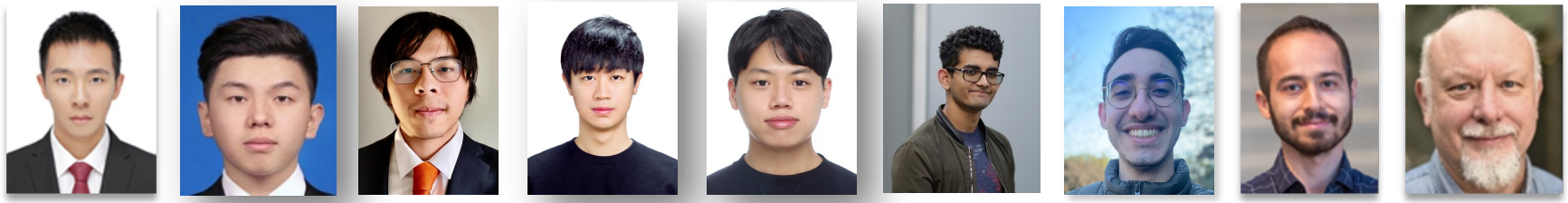
Efficient Inference and Training of Large Neural Network Models

Zhen Dong, Sheng Shen, Yang Zhou, Sehoon Kim, Woosuk Kwon, Aniruddha Nrusimha, Lutfi Eren Ergodan, Amir Gholami, **Kurt Keutzer**

University of California at Berkeley



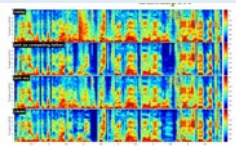
Diverse Application Areas with ML/DL Since 2008

Image Classification

Object Detection

Image Segmentation

Computer Vision



Audio Enhancement

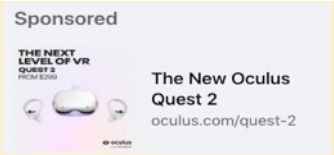
Call-center Sentiment Analysis

Speech Recognition

Audio Analysis And ASR

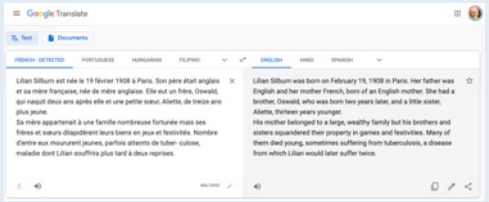
Sentiment Analysis

Music Recommendation



Ad Recommendation

Multimedia and Rec Systems



Translation

Question answering

Document Understanding

Natural Language Processing

Diverse Application Areas are Converging on one/few DNN Models

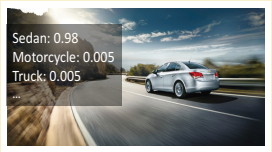
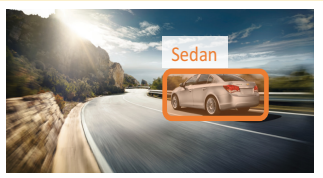
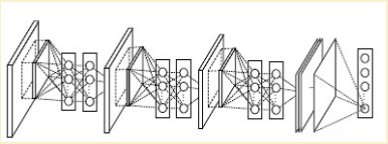


Image Classification

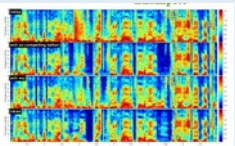


Object Detection

Image Segmentation



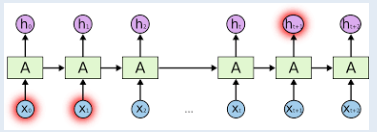
Convolutional NN



Audio Enhancement

Call-center Sentiment Analysis

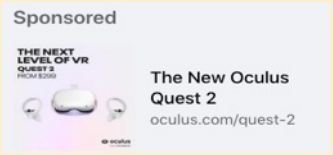
Speech Recognition



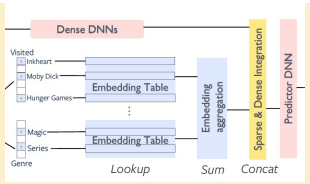
Recurrent NN

Sentiment Analysis

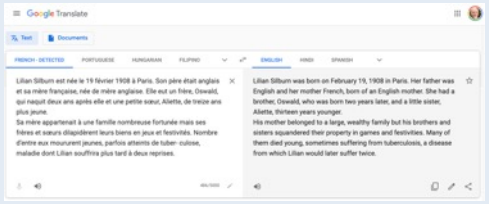
Music Recommendation



Ad Recommendation



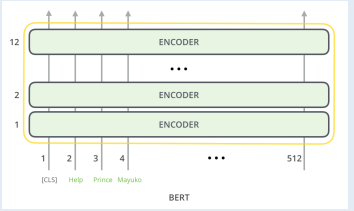
DLRM



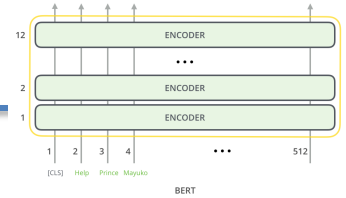
Translation

Question answering

Document Understanding



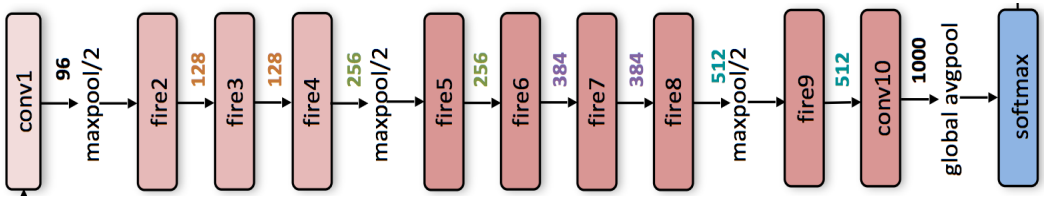
Transformer



Transformer

Our Research Group's Focus
Efficient Deep Learning/Efficient DNNs

Efficient Inference at the Edge



- **Computer Vision**
  - SqueezeNet, SqueezeNext
  - Shift
  - SqueezeDet
  - SqueezeSeg
- **ASR and NLU**
  - SqueezeWave, SqueezeBERT
  - SqueezeFormer

Squeeze Family of DNNs

Efficient/Scalable Training and Inference in the Cloud

Efficient Training
- FireCaffe, LARS, LAMB
- Staged-Training

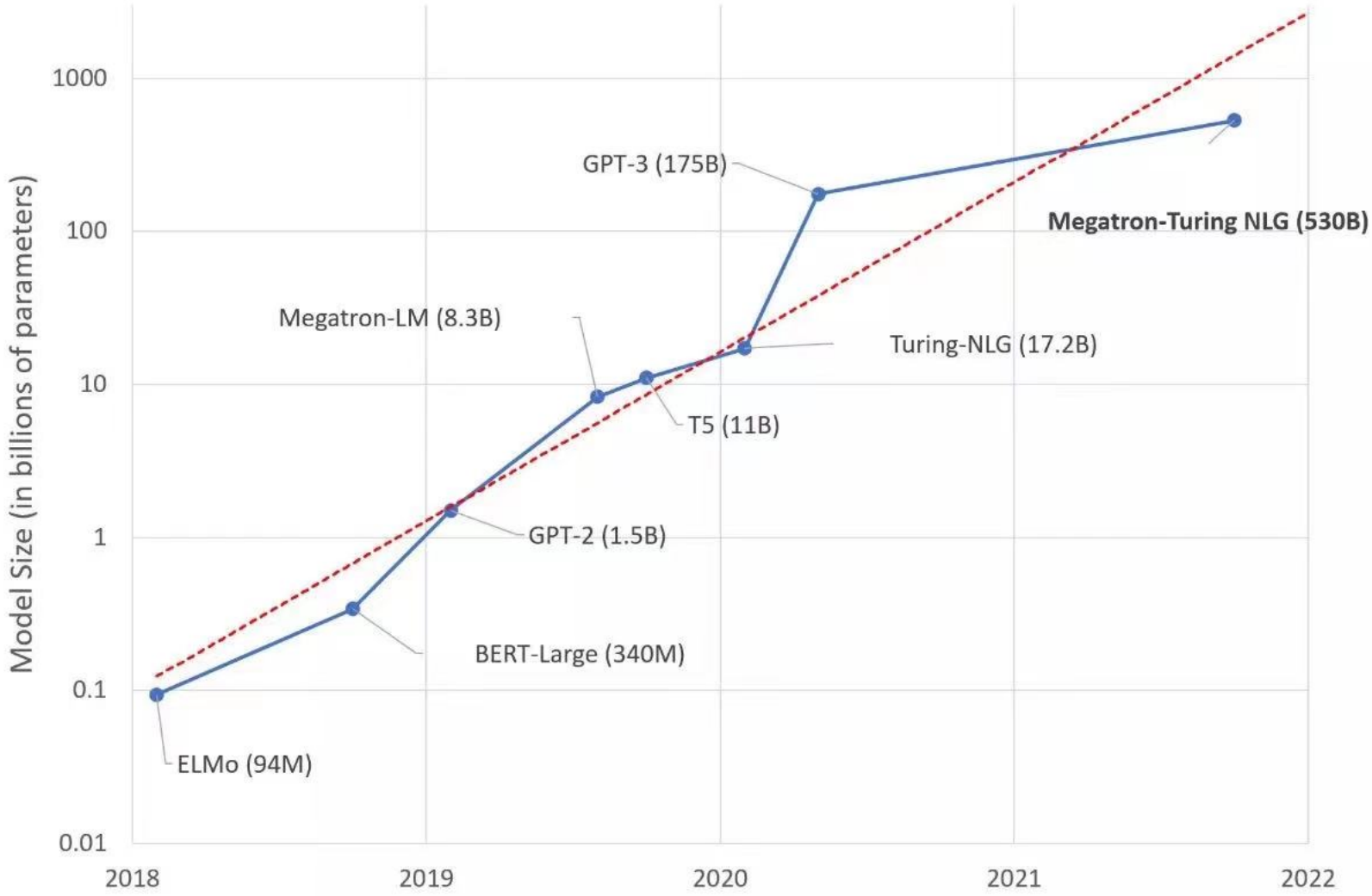
Efficient Inference
- Learned Token Pruning
- TASC

Invited/Keynote Speaker at EMDNN (NeurIPS 2016), ESWEEK 2017, EDLCV (CVPR 2017), CVPRAD (CVPR 2018) MLPCD NeurIPS (2018) LPIRC (2019), EMC^2 (NeurIPS 2019), HENP (ESWEEK 2020), EVW (ICLR 2021), ENLP (2021), Design Automation Conference 2021, VLSI SOC 2022, MLSYSArc (ISCA 2022), SustaiNLP (EMNLP 2022)

Outline

- Introduction
- LTP: A Fast Post-Training Pruning Framework for Transformers
- Staged Training for Transformer LM
- TASC: Topology-Aware Structured Communications
- DQRM: Deep Quantized Recommendation Models
- Conclusion

Model Size and Computation are Increasing



[1] Using DeepSpeed and Megatron to Train Megatron-Turing NLG 530B, the World's Largest and Most Powerful Generative Language Model.

Outline

- Introduction
- LTP: A Fast Post-Training Pruning Framework for Transformers
- Staged Training for Transformer LM
- TASC: Topology-Aware Structured Communications
- DQRM: Deep Quantized Recommendation Models
- Conclusion

How to prune without retraining?

3-stage Pruning Pipeline to retain high accuracy without retraining

1. Fisher-based Mask Search

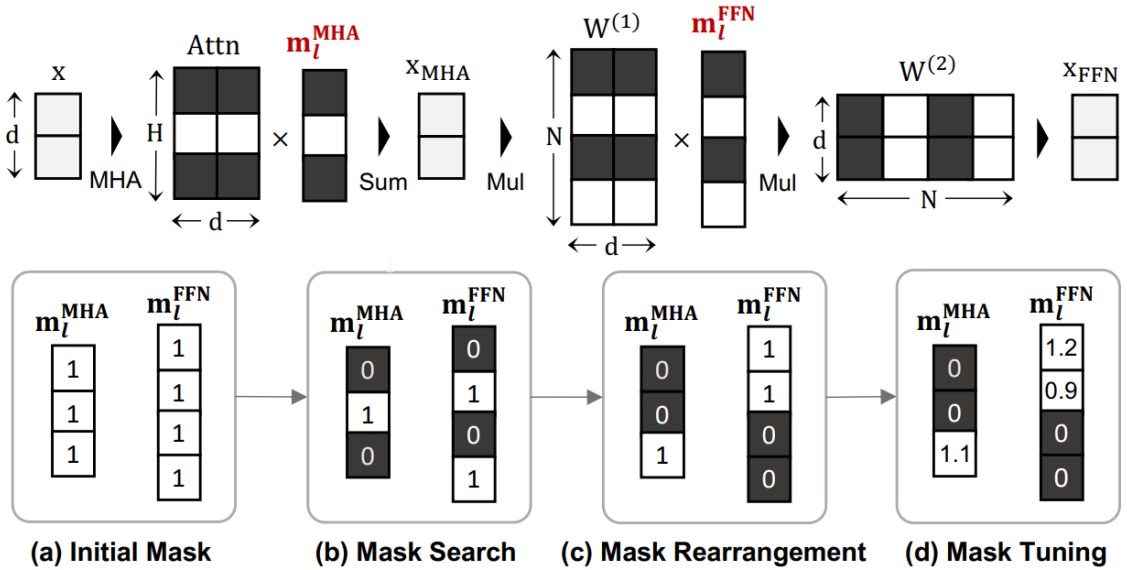
- **Finds which heads/filters to prune** based on **diagonal approximation** of the Fisher information matrix

2. Fisher-based Mask Rearrangement

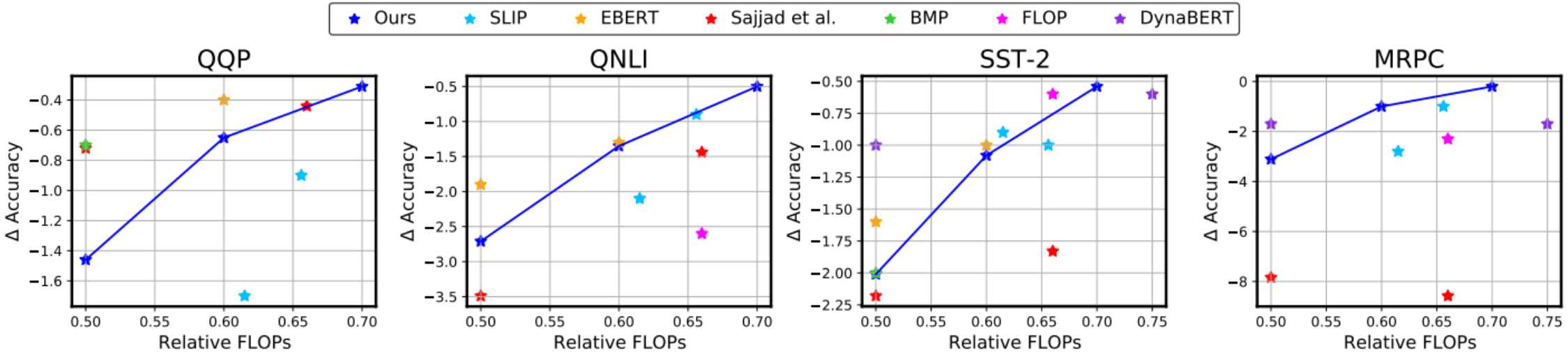
- **Rearranges the pruned heads/filters** by capturing **intra-layer interactions**

3. Mask Tuning

- **Adjusts the non-zero mask variables** to ensure that the output signal is recovered for each layer



LTP Results



	# Epochs	E2E time (hr)
DynaBERT [23]	4	12
EBERT [46]	6	5
BMP [36]	20	17
Ours	**0**	**0.01**

- Surprisingly, even ***without retraining***, our pruning strategy achieves **comparable accuracy-FLOPs tradeoff** compared to other state-of-the-art Transformer pruning methods
- End-to-end pruning time is **2~3 orders of magnitude less** than other methods (a few seconds vs. a few hours)

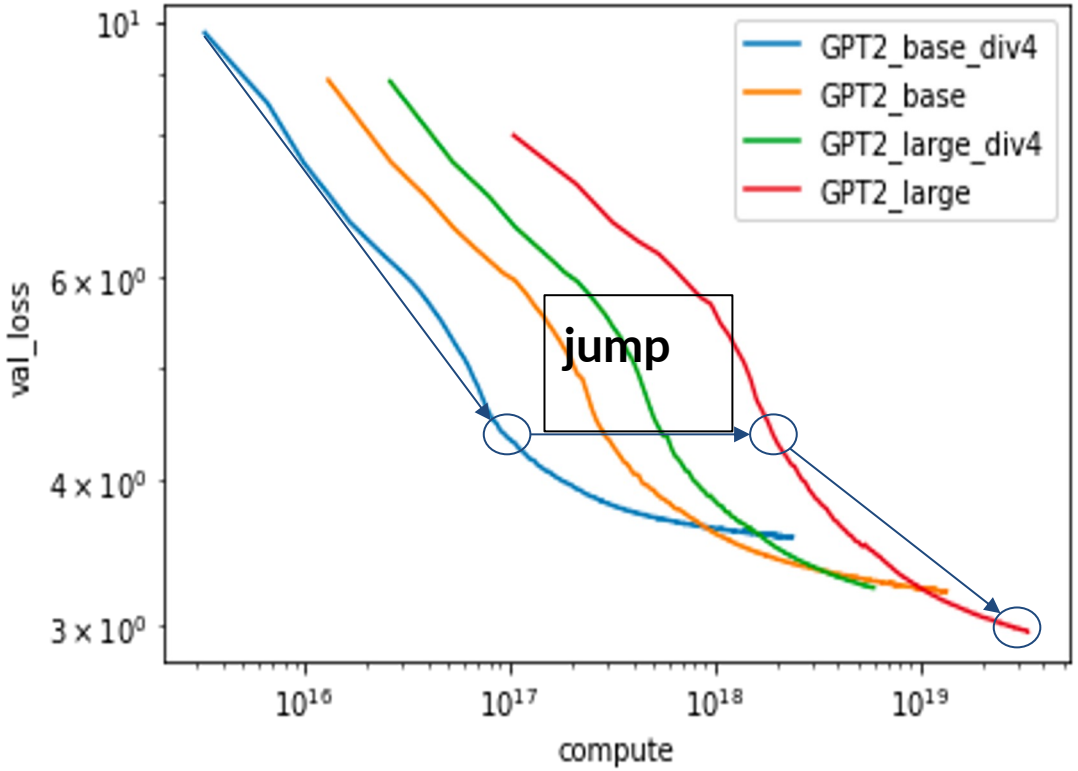
Kwon, Woosuk, Sehoon Kim, Michael W. Mahoney, Joseph Hassoun, Kurt Keutzer, and Amir Gholami. "A Fast Post-Training Pruning Framework for Transformers." KDD 2022.

Outline

- Introduction
- LTP: A Fast Post-Training Pruning Framework for Transformers
- Staged Training for Transformer LM
- TASC: Topology-Aware Structured Communications
- DQRM: Deep Quantized Recommendation Models
- Conclusion

Staged Training for Transformer-LM

- Grow GPT-small to GPT-large while being
  - **loss-preserving**
  - **training-dynamics-preserving**
- Saved up to **20%** compute in total for GPT2-large (774M) training



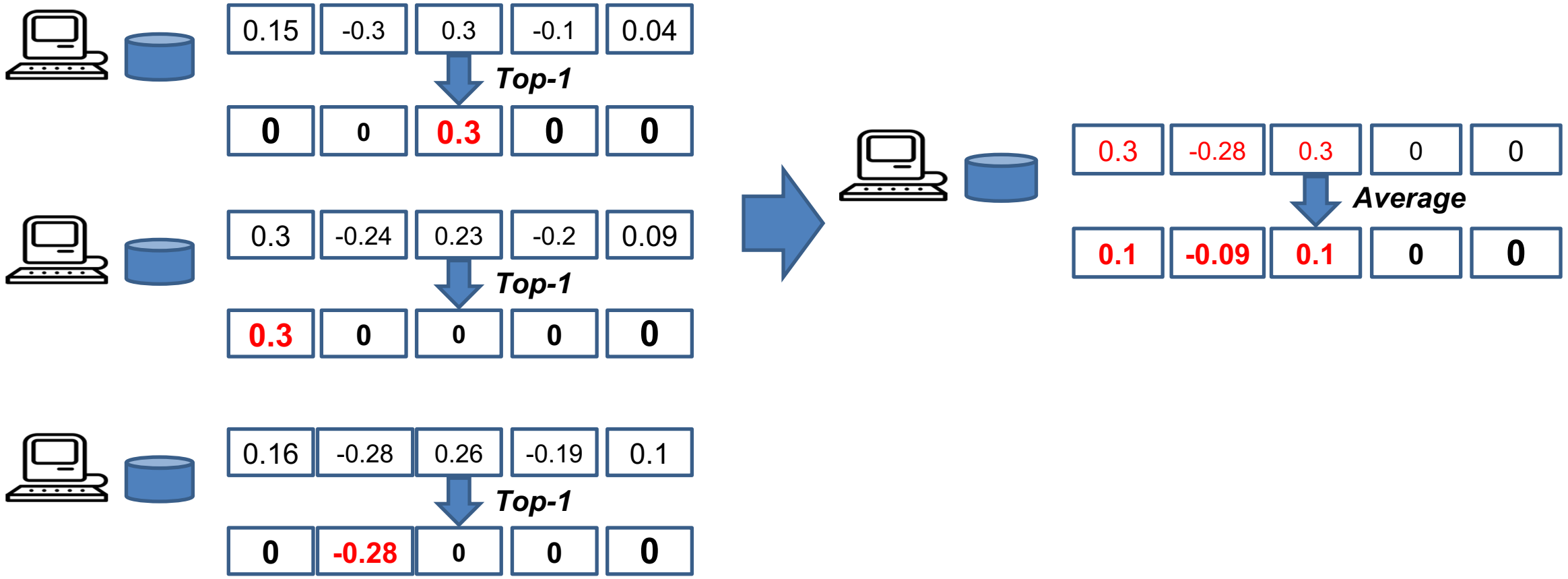
Shen, Sheng, Pete Walsh, Kurt Keutzer, Jesse Dodge, Matthew Peters, and Iz Beltagy. "Staged Training for Transformer Language Models." ICML 2022.

Outline

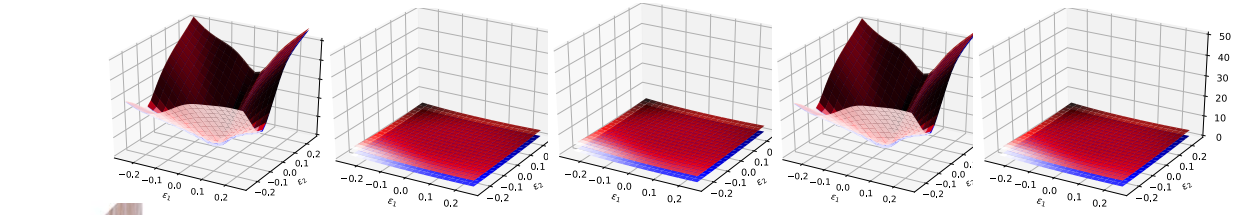
- Introduction
- LTP: A Fast Post-Training Pruning Framework for Transformers
- Staged Training for Transformer LM
- TASC: Topology-Aware Structured Communications
- DQRM: Deep Quantized Recommendation Models
- Conclusion

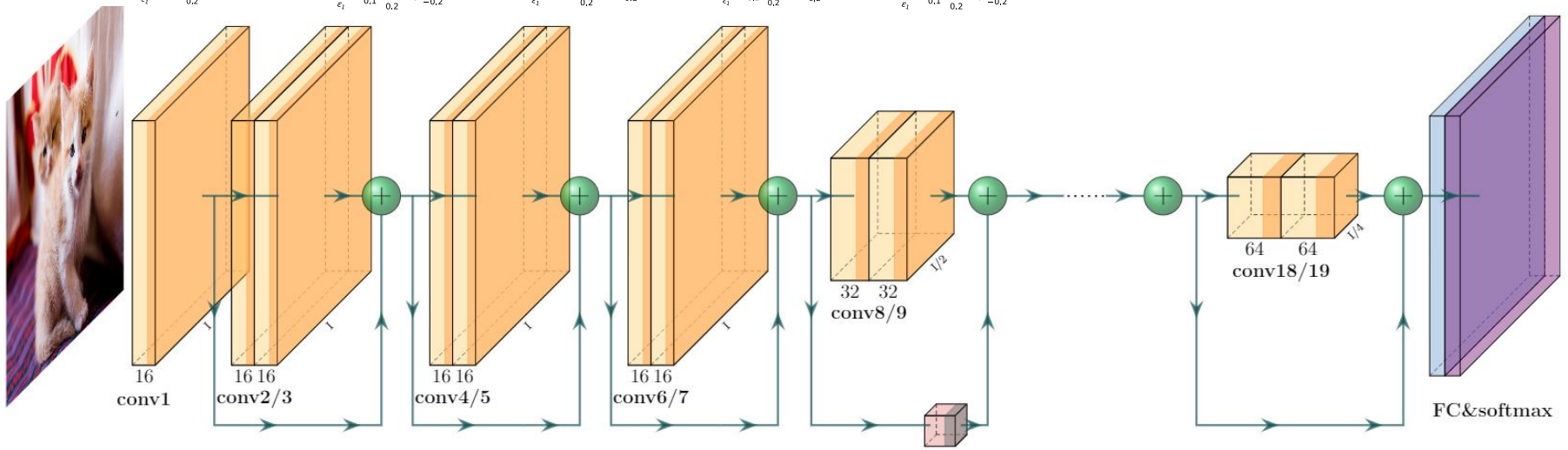
Problem I: Scalability v.s. Sparsity

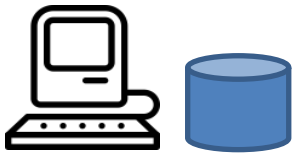
- Need very high sparsity to alleviate overhead
- However, Number of Machines ↑↑ Sparsity ↓↓



Problem II: Is current Top K selection based on correct metric?

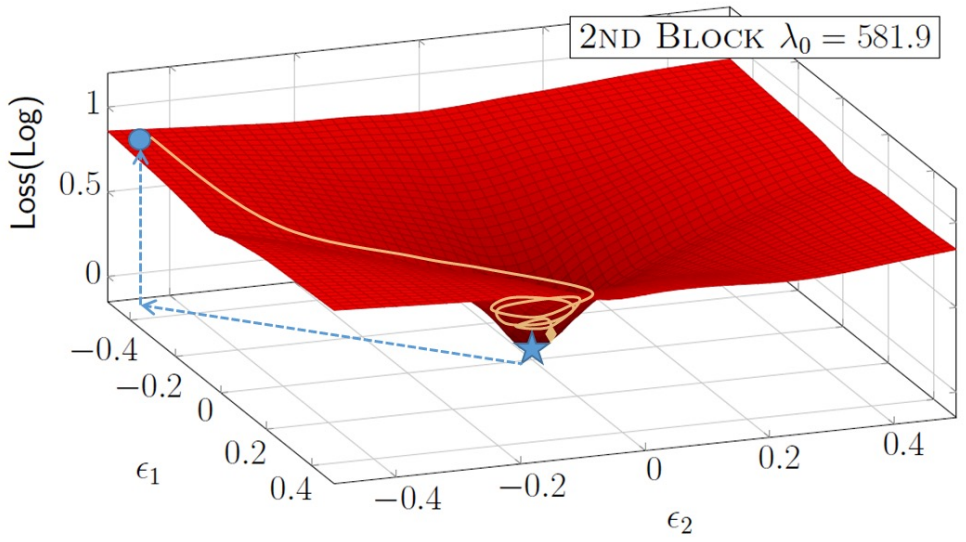


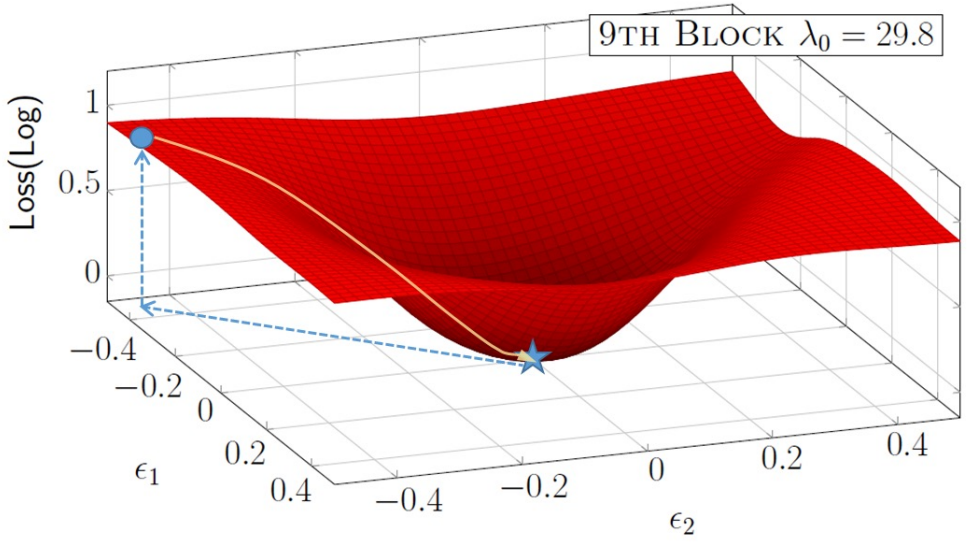




Landscape Topology Analysis

- Importance of a gradient value is relative to the topology of the loss landscape.
- A **small value** in a **sharp** loss landscape **is important**
- A **large value** in a **flat** loss landscape is **not important**





Solution: Topology-Aware Structured Communications

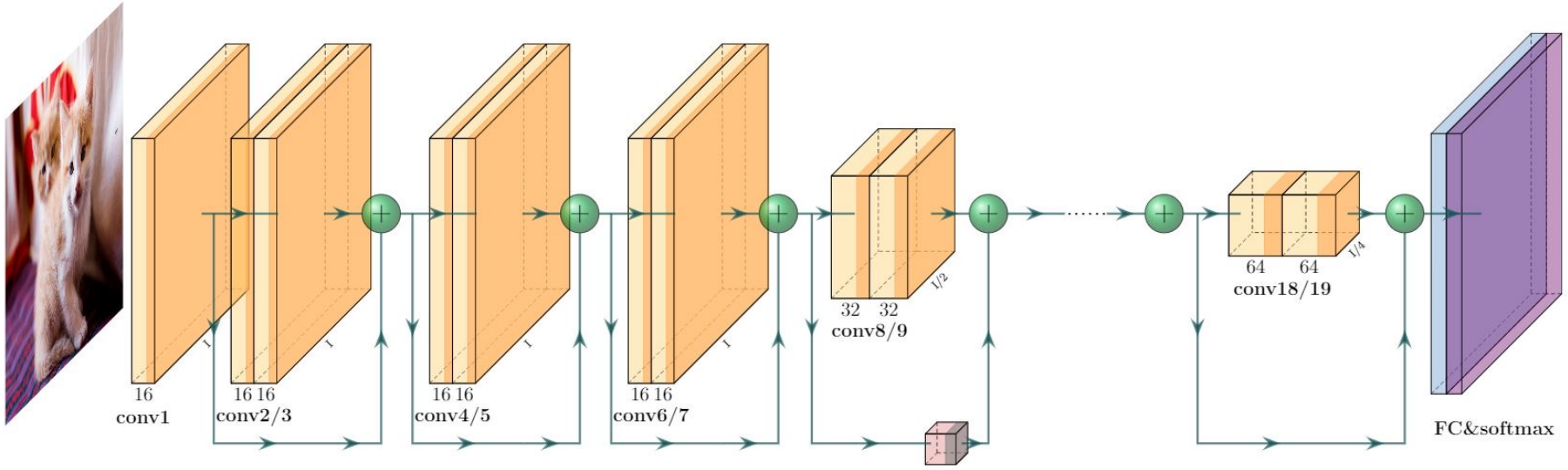
0.34	-0.3	0.15	-0.1	0.04	0.1	-0.36	0.46	-0.15	0.06

Top-2

0.34	-0.3	0	0	0	0	0	0	0	0

$$L(W+g) = L(W) + \partial L(W)^T g + \frac{1}{2} g^T H(W) g + o^n$$

$$\Delta L \approx ||g||^2 + \frac{1}{2} g^T H(W) g \approx ||g||^2 + \frac{1}{2} \frac{||g||^2}{n} Trace$$



TASC: VGG-19 on ImageNet

TASC versus DistributedDataParallel: (Single Node)

Method	Batch Size	Machine	Time (ms)	Comm. Time (ms)	Comm. SpeedUp
No Sync	128	8 K80	438	0	NA
TASC	128	8 K80	660	222	2.63
RingAllReduce + Hide	128	8 K80	915	477	1.22
RingAllReduce	128	8 K80	1021	583	1

TASC: VGG-19 on ImageNet

TASC versus DistributedDataParallel: (Multi Nodes)

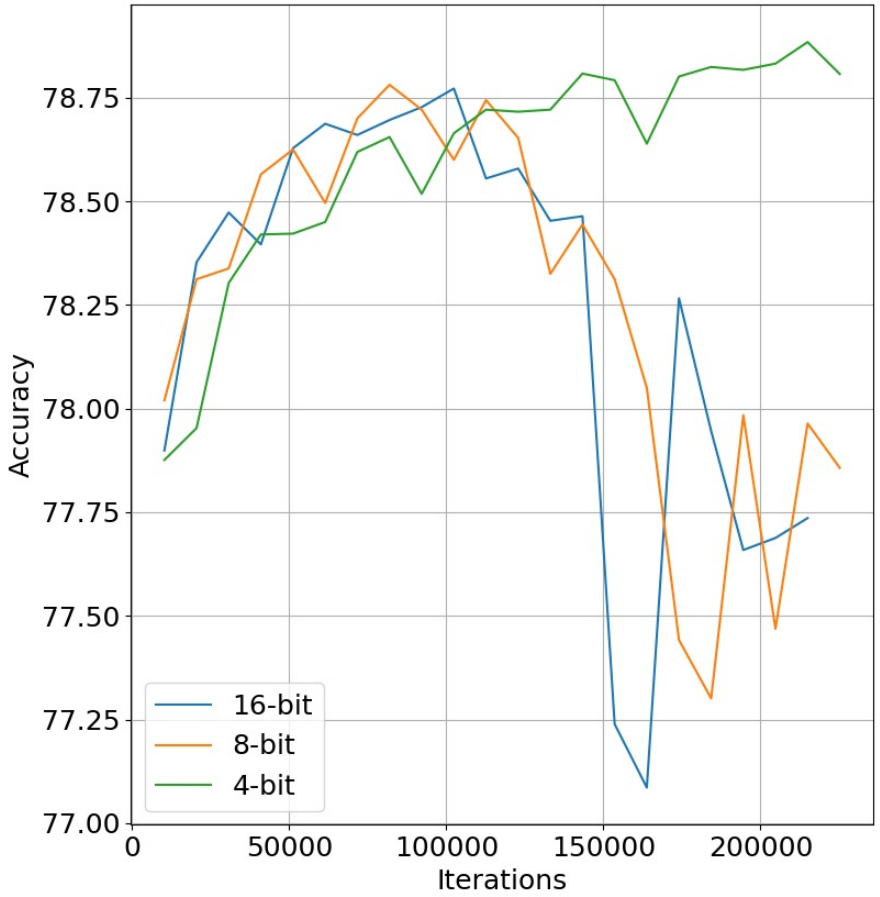
Method	Batch Size	Machine	Time (ms)	Comm. Time (ms)	Comm. SpeedUp
No Sync	128	2* 8 K80	372	0	NA
TASC	128	2* 8 K80	515	143	11.80
RingAllReduce + Hide	128	2* 8 K80	1859	1487	1.13
RingAllReduce	128	2* 8 K80	2059	1687	1

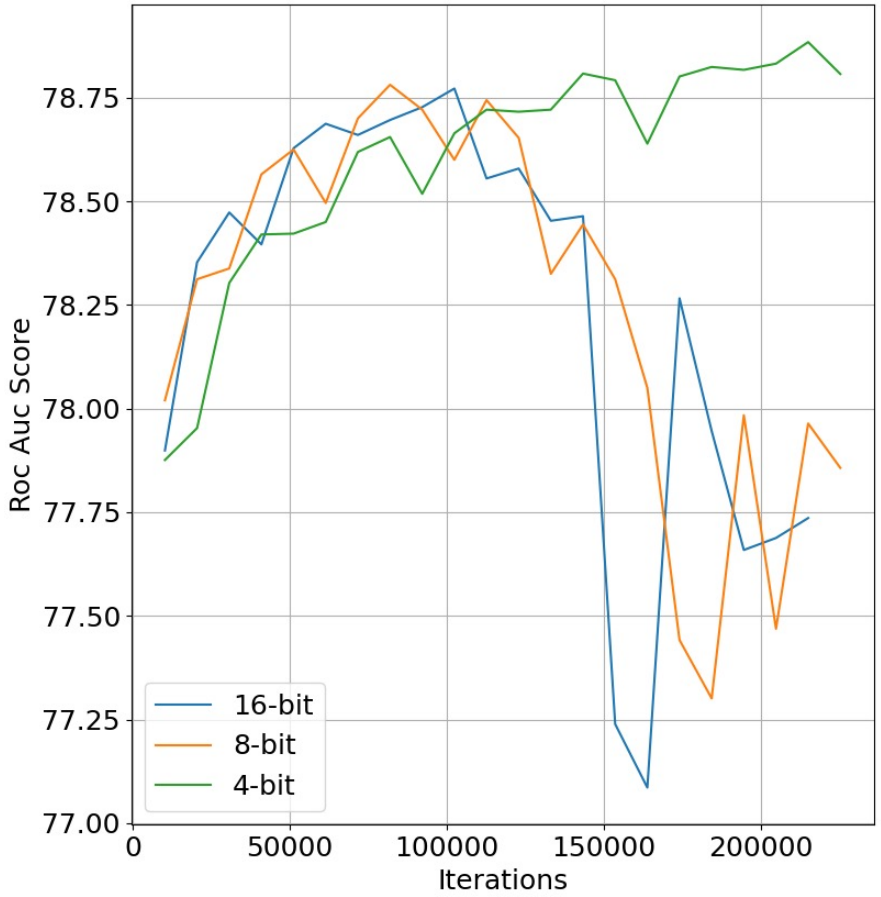
Outline

- Introduction
- LTP: A Fast Post-Training Pruning Framework for Transformers
- Staged Training for Transformer LM
- TASC: Topology-Aware Structured Communications
- DQRM: Deep Quantized Recommendation Models
- Conclusion

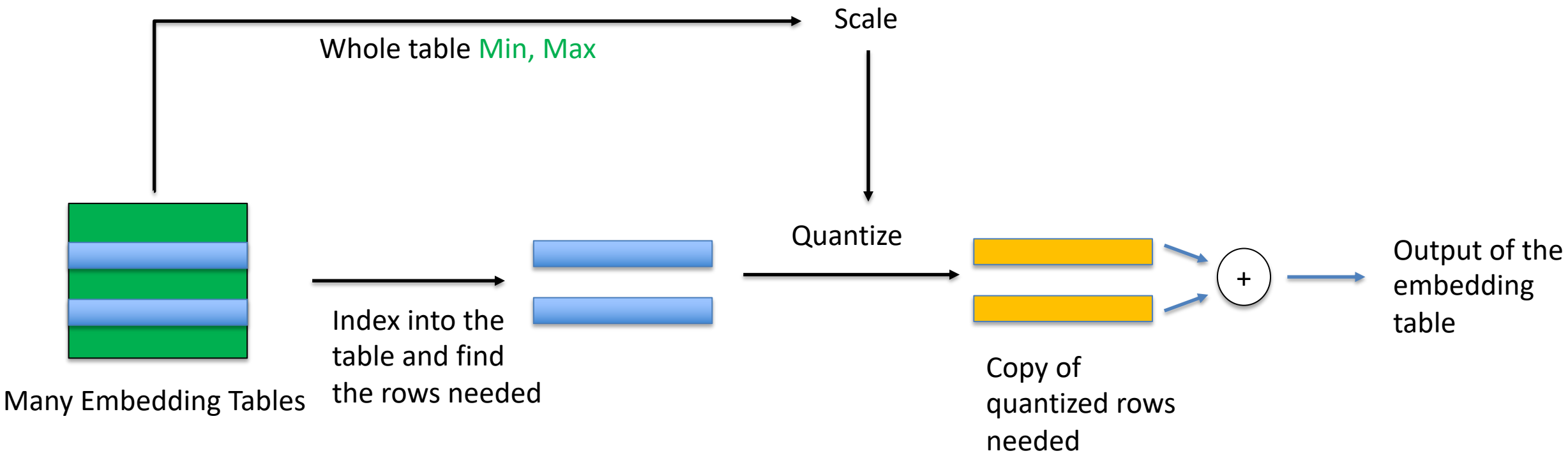
Training DLRM model leads to fast overfitting

We found that 4-bit quantization of embedding tables can alleviate the overfitting.



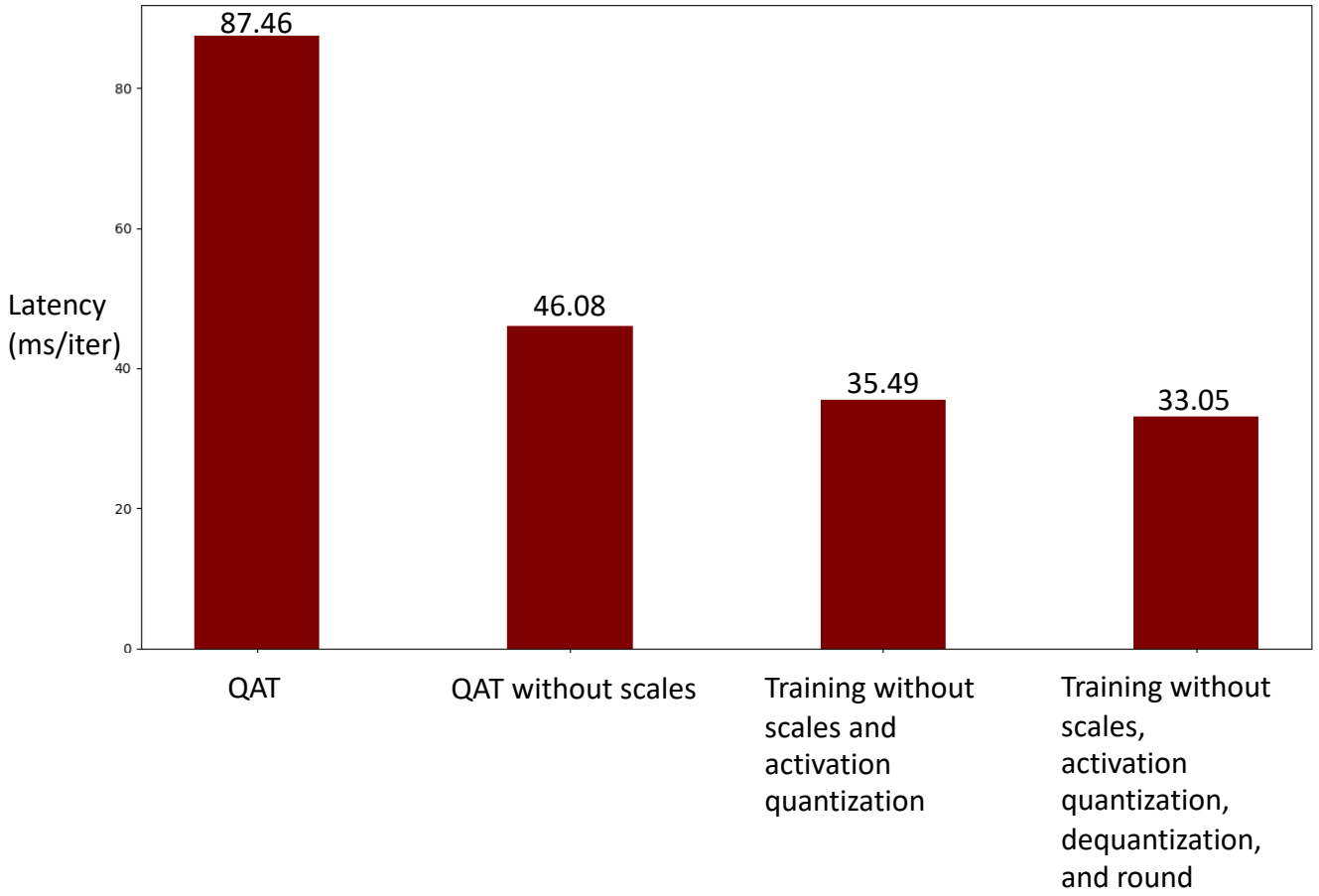


Technique No. 1
A better QAT pipeline with no copy of embedding table

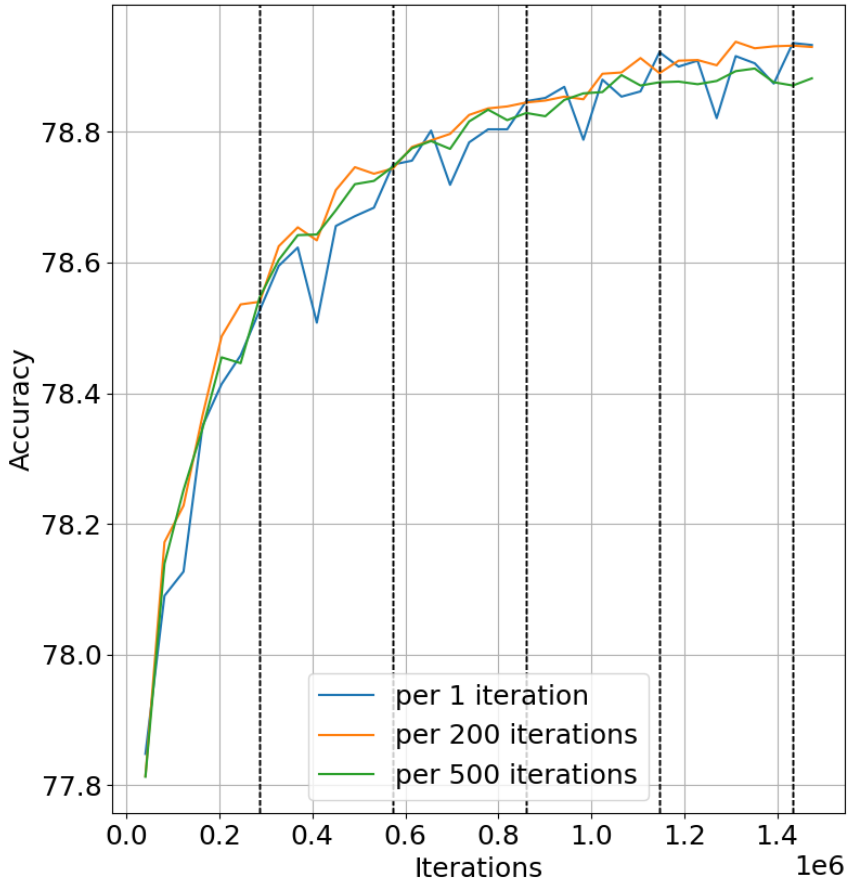


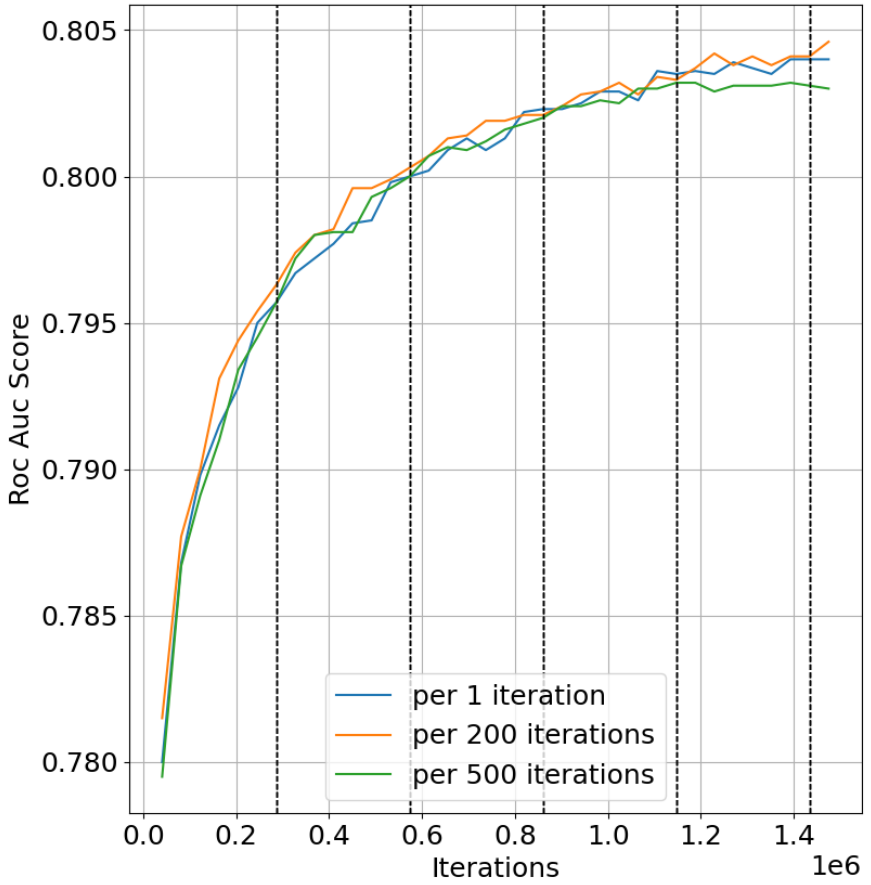
Technique No. 2
Periodic Update of Scale

- The right figure shows the bar graph of inference latency of the entire DLRM.
- Ablation studies – comparison with
  - No scaling
  - No quantization of activations
  - No quantization of activations, dequantization and round



Periodically update min and max to compute quantization scale





- We found that calculating min and max once every 200 iterations doesn't hurt accuracy.

- Periodical update of quantization scales makes the convergence under less fluctuance.

Gradient Sparsification and Quantization

Sparsification: only communicate gradient values that are used and nonzero.

Quantization: Use uniform quantization on gradients

GPUs – nccl doesn't support sparsification, only gloo backend is available, but it has many restrictions.

CPUs – gloo, mpi, and oneAPI oneCCL, we use **oneCCL** for the best support and optimization.

Backend:	Gloo		mpi		nccl	
Device	CPU	GPU	CPU	GPU	CPU	GPU
send	✓	✗	✓	?	✗	✓
recv	✓	✗	✓	?	✗	✓
broadcast	✓	✓	✓	?	✗	✓
all_reduce	✓	✓	✓	?	✗	✓
reduce	✓	✗	✓	?	✗	✓
all_gather	✓	✗	✓	?	✗	✓
gather	✓	✗	✓	?	✗	✓
scatter	✓	✗	✓	?	✗	✗
reduce_scatter	✗	✗	✗	✗	✗	✓
all_to_all	✗	✗	✓	?	✗	✓
barrier	✓	✗	✓	?	✗	✓

Outline

- Introduction
- LTP: A Fast Post-Training Pruning Framework for Transformers
- Staged Training for Transformer LM
- TASC: Topology-Aware Structured Communications
- DQRM: Deep Quantized Recommendation Models
- Conclusion

Summary

We systematically studied efficient inference and training of large neural network models.

- In LTP we accelerate the inference of Transformers.
- In Staged Training and TASC we accelerate the training of CNNs and Transformers.
- For even larger models, such as DLRM, we propose DQRM to alleviate the cost of communications during training on distributed systems.
- CPUs and oneAPI oneCCL are suitable for running the training and inference of large models like DLRM.

Open-sourced Repos:
https://github.com/allenai/staged-training
https://github.com/WoosukKwon/retraining-free-pruning
https://github.com/kssteven418/LTP

Thank you for listening!