# Dynamic Block Sparse Reparameterization of Convolutional Neural Networks

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

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

ICCV 2019
Seoul, Korea

## MOTIVATION

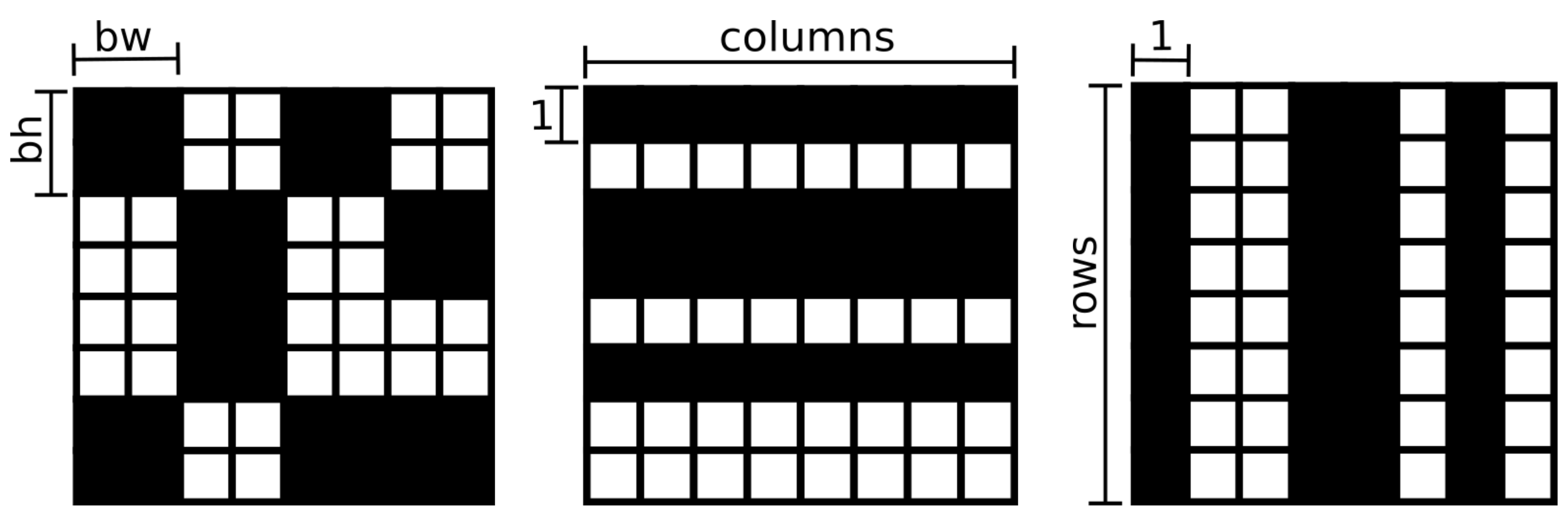
- Sparse neural networks are efficient in memory and compute. But has poor run time performance on parallel architectures like GPU/TPU. Only way out is structured sparse neural networks.
- Structured sparse neural networks obtained by pruning are suboptimal as structure is induced after training.
- Need 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 **(ɣ*|S|)** to the loss function. Hyper parameter **ɣ** controls the amount of sparsity.

$$\underset{S}{\begin{bmatrix} 0.5 & 0.1 \\ 0.2 & 0.3 \end{bmatrix}} *_e \underset{W}{\begin{bmatrix} 1 & 8 & 8 & 6 \\ 4 & 5 & 3 & 2 \\ 1 & 4 & 9 & 5 \\ 6 & 7 & 2 & 1 \end{bmatrix}} = \underset{W_S}{\begin{bmatrix} 0.5 & 4 & 0.8 & 0.6 \\ 2 & 2.5 & 0.3 & 0.2 \\ 0.2 & 0.8 & 2.7 & 1.5 \\ 1.2 & 1.4 & 0.6 & 0.3 \end{bmatrix}}$$

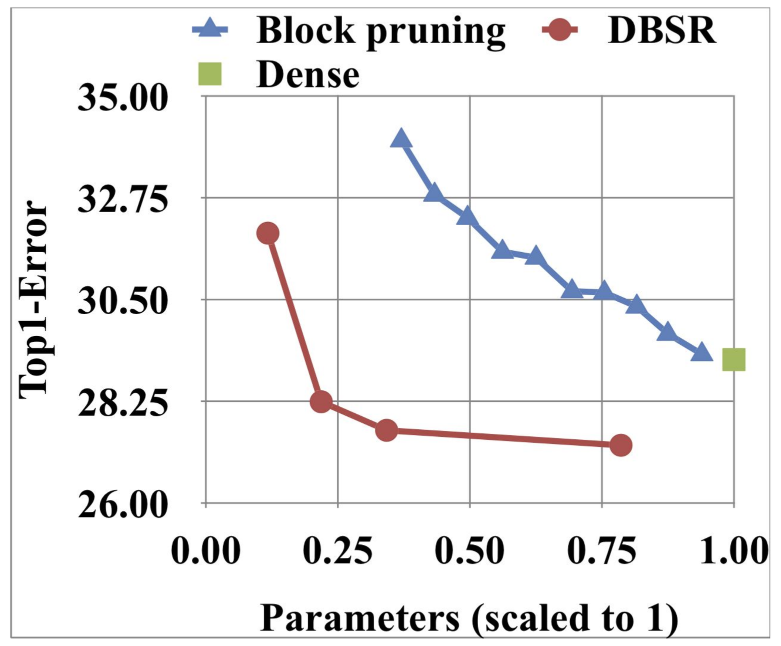
## RESULTS

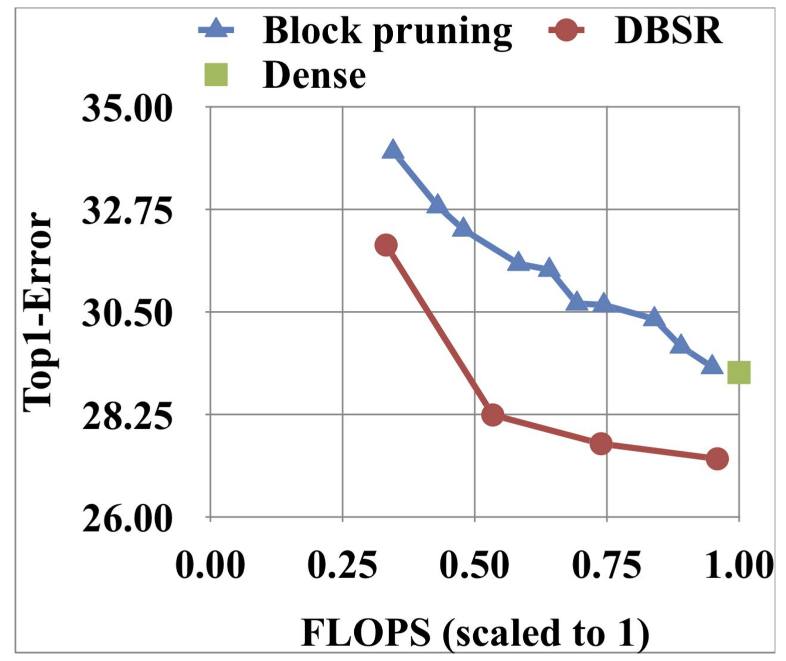
For the block size of 32x32, parameters and FLOPS are reduced by 30% for Resnet50/ Imagenet, and 50% for ResneXt50/Imagenet with only ~0.5 increase in Top-1 error.

| Model | Top-1 Error | Top-5 Error | Params | #FLOPs |
|---|---|---|---|---|
| ResNet-34-pruned [12] | 27.44 | - | 19.9M | 3.08B |
| Resnet-50-pruned [12] ( From [9]) | 27.12 | 8.95 | - | 3.07B |
| Resnet-50-pruned(2x) [8] | 27.70 | 9.20 | - | 2.73B |
| Resnet50-pruned (ThinNet-30) [17] | 27.96 | 9.33 | 16.94M | 2.44B |
| **Resnet-50-32x32-B (Ours)** | **26.36** | **8.23** | **13.36M** | **2.19B** |
| Resnet-101-pruned [32] | 25.44 | - | 17.30M | 3.69B |
| **Resnet50-32x32-A (Ours)** | **25.08** | **7.73** | **17.89M** | **2.74B** |

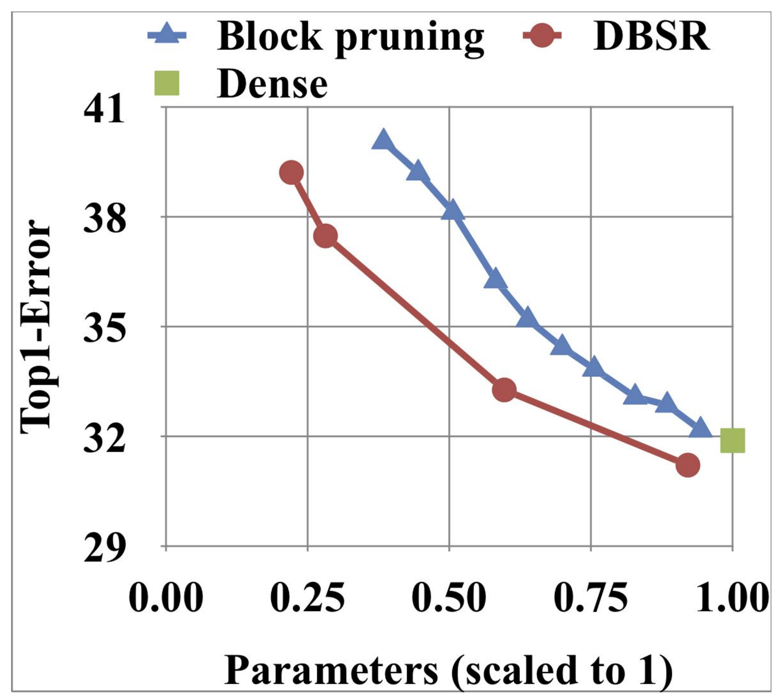
## COMPARISON WITH BLOCK PRUNING

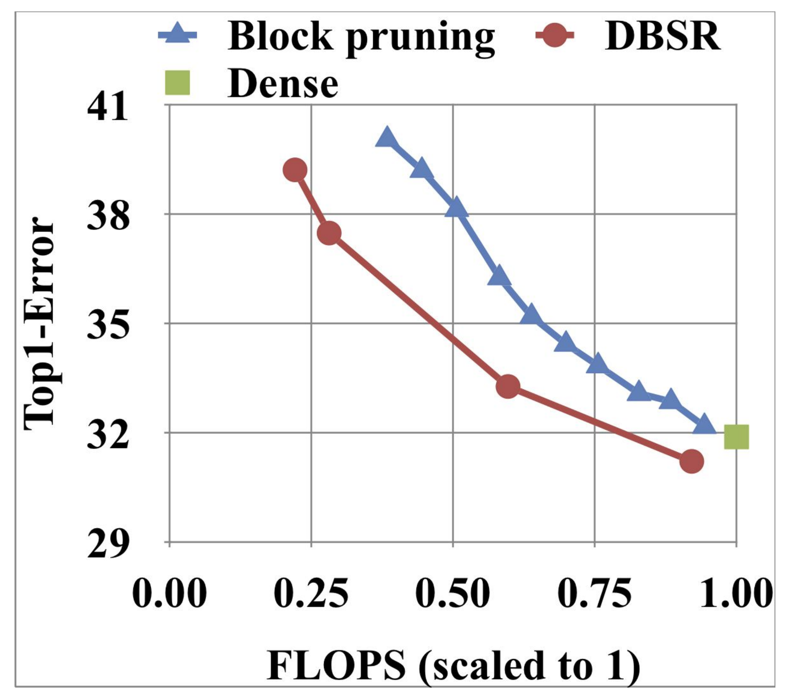
VGG11/CIFAR100 with block size 32x32




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**