EarlyBERT: **Efficient** BERT **Training** via Early-bird Lottery Tickets

Xiaohan Chen[1], Yu Cheng[2], Shuohang Wang[2], Zhe Gan[2], Zhangyang Wang[1], Jingjing Liu[2]

[1] The University of Texas at Austin

[2] Microsoft Corporation

Introduction

- Large-scale pre-trained language models achieve impressive empirical success at a price -- **computational inefficiency** due to
  - More complexed operations such as self-attention
  - Extreme overparameterization
  - E.g., $\text{BERT}_{\text{Large}}$ has over 340M parameters and T5 over 10B

- Such complexity results in many drawbacks
  - Long inference time due to computational inefficiency
  - Data-hungry and resource-demanding pre-training is necessary
  - Most compression works focus on reducing inference time and resources

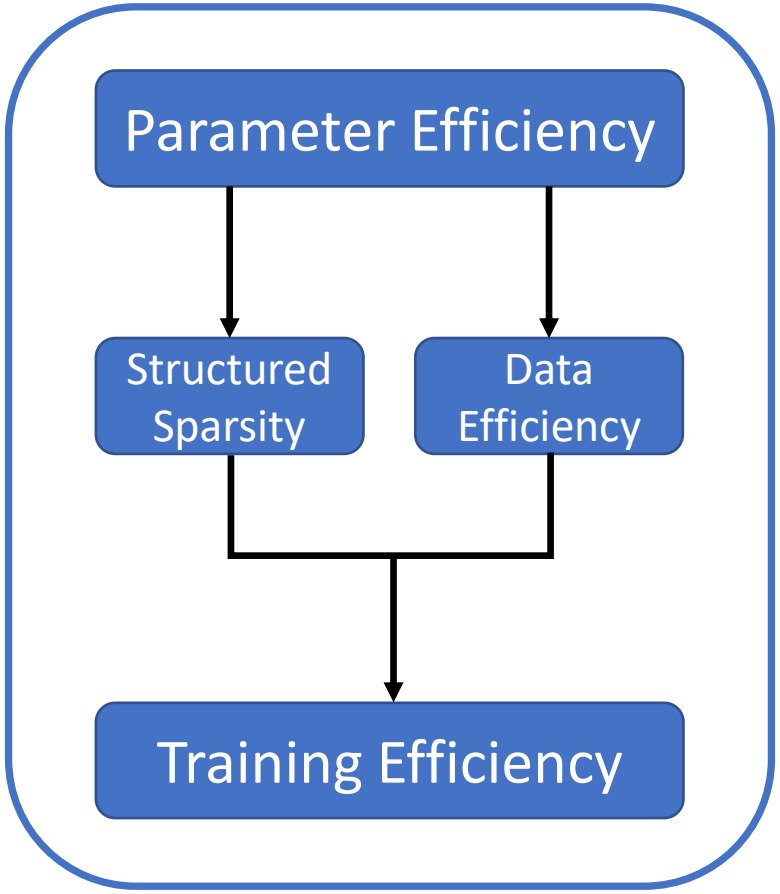
(Early-bird) Lottery Tickets

- Lottery Ticket Hypothesis (LTH) [1] suggests the existence of highly trainable sparse networks at random initialization – **winning tickets**
- However, LTH has two drawbacks
  - The method for finding winning tickets (IMP) is computationally expensive
  - Only unstructured sparsity is achievable – hard for acceleration
- Early-bird Lottery Tickets [2]
  - Structured sparsity
  - Emerge early during training
  - But with (acceptable) performance drops

[1] “The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks”, Jonathan Frankle & Michael Carbin, ICLR 2019.
[2] “Drawing Early-Bird Tickets: Toward More Efficient Training of Deep Networks”, Haoran You et al., ICLR 2020.

EarlyBERT – Early-bird Lottery Tickets in BERT

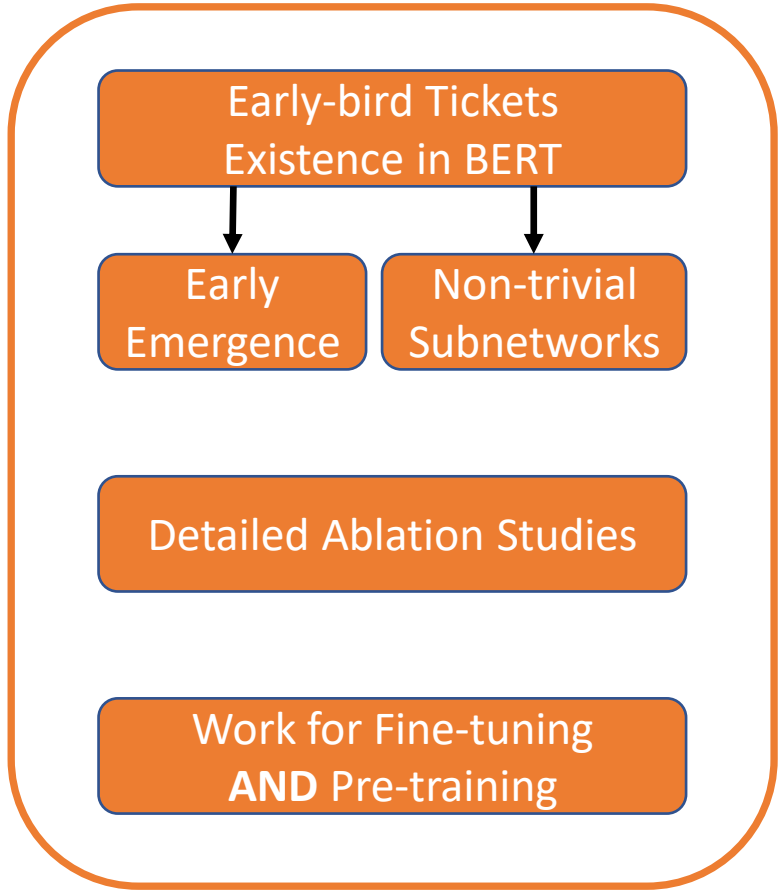
Efficiency Level



EarlyBERT

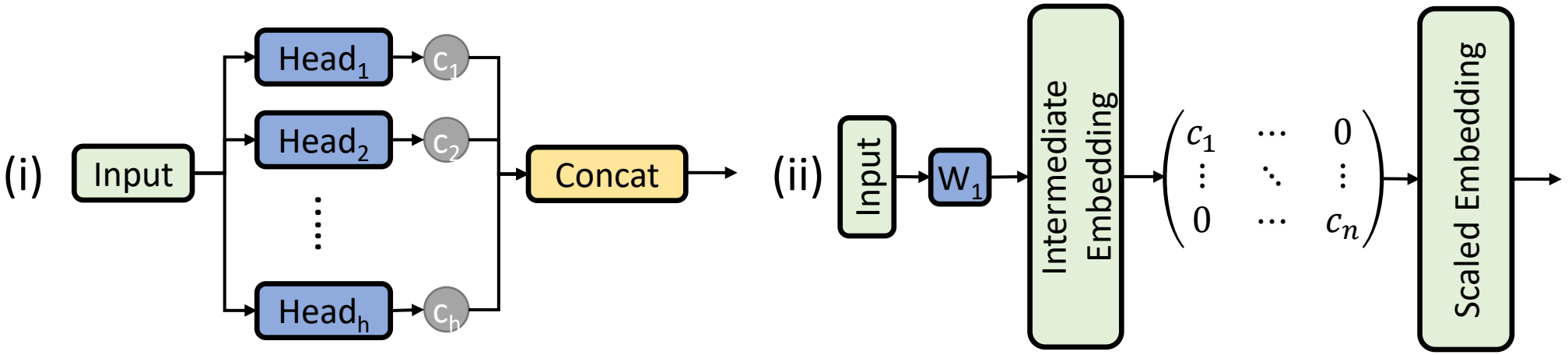
A general algorithmic framework for **efficient training** of BERT models

Algorithm Level



How to Search EarlyBERT Tickets?

- We follow the main idea of Network Slimming (NS) [3]
  - Batch normalization is not used in most NLP models
  - We manually add learnable coefficients to
    i. Multi-head self-attention modules
    ii. Intermediate neurons in the feed-forward networks (FFN)



$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{h}_1, \ldots, \text{h}_n)W^O$$
$$\text{h}_i = c_i^h \cdot \text{Attention}(QW_i^Q, KW_i^K, VW_i^V).$$

$$\text{FFN}(x) = c^f \cdot \max(0, xW_1 + b_1)W_2 + b_2. \quad (3)$$

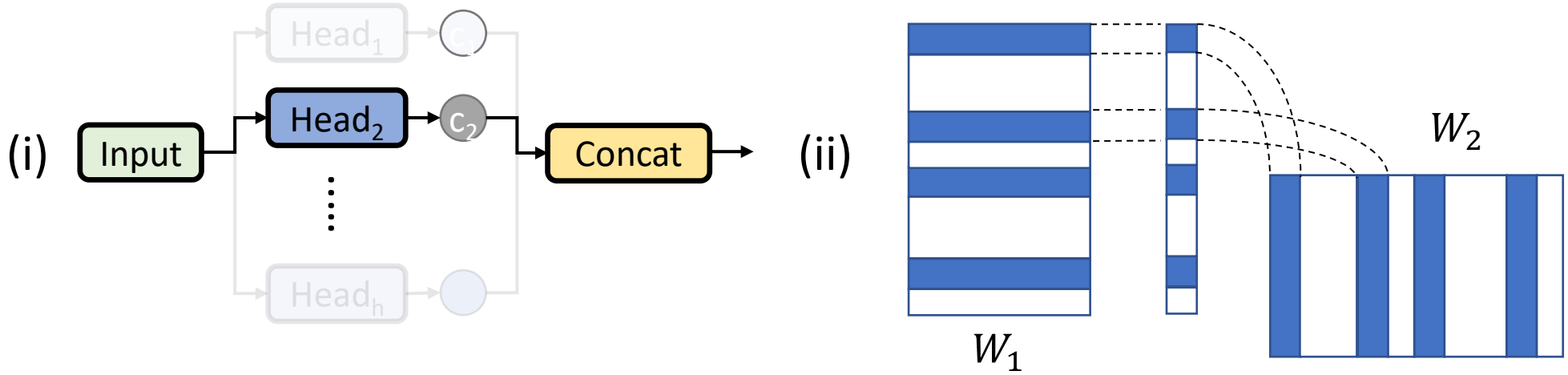
[3] "Learning efficient convolutional networks through network slimming", Zhuang Liu et al., ICCV 2017.

How to Draw EarlyBERT Tickets?

- We train the coefficients along with the model parameters, with l_1 regularization loss to promote sparsity

$$\mathcal{L}(f(\cdot;\theta),c) = \mathcal{L}_0(f(\cdot;\theta),c) + \lambda\|c\|_1,$$

- When the coefficients are sufficiently trained, we prune self-attention heads and the intermediate neurons whose coefficients have smallest magnitudes



Prune self-attention heads layer-wise

Prune intermediate neurons globally

The Overall EarlyBERT framework

I. Searching Stage

II. Draw EarlyBERT Tickets

III. Efficiently train the EarlyBERT tickets. Pre-train **OR** Fine-tune

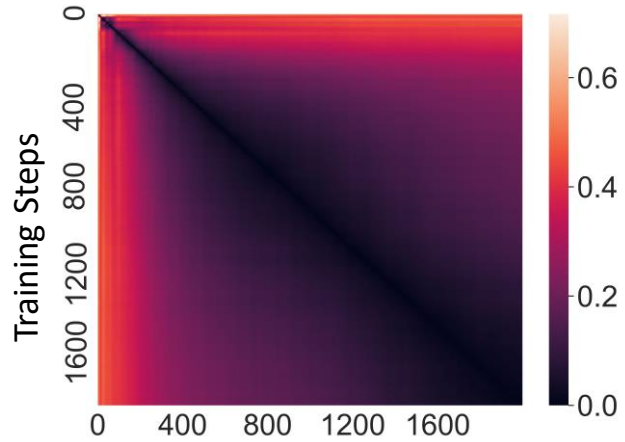
Dense training. Need to **sufficiently** train the coefficients.

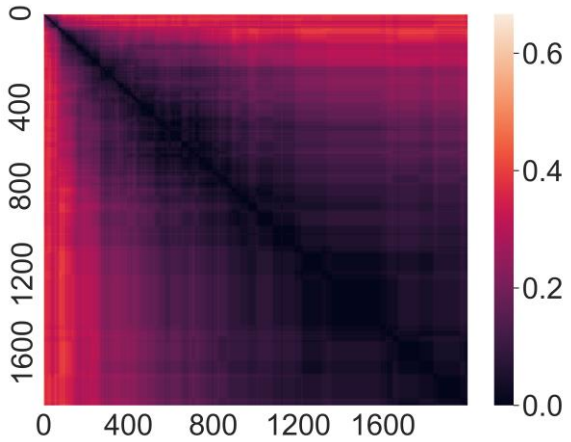
Take the sub-structure that is essential for the performance. No extra cost.

Enjoy (i) computation efficiency due to **structured** sparsity; and (2) data efficiency due to reduced parameter complexity.

EarlyBERT Tickets Emerge Early

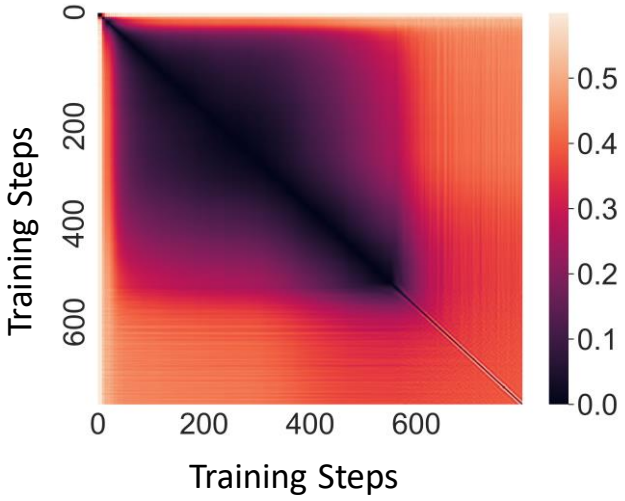
(a) Intermediate neuron during fine-tuning

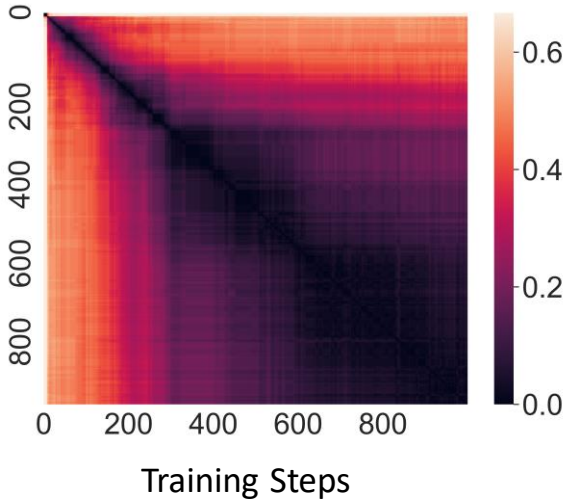




(b) Self-attention heads during fine-tuning

(c) Intermediate neuron during pre-training





(d) Self-attention heads during pre-training

EarlyBERT Finds Non-Trivial Subnetworks

- Compare
  a) $BERT_{Base}$
  b) $EarlyBERT_{Base}$
  c) Random pruning

- Only self-attention heads are pruned here

Methods	MNLI	QNLI	QQP	SST-2
$BERT_{BASE}$	83.16	90.59	90.34	91.70
$EarlyBERT_{BASE}$	83.58	90.33	90.41	92.09
Random	82.26	88.87	90.12	91.17
Methods	CoLA	RTE	MRPC	
$BERT_{BASE}$	0.535	65.70	80.96	
$EarlyBERT_{BASE}$	0.527	66.19	81.54	
Random	0.514	63.86	78.57	

Empirical Results – Fine-tuning

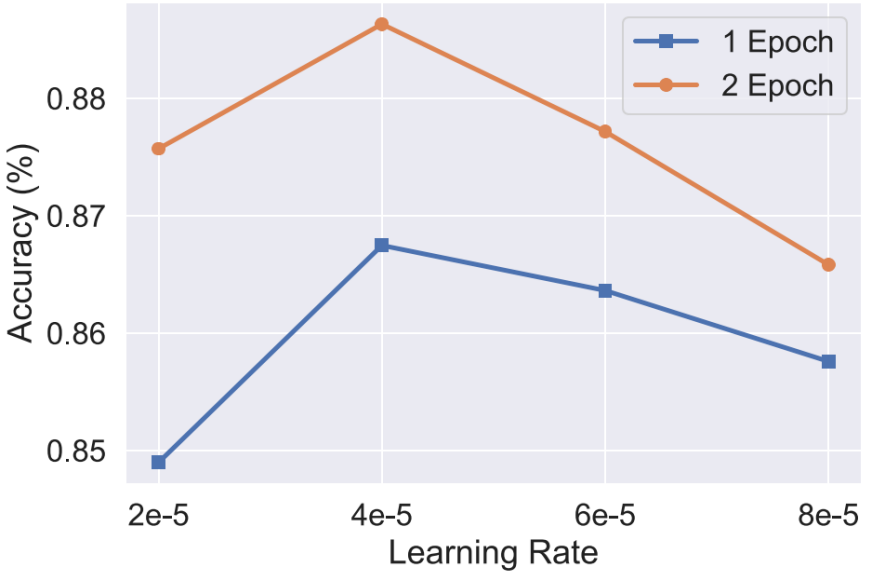
- We empirically evaluate EarlyBERT on GLUE and SQuAD tasks
  - We prune 4 self-attention heads in each layer
  - We prune 40% intermediate neurons globally

Methods	MNLI	QNLI	QQP	SST-2	SQuAD	Time Saved[2]
$\text{BERT}_{\text{BASE}}$	83.16	90.59	90.34	91.70	87.50	-
$\text{EarlyBERT}_{\text{BASE}}$	81.81	89.18	90.06	90.71	86.13	40∼45%
$\text{Random}_{\text{BASE}}$	79.92	84.46	89.42	89.68	84.47	45∼50%
LayerDrop (Fan et al., 2019)	81.27	88.91	88.06	89.89	84.25	∼33%
$\text{BERT}_{\text{LARGE}}$	86.59	92.29	91.59	92.21	90.76	-
$\text{EarlyBERT}_{\text{LARGE}}$	85.13	89.22	90.64	90.94	89.45	35∼40%
$\text{Random}_{\text{LARGE}}$	78.45	84.46	89.89	88.65	88.79	40∼45%
LayerDrop (Fan et al., 2019)	85.12	91.12	88.88	89.97	89.44	∼33%

Ablation Studies

Time Saving / Prune Ratio	3 Heads	4 Heads	5 Heads	6 Heads
FC - 30%	-35.78%	-38.66%	-41.26%	-45.34%
	89.62%	89.55%	89.60%	89.50%
FC - 40%	-39.72%	-42.97%	-43.93%	-44.49%
	89.66%	89.61%	89.58%	89.38%
FC - 50%	-43.89%	-45.54%	-47.02%	-48.53%
	89.54%	89.35%	89.34%	89.31%

(a) Performance-Efficiency Trade-off



(b) Data Efficiency of EarlyBERT

λ	10^{-4}	10^{-3}	10^{-2}
	88.55	88.43	88.42
# Pruned Heads	4	5	6
Layer-wise pruning	**88.55**	88.13	87.65
# Pruned Neurons	30%	40%	50%
Layer-wise pruning	88.18	**88.22**	87.90
Global pruning	**88.31**	88.23	87.91

(c) Regularization and pruning method

Empirical Results – Pre-training

- We perform 400 steps of training during the searching stage
- During the ticket-drawing stage
  - We prune 4 self-attention heads in each layer
  - We prune 30% intermediate neurons globally
- The EarlyBERT ticket is then pre-trained with reduced number of steps

Methods	CoLA	MNLI	MRPC	QNLI	QQP	RTE	SST-2	SQuAD
$\text{BERT}_{\text{BASE}}$	0.45	81.40	84.07	89.86	89.80	60.29	90.48	87.60
$\text{EarlyBERT}_{\text{BASE}}$	0.41	79.97	80.39	89.86	89.44	61.01	90.94	85.48
$\text{BERT}_{\text{LARGE}}$	0.50	83.56	85.90	90.44	90.45	59.93	92.55	90.43
$\text{EarlyBERT}_{\text{LARGE}}$	0.47	82.54	85.54	90.46	90.38	61.73	91.51	89.36

Thank you for your attention!

Welcome to contact me for further questions or discussion via xiaohan.chen@utexas.edu

Our paper

Our GitHub