



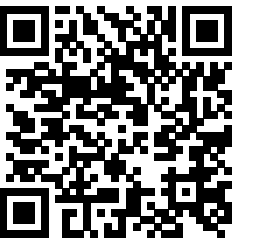
Carnegie
Mellon
University

Backprop with Approximate Activations for Memory-efficient Network Training

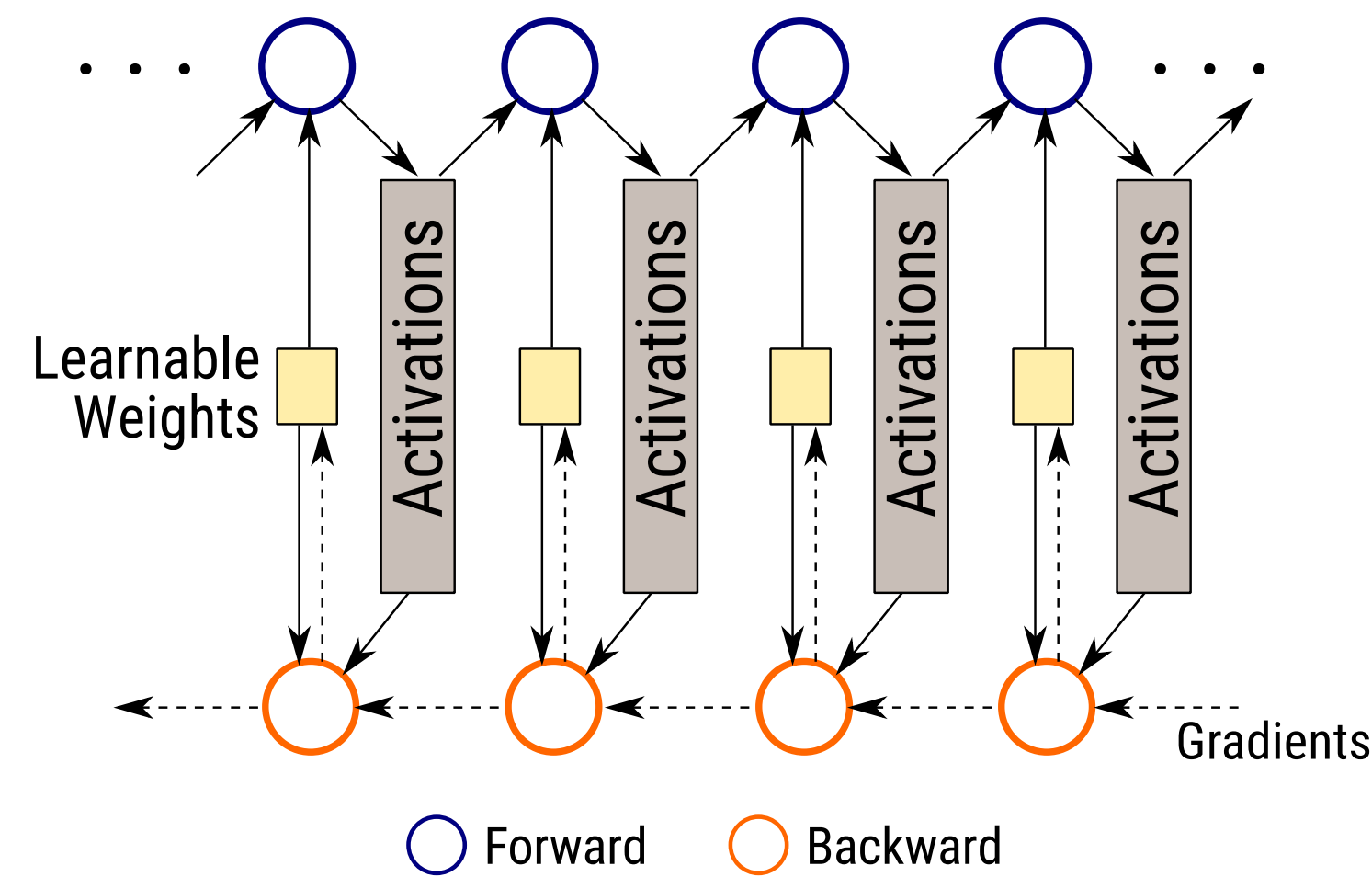
Ayan Chakrabarti
ayan@wustl.edu

Benjamin Moseley
moseleyb@andrew.cmu.edu

<https://projects.ayanc.org/blpa>

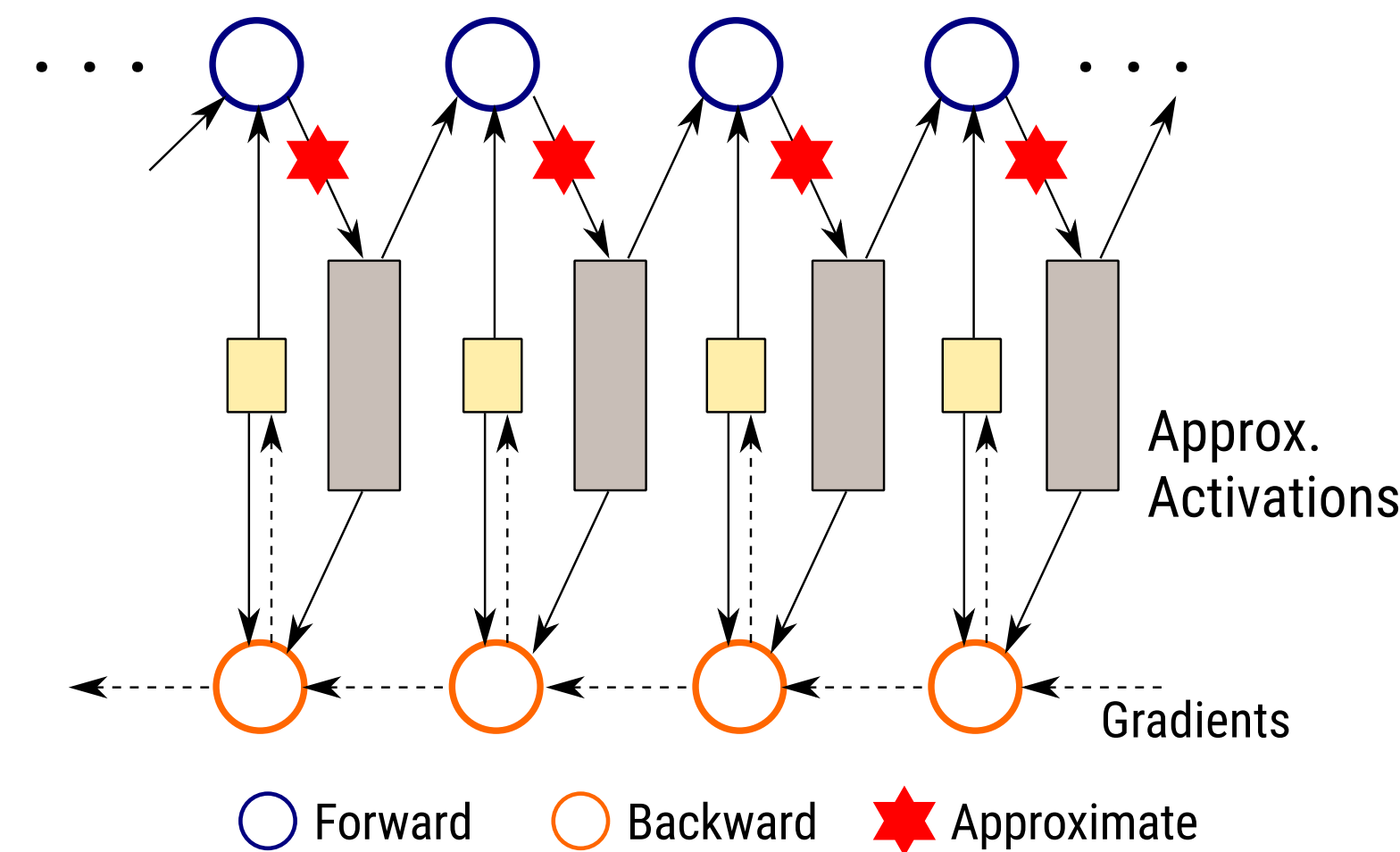


Standard Backprop



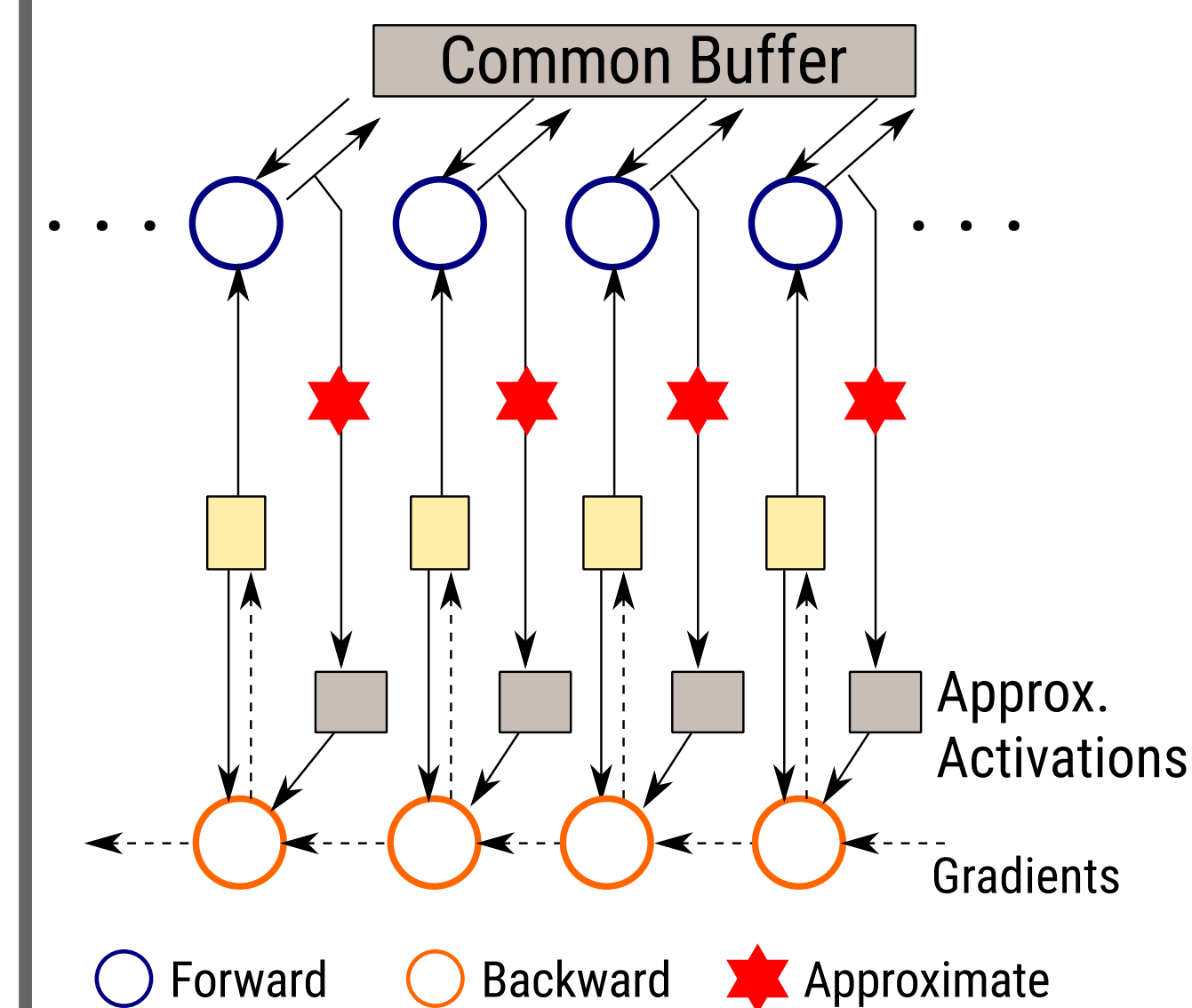
- Need to store all intermediate activations in device memory.
- **Compute becomes memory bound !**
 - Limits batch size per device.
 - Underutilizes available cores on GPUs / TPUs.

Direct Approximation



- Approximating / quantizing activations can save memory.
[Gupta et al., 2015], [Micikevicius et al., 2017], [Banner et al., 2018].
- But when we do this *within* every layer to approximate outputs:
Approximation error compounds across layers.

Proposed Method



Discard exact activations **after** use in forward, store approximate versions for use in backward.

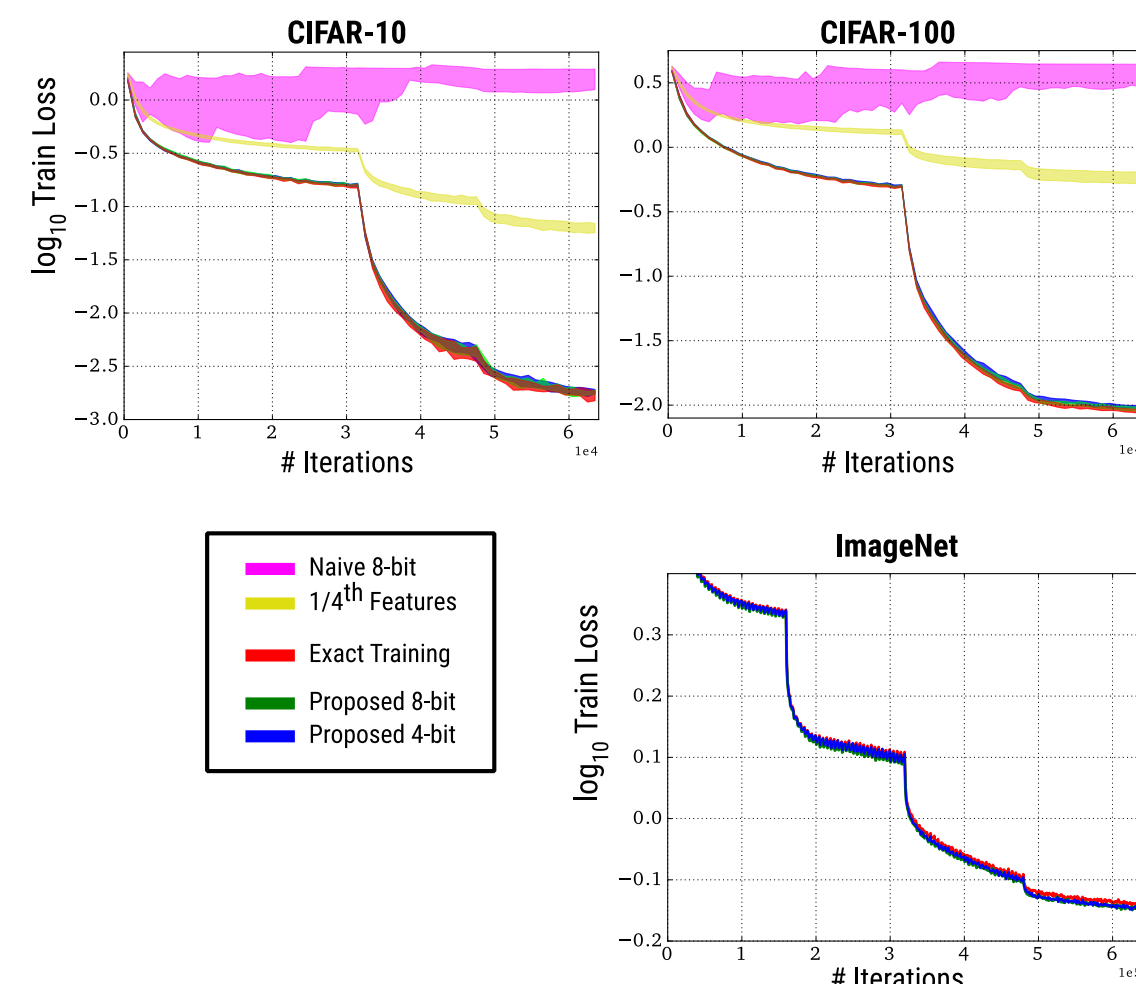
- Forward pass happens at full precision, **no approximation errors in forward pass !**
- Discard exact activations after use, **saving memory !**
- Use approximate activations to compute gradients:
For ReLU layers & sign-preserving approximation:
minimal error compounding in backward pass !

Allows for much higher approximation rate, and thus much more memory savings !

★ 32-bit Float $\Rightarrow$ 8 or 4-bit fixed point
 $\approx 4x - 8x$ less memory

Results

Training Plots



Accuracy of Trained Models

(α = approximation/memory-savings rate)

		CIFAR-10 Test Set Error	CIFAR-100 Test Set Error	ImageNet Val Set Top-5 Error
Exact	($\alpha = 1$)	5.36% $\pm$ 0.15	23.44% $\pm$ 0.26	7.20%
Exact w/ fewer features	($\alpha = 1/4$)	9.49% $\pm$ 0.12	33.47% $\pm$ 0.50	-
Naive 8-bit Approx.	($\alpha = 1/4$)	75.49% $\pm$ 9.09	95.41% $\pm$ 2.16	-
Proposed Method				
8-bit	($\alpha = 1/4$)	5.48% $\pm$ 0.13	23.63% $\pm$ 0.32	7.70%
4-bit	($\alpha = 1/8$)	5.49% $\pm$ 0.16	23.58% $\pm$ 0.30	7.72%

Maximum Batch-size & Run-time per Sample

# Layers		1001 (4x)	1001	488	254	164
Maximum Batch-size	Exact	26	134	264	474	688
	4-bit	182	876	1468	2154	2522
Run-time per Sample	Exact	130.8 ms	31.3 ms	13.3 ms	6.5 ms	4.1 ms
	4-bit	101.6 ms	26.0 ms	12.7 ms	6.7 ms	4.3 ms

Gradient Errors

