Dynamic Block Sparse Reparameterization of Convolutional Neural Networks

Dharma Teja*, Girish Varma, Kishore Kothapalli
Center for Security, Theory and Algorithmic Research, IIIT Hyderabad, India

MOTIVATION

● Sparse neural networks are efficient in memory and compute. But has poor run time performance on parallel architectures like GPU/TPU. Only way out is structured sparse neural networks.

● Structured sparse neural networks obtained by pruning are suboptimal as structure is induced after training.

● Need for an approach where the structure is integrated into the training process.

BLOCK SPARSITY

Non zero elements in the matrix are arranged in the form of blocks of size (bh,bw).

WHY ?

● Has good run time performance on parallel architectures and leads to ideal speedups. (50% sparsity results in ~2x speedup)

● A generic sparsity pattern, with channel and filter sparsity patterns as sub cases.

OUR APPROACH (DBSR)

For a given convolutional layer, divide the dense 4D weight tensor \((ofm,ifm,kh,hw)\) into blocks of size \((bh,bw,kh,kw)\) by performing blocking on outer two dimensions.

Assign a trainable scaling parameter to a block and scale the block during the forward pass.

Push scaling parameters \(S\) to zero by adding \((\gamma|S|)\) to the loss function. Hyper parameter \(\gamma\) controls the amount of sparsity.

RESULTS

For the block size of 32x32, parameters and FLOPS are reduced by 30% for Resnet50/Imagenet, and 50% for ResneXt50/Imagenet with only ~0.5 increase in Top-1 error.

COMPARISON WITH BLOCK PRUNING

VGG11/CIFAR100 with block size 32x32

Resnet20/CIFAR100 with block size 8x8

CONCLUSION

● Using DBSR approach, structured sparse neural networks can be generated which are efficient in compute, memory and runtime.

● DBSR is easy to use as it uses only one extra hyper parameter apart from those used for training a dense model.

CODE : https://github.com/idharmateja/bsnn