SmartPipe: Intelligently Freezing Layers in Pipeline Parallelism for Distributed DNN Training

Nadia Niknami, Abdalaziz Sawwan and Jie Wu
Dept. of Computer and Information Sciences Temple University, USA

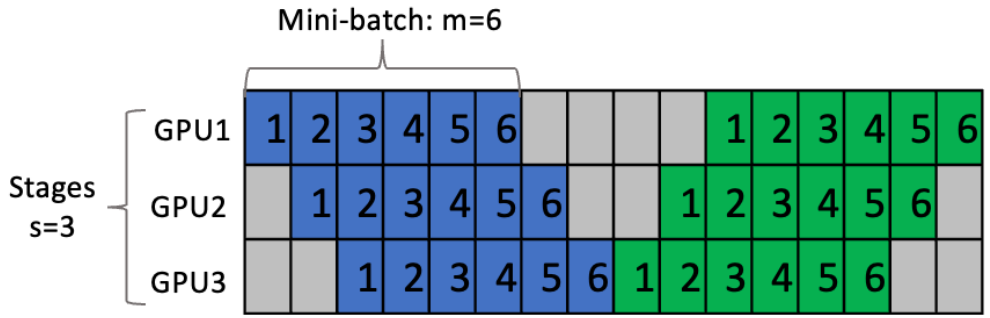
ICPADS 2023

Outline

1. Introduction
2. Motivation for Extended Pipeline Parallelism
3. Experiment
4. Conclusions

1. Introduction

- Distributed DNN Training
  - Data Parallelism
    - Partition data and assign to multiple workers
    - Each worker node has parameters of the whole model
  - Model Parallelism
    - Partition models
  - **Pipeline Parallelism**
    - Data + Model parallelism



2. Motivation

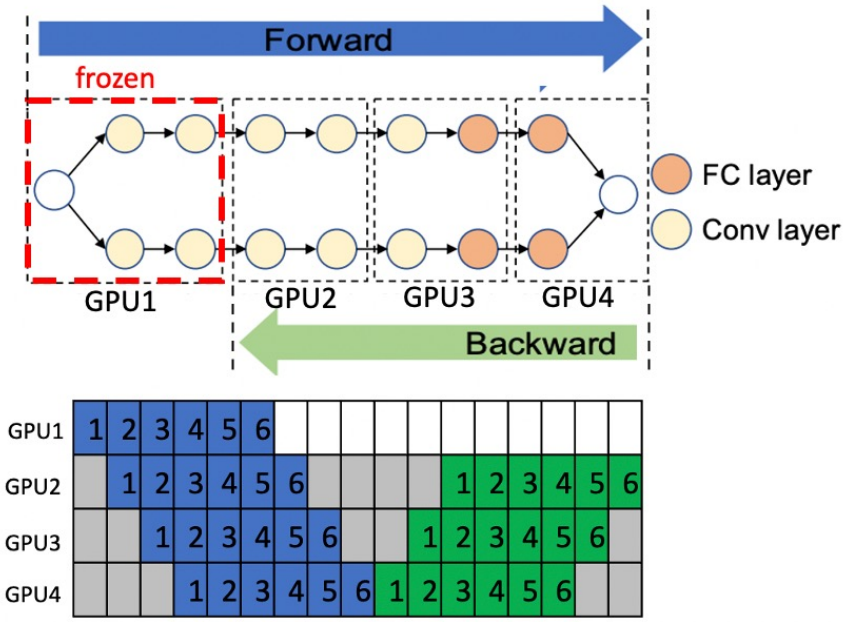
- Training Models
  - Deep learning models require larger training times as the depth of a model increases
  - Challenges and prolonged training durations

- Transfer learning
  - By incorporating pre-trained models into a new model, training time is significantly reduced.
  - Generalization error is lowered, enhancing the model's ability to perform well on new, unseen data

2. Motivation

- Freezing Layers in Transfer Learning
  - The front layers become well-trained much earlier
  - The deeper layers are more task-specific and capture complicated features outputted by front layers.
- Insights
  - Not all layers need to undergo training for the entire training duration.
  - We can reduce computation by consecutively freezing layers
- Consideration
  - Ensure optimal accuracy.
  - The freezing process should be applied at an appropriate stage

2. Motivation



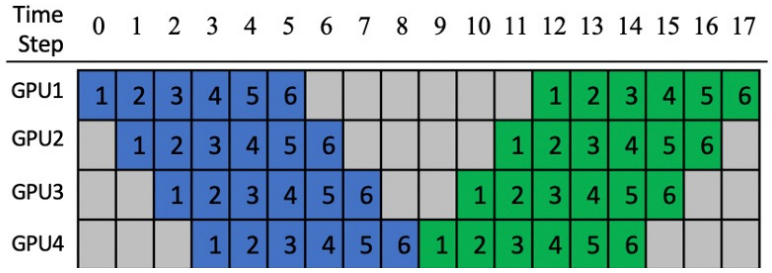
- We can reduce computation and prevent overfitting by consecutively freezing layers

Example 1 -Gpipe pipeline

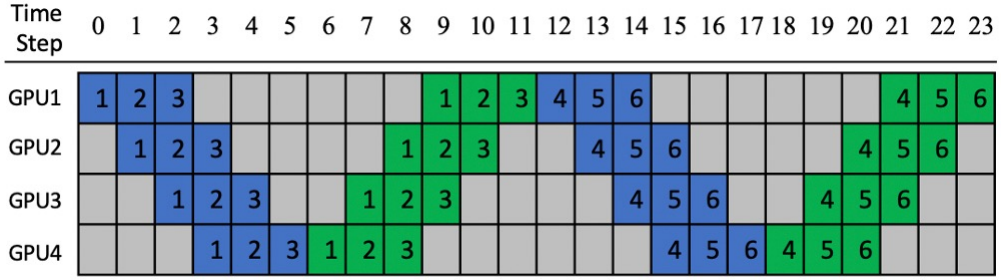
- Consider a dataset of size D = 6 and a total of s = 4 stages.

In diagram, the blue and green colors denote forward and backward propagation operations, respectively. The bubbles are depicted in gray areas.

batch size is 6



batch size is 3

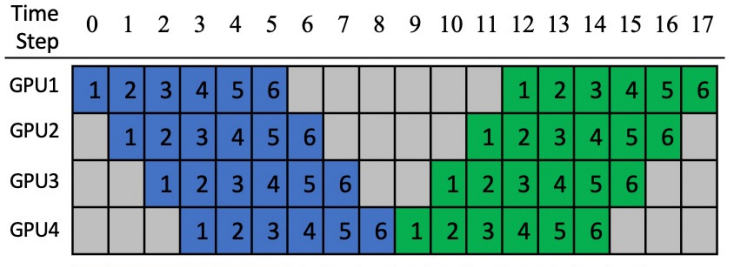


Example 1 (Cont.)

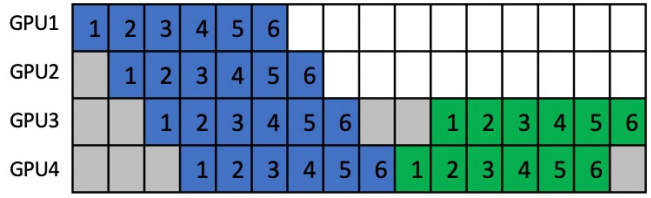
- *No Freezing* method is employed with a batch size of b = 6:
  - The number of bubbles is 24,
  - The training time amounts to 18 units.

- The proposed method (SmartPipe) in employed with b = 6 and by freezing two layers:
  - The number of bubbles is reduced to 9,
  - The training time is reduced to 16 units.

No Freezing



SmartPipe



batch size is 6

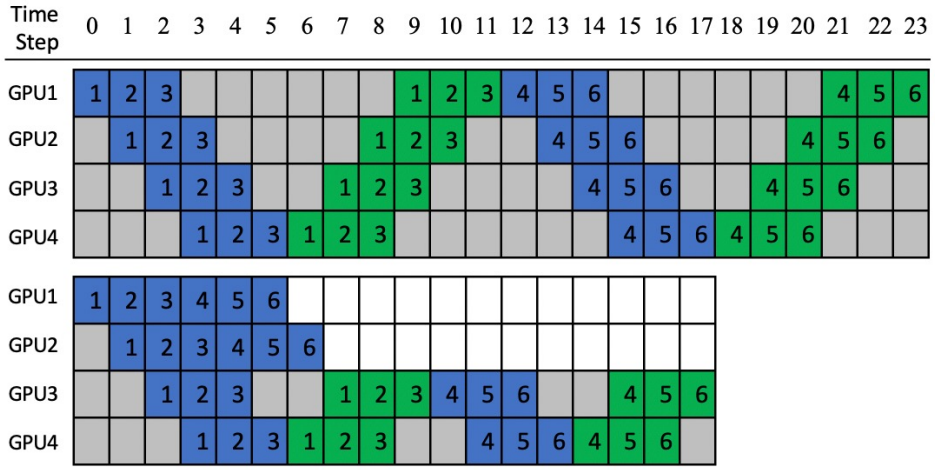
Example 1 -Gpipe pipeline (Cont.)

- *No Freezing* method is employed with a batch size of b = 3:
  - The number of bubbles is 48,
  - The training time amounts to 24 units.

- The proposed method (SmartPipe) in employed with a batch size of b = 3 and by freezing two layers:
  - The number of bubbles is reduced to 13,
  - The training time is reduced to 18 units.

No Freezing

SmartPipe



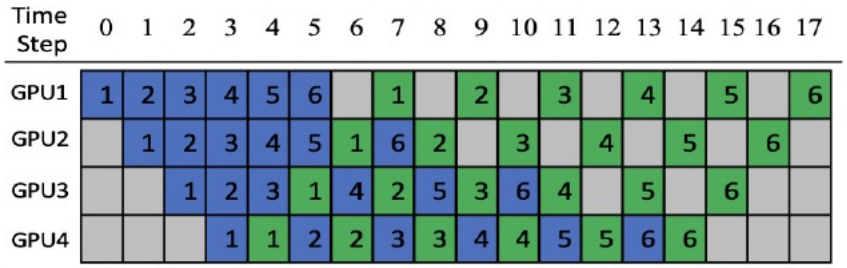
batch size is 3

Example 2 -1F1B pipeline

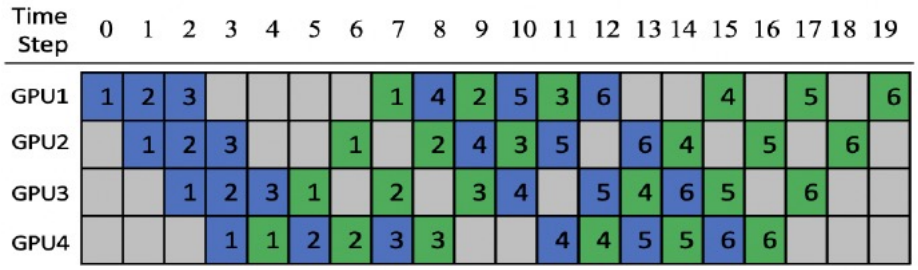
- Consider a dataset of size D = 6 and a total of s = 4 stages.

In diagram, the blue and green colors denote forward and backward propagation operations, respectively. The bubbles are depicted in gray areas.

batch size is 6



batch size is 3



Example 2 -1F1B pipeline (Cont.)

No Freezing method is employed with a batch size of b = 6:

The number of bubbles is 24,

The training time amounts to 18 units.

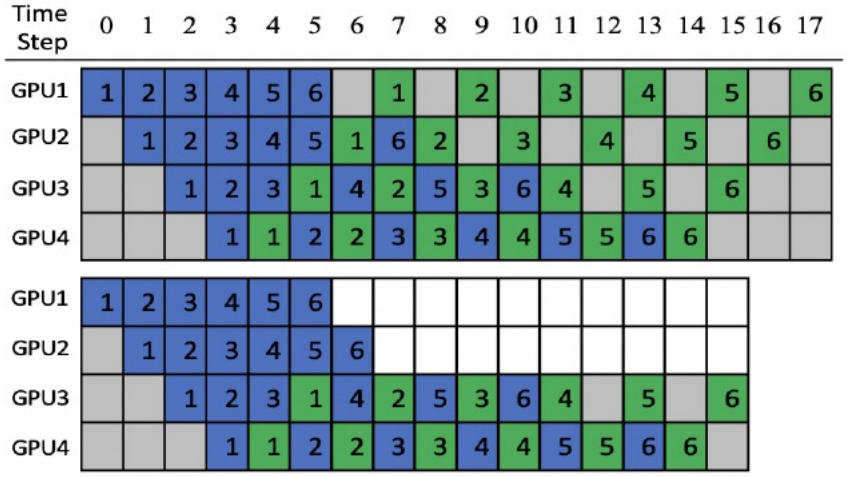
The proposed method (SmartPipe) in employed with a batch size of b = 6 and by freezing two layers:

The number of bubbles is reduced to 9,

The training time is reduced to 16 units.

No Freezing

SmartPipe



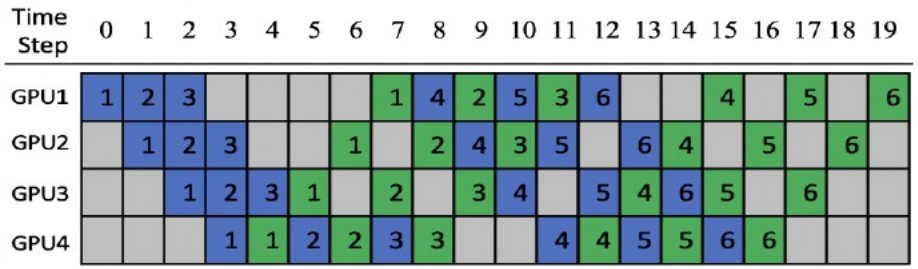
batch size is 6

Example 2 -1F1B pipeline (Cont.)

- *No Freezing* method is employed with a batch size of b = 3:
  - The number of bubbles is 32,
  - The training time amounts to 20 units.

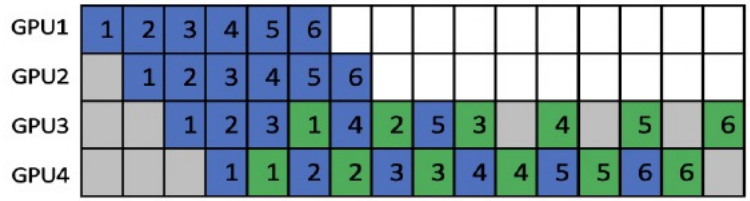
- The proposed method (SmartPipe) in employed with b = 3 and by freezing two layers:
  - The number of bubbles is reduced to 10,
  - The training time is reduced to 16 units.

No Freezing



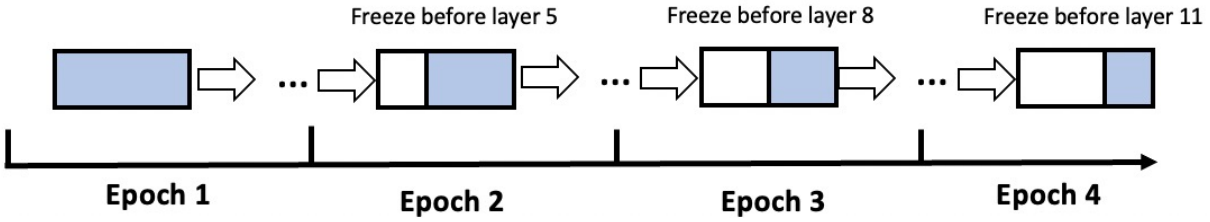
batch size is 3

SmartPipe



Progressive Layer Freezing

- The first training iteration involves all layers being updated.
- Subsequent iterations progressively freeze layers, starting from early layers and progressing towards the latest layers in an orderly fashion



Learning the Importance of Layers

Definition 1. (Freezing Decision) For a layer l, whose weights at timestamp j can be denoted as W_l^j: Given a sequence of its weight history $(W_l^j)_{j=0}^t$ at timestamp t, yield a positive decision to freeze the layer at current epoch if the layer is ready to be frozen, and yield a negative decision if the layer needs further training.

Definition 2. (Gradient Norm Difference) We define the alteration in gradient norm for layer l at time t as η:

$$\eta_l = \frac{\|g_l^{t-1} - g_l^t\|}{\|g_l^{t-1}\|}.$$

Frozen Layers Sequence

Definition 3. (Frozen Layer Sequence) A layer can be marked for freezing if all preceding layers are frozen, and it holds the layer that experiences the slow rate of change. Therefore, to implement the freezing algorithm for layer l_i, two conditions should be considered:

1) Gradient norm difference of l_i should be smaller than given *Threshold* ($\eta_{l_i} < \tau$)
2) For $k = 0$ to $k = i - 1$: gradient of l_k should be zero ($g_{l_k} = 0$).

Definition 4. (Freezing Rate) We propose an adaptive freeze algorithm to define $L^{(t)}_{frozen}$ as the set of frozen layers at time step t where $t \geq 1$ as follows:

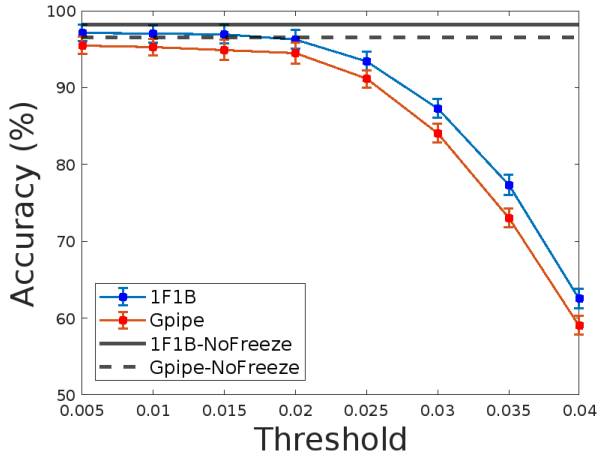
$$|L^{(t)}_{frozen}| = |L^{(t-1)}_{frozen}| + \lambda_t |(L - L^{(t-1)}_{frozen})|,$$

3. Experiment Results

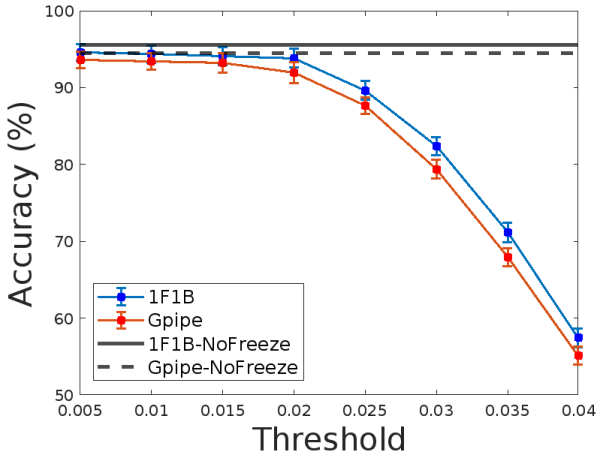
Comparison of accuracy and training time on different datasets

Model	Method	CIFAR-100		ImageNet	
		Accuracy	Time(s)	Accuracy	Time(s)
RestNet50	Full training	$96.10\% \pm 0.12\%$	$2,594$	$81.68\% \pm 0.15\%$	$2,500$
	Fixed Freezing (k=2)	$96.05\% \pm 0.25\%$	$1,980$	$81.48\% \pm 0.29\%$	$1,844$
	Linear Freezing (k=#epoch)	$95.03\% \pm 0.21\%$	$2,424$	$78.30\% \pm 0.48\%$	$1,963$
	SmartPipe (ours)	$96.02\% \pm 0.11\%$	$1,955$	$81.63\% \pm 0.22\%$	$1,787$
AlexNet	Full training	$97.48\% \pm 0.20\%$	$14,603$	$85.03\% \pm 0.28\%$	$14,628$
	Fixed Freezing (k=2)	$81.06\% \pm 0.23\%$	$8,760$	$73.89\% \pm 0.19\%$	$9,956$
	Linear Freezing (k=#epoch)	$90.56\% \pm 0.22\%$	$10,368$	$76.10\% \pm 0.32\%$	$10,654$
	SmartPipe (ours)	$97.35\% \pm 0.16\%$	$8,662$	$84.82\% \pm 0.25\%$	$9,599$
GoogLeNet	Full Training	$93.36\% \pm 0.17\%$	$2,698$	$74.95\% \pm 0.11\%$	$2,703$
	Fixed Freezing (k=2)	$92.35\% \pm 0.19\%$	$1,587$	$73.26\% \pm 0.23\%$	$1,846$
	Linear Freezing (k=#epoch)	$92.08\% \pm 0.19\%$	$10,291$	$72.26\% \pm 0.23\%$	$1,846$
	SmartPipe (ours)	$93.28\% \pm 0.25\%$	$1,554$	$74.66\% \pm 0.29\%$	$1,831$

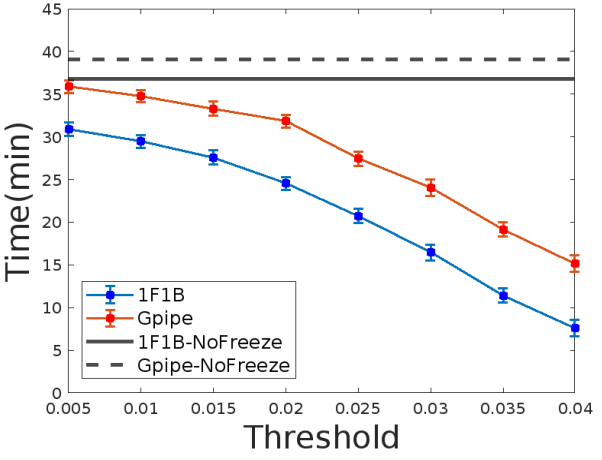
3. Experiment Results



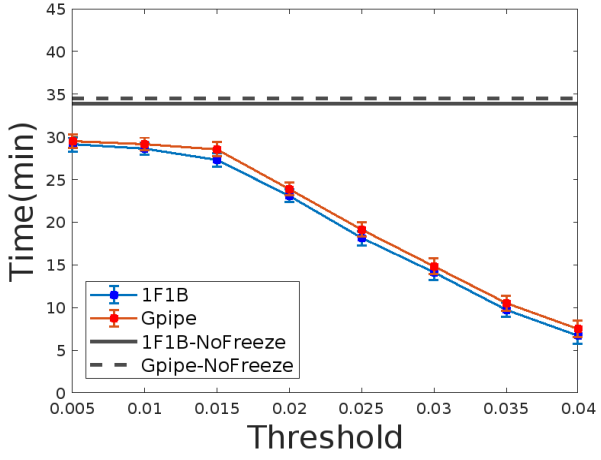
Batch size: 1024



Batch size: 8192

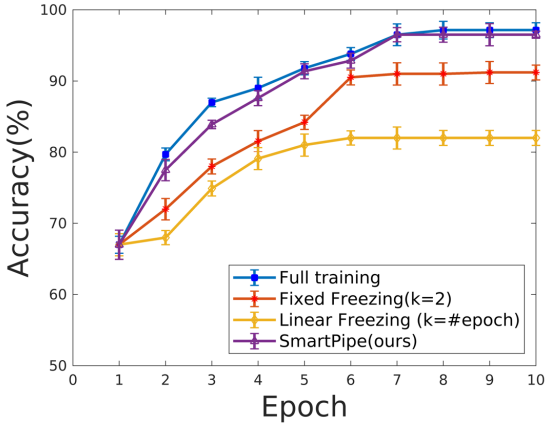


Batch size: 1024

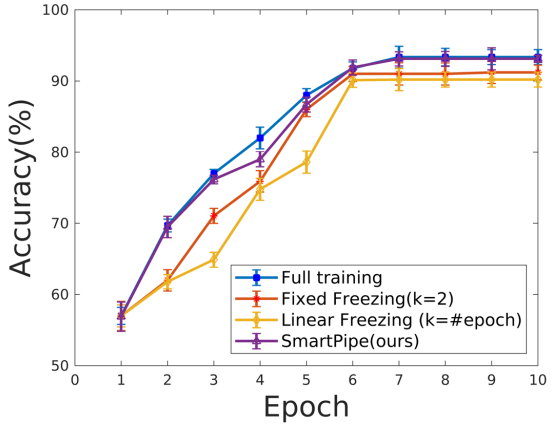


Batch size: 8192

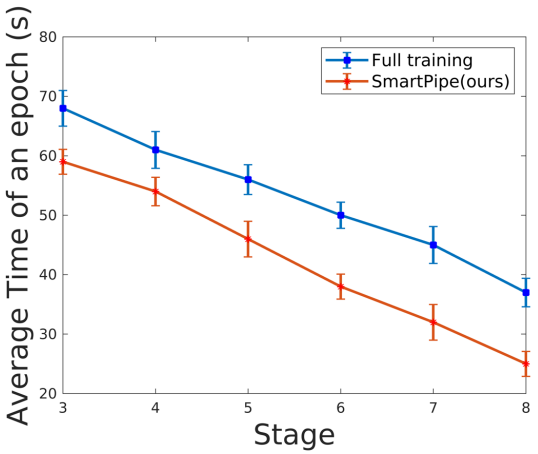
3. Experiment Results



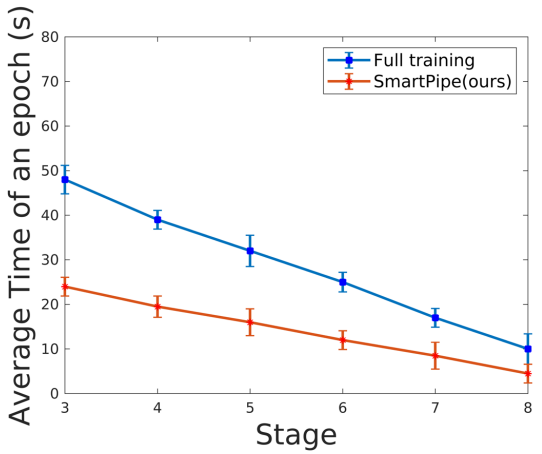
AlexNet



GoogLeNet



AlexNet



GoogLeNet

4. Conclusion

- *Layer freezing involves excluding specific layers from backpropagation, thus retaining their unchanged weights and reducing the computation time during backpropagation.*

- *By dynamically selecting layers to freeze during the training process, we aim to optimize the training efficiency further.*

- *The results of our approach clearly demonstrate its efficiency in reducing the time required for a single training iteration without significantly compromising accuracy*

Thank you!
Q & A

tun03933@temple.edu