Backprop with Approximate Activations for Memory-efficient Network Training

Ayan Chakrabarti  
yaan@wustl.edu  
Benjamin Moseley  
moseleyb@andrew.cmu.edu

Introduction

- Back-propagation requires saving activations of all intermediate layers.
- With very deep networks and high-resolution activation maps, this can take a huge amount of on-device (GPU) memory!
- Computation becomes memory bound: can fit only small batches in memory, can’t use all available GPU cores.
- Limits the kind of architectures we can explore for a given problem.

Current Solutions

- Use multiple GPUs: wasteful, not using all cores on each GPU.
- Checkpointing: requires re-computing forward passes.
- Just use 16-bit (or smaller) representations; errors build up across layers.

Simple Idea: Replace activations with low-precision versions after using the exact versions for the forward computation.

- Forward pass happens at full precision, no build-up of approximation errors.
- Discard full-precision activations as soon as we use them, saving memory.
- Use low-precision activations for approximate gradient computation.
- For ReLU layers: gradients flowing back to the input of each layer only depend on sign of activations. Prevents build-up of errors in backward pass too!

Proposed Method

Results

Training Plots

- Accuracy of Models Trained for CIFAR and ImageNet with Exact Training vs Our Method

Maximum Batchsize & Run-time for different architectures with Exact Training vs Our Method