fig. 1. Literature development of sparsity over the years.

THE LOTTERY TICKET HYPOTHESIS: FINDING SPARSE, TRAINABLE NEURAL NETWORKS

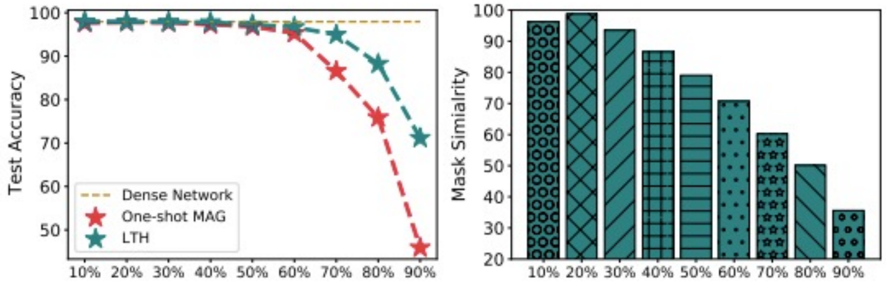
Jonathan Frankle
MIT CSAIL
jfrankle@csail.mit.edu

Michael Carbin
MIT CSAIL
mcarbin@csail.mit.edu

Hoefler, T., Alistarh, D., Ben-Nun, T., Dryden, N., & Peste, A. (2021). Sparsity in Deep Learning: Pruning and growth for efficient inference and training in neural networks. J. Mach. Learn. Res., 22(241), 1-124.

Question: as our models are getting huge(r) …

Does there exist a principled and cheaper approach for fastly drawing high-quality lottery tickets in large pretrained models within a limited computational budget, while preserving its performance and transferability?



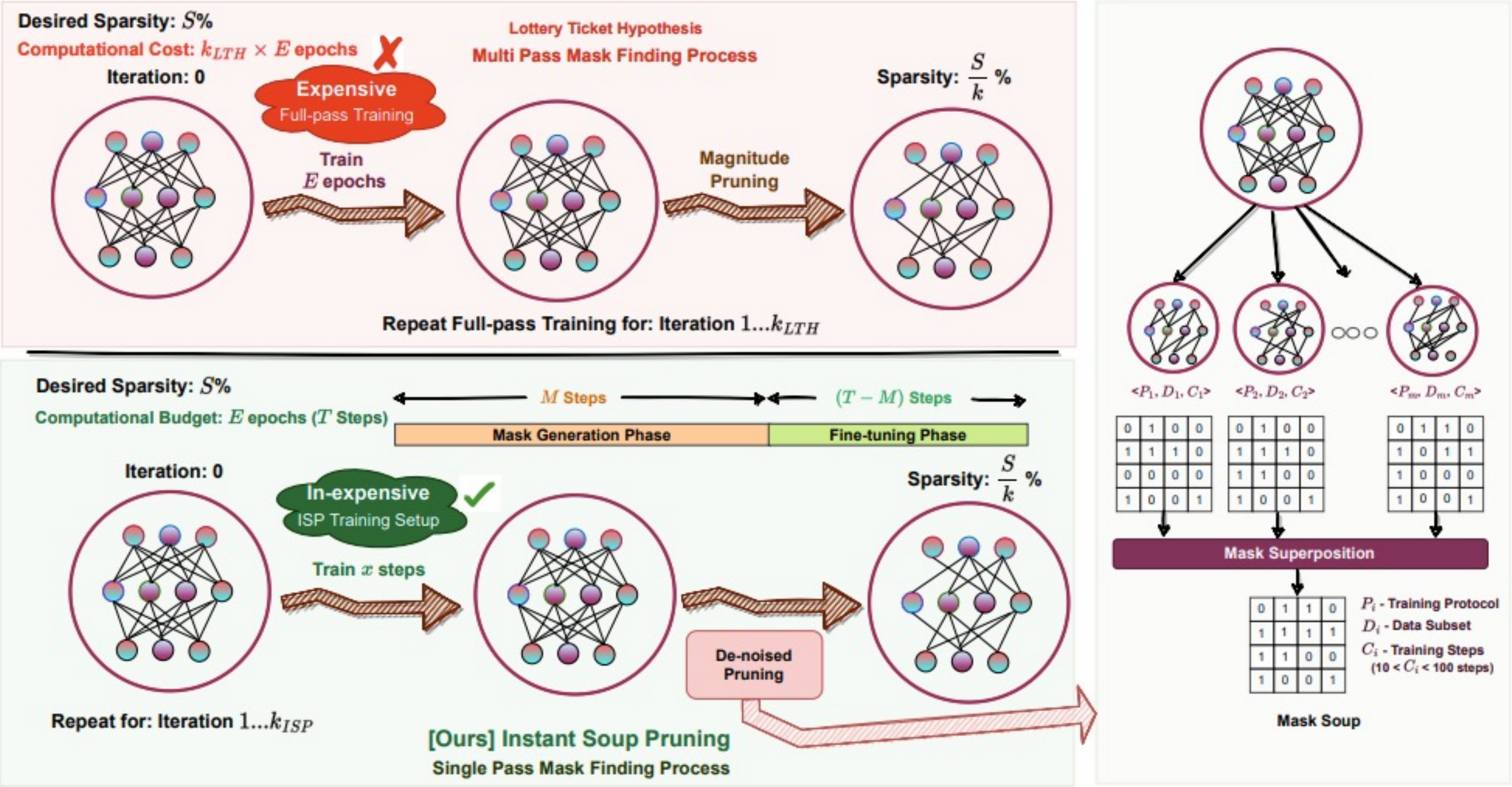

Desired Sparsity: S%
Computational Cost: $k_{LTH} \times E$ epochs
Lottery Ticket Hypothesis
Multi Pass Mask Finding Process
Iteration: 0
Expensive
Full-pass Training
Train
E epochs
Magnitude
Pruning
Sparsity: $\frac{S}{k}$ %
Repeat Full-pass Training for: Iteration $1...k_{LTH}$
Desired Sparsity: S%
Computational Budget: E epochs (T Steps)
M Steps
$(T-M)$ Steps
Mask Generation Phase
Fine-tuning Phase
Iteration: 0
In-expensive
ISP Training Setup
Train x steps
Sparsity: $\frac{S}{k}$ %
De-noised
Pruning
Repeat for: Iteration $1...k_{ISP}$
[Ours] Instant Soup Pruning
Single Pass Mask Finding Process
<P_1, D_1, C_1>
<P_2, D_2, C_2>
<P_m, D_m, C_m>
Mask Superposition
P_i - Training Protocol
D_i - Data Subset
C_i - Training Steps
($10 < C_i < 100$ steps)
Mask Soup

Table 2. Details of fine-tuning CLIP (`ViT-B32`) at varying sparsity levels using Instant Soup Pruning following the settings listed in Table 1. Learning rate decays linearly from the initial value to zero. The evaluation metrics follow standards in (Radford et al., 2021). Entries with errors are the average across three runs, and errors are the standard deviations. LTH results are obtained using IMP.

Pruning Method	Cars			MNIST			SVHN			GTSRB			CIFAR10			CIFAR100		
	30%	40%	50%	70%	80%	90%	70%	80%	90%	50%	60%	70%	60%	70%	80%	60%	70%	80%
Full CLIP$_{\texttt{ViT-B32}}$	76.43 ± 0.5			99.61 ± 0.07			97.40 ± 0.11			99.08 ± 0.24			97.6 ± 0.24			89.35 ± 0.19		
Random	9.11	5.58	4.47	98.71	97.42	87.04	89.85	85.61	73.74	93.76	93.65	90.97	74.51	69.84	64.96	45.20	39.92	43.24
One-shot [Mag]	71.95	68.07	56.79	99.27	98.87	97.47	95.02	91.96	85.76	98.60	97.84	96.14	95.31	86.56	75.85	80.39	63.18	46.63
Progressive [Mag]	69.85	68.62	64.43	99.52	97.77	95.19	95.78	90.53	85.75	98.97	97.57	96.34	95.25	90.87	78.11	81.79	70.53	60.93
EarlyBird (You et al., 2019)	72.53	70.76	65.90	99.38	98.96	97.64	96.34	95.93	87.02	98.15	98.19	97.26	96.06	94.18	86.84	84.22	76.79	65.67
SNIP (Lee et al., 2018)	71.51	68.79	59.01	99.25	98.72	97.50	95.33	91.94	82.98	98.62	97.95	96.22	95.01	87.45	76.12	81.10	62.89	55.89
GraSP (Wang et al., 2020)	71.42	68.55	58.12	99.30	98.51	97.15	95.09	91.44	84.72	98.37	97.42	95.91	95.20	86.89	75.88	80.67	66.31	52.30
LTH (Frankle & Carbin, 2018)	73.97	72.02	66.12	99.41	99.38	98.22	96.69	95.28	87.41	98.71	98.35	97.79	96.42	94.91	87.47	84.25	78.60	65.38
LTH - Rewind	74.28	72.09	66.07	99.62	**99.64**	98.18	96.72	95.22	87.47	98.78	98.36	97.87	96.53	94.88	87.28	84.46	78.62	65.71
Lottery Pool (Yin et al., 2022)	73.10	70.53	64.67	99.25	98.97	97.76	96.54	95.12	87.29	98.52	98.30	97.55	96.14	94.50	87.11	84.07	78.21	64.39
ISP [Ours]	**75.13**	**72.20**	**66.32**	**99.69**	99.61	**98.82**	**96.93**	**96.46**	**87.59**	**99.06**	**99.01**	**98.52**	**96.82**	**95.18**	**91.20**	**85.11**	**79.57**	**71.09**
(std.)	±0.34	±0.27	±0.82	±0.07	±0.15	±0.21	±0.08	±0.05	±0.11	±0.15	±0.29	±0.32	±0.15	±0.20	±0.14	±0.19	±0.22	±0.17

Table 3. Details of fine-tuning BERT (BASE) at varying sparsity levels using Instant Soup Pruning following the settings listed in Table 1. Learning rate decays linearly from the initial value to zero. The evaluation metrics follow standards in (Wolf et al., 2019). Entries with errors are the average across three runs, and errors are the standard deviations. LTH results are obtained using IMP.

Dataset	MNLI	QQP	STS-B	WNLI	QNLI	MPRC	RTS	SST-2	CoLA
Sparsity	70%	90%	50%	90%	70%	50%	60%	60%	50%
Full $\text{BERT}_{\text{BASE}}$	82.4 ± 0.5	90.2 ± 0.5	88.4 ± 0.3	54.9 ± 1.2	89.1 ± 1.0	85.2 ± 0.1	66.2 ± 3.6	92.1 ± 0.1	54.5 ± 0.4
Random	67.5	76.3	21.0	53.5	61.9	69.6	56.0	83.1	9.6
One-shot	78.8	86.2	83.9	53.1	86.2	83.7	62.9	86.5	49.7
Progressive	79.1	87.5	85.0	53.3	87.2	83.8	65.4	86.6	52.2
EarlyBird	82.5	89.4	88.1	54.0	88.5	84.6	66.1	91.2	53.5
Lottery Ticket	82.6	90.0	88.2	54.9	88.9	84.9	65.0	91.9	53.8
Lottery Pool	80.4	89.1	86.4	50.9	87.6	84.5	62.7	90.9	52.6
ISP	$\mathbf{82.71 \pm 0.6}$	$\mathbf{90.59 \pm 0.5}$	$\mathbf{88.64 \pm 0.1}$	$\mathbf{55.33 \pm 0.3}$	$\mathbf{90.06 \pm 1.0}$	$\mathbf{85.38 \pm 0.1}$	$\mathbf{65.96 \pm 0.3}$	$\mathbf{92.43 \pm 0.6}$	53.61 ± 0.2

Q&A

Reach me at: ajayjaiswal@utexas.edu

VITA