# Hybrid Binary Networks

**A. Prabhu**, V. Batchu, R. Gajawada, A. Munagala, A. Namboodiri

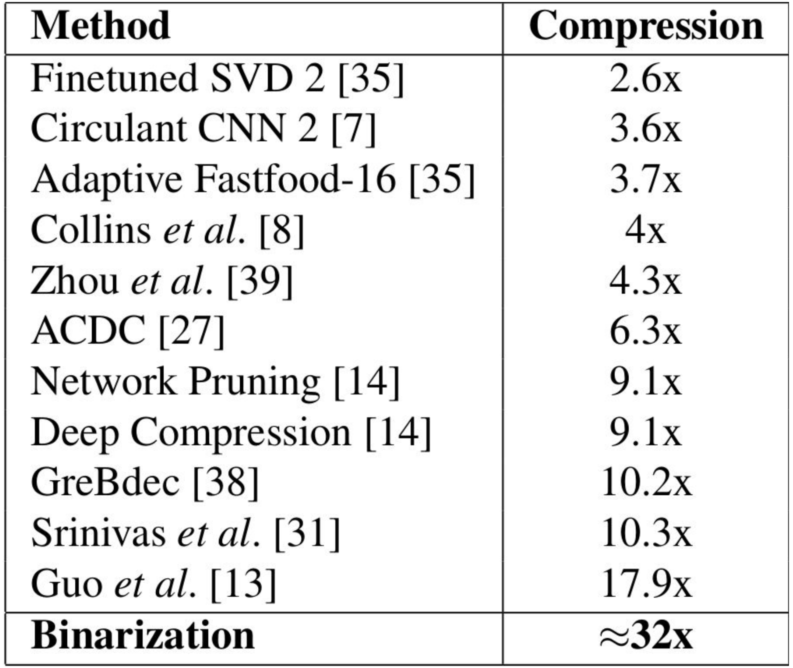
Center For Visual Information Technology, KCIS, IIIT Hyderabad, India

# Why compress neural networks?

- Reduce Neural Network size
- Lesser memory overhead, faster computation
- Lesser power consumption

# Why Binarization?

- Extreme form of Quantization
- Layer weights and activations mapped to {-1, 1}
- Allows XNOR-Popcount operations for convolutional operations
- x58 speedup, x32 compression rate
- High accuracy losses!
  (Examples - XNOR Net)



| **Method** | **Compression** |
|---|---|
| Finetuned SVD 2 [35] | 2.6x |
| Circulant CNN 2 [7] | 3.6x |
| Adaptive Fastfood-16 [35] | 3.7x |
| Collins *et al.* [8] | 4x |
| Zhou *et al.* [39] | 4.3x |
| ACDC [27] | 6.3x |
| Network Pruning [14] | 9.1x |
| Deep Compression [14] | 9.1x |
| GreBdec [38] | 10.2x |
| Srinivas *et al.* [31] | 10.3x |
| Guo *et al.* [13] | 17.9x |
| **Binarization** | **$\approx$32x** |

# Why binarize the entire network?

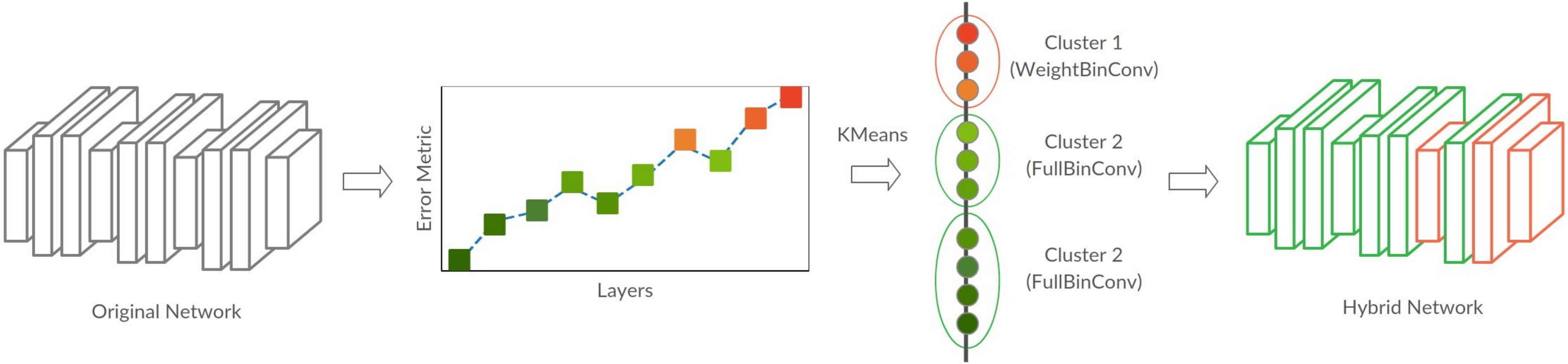
- Some convolutional Layers are **highly** computationally intensive, some others are not as intensive.
- Binarizing high-computation layers with low binarization error is much more useful than low-computation layers with high binarization errors.
- Our Contribution: Analyse *where* in the network it's useful to binarize activations, and binarize only those parts!

# Binarization Metric (M)

- Two factors, E and NF $\mathbf{M} = \mathbf{E} + \gamma \cdot \frac{1}{\mathbf{NF}}$
- $\mathbf{E} = \frac{\| \mathbf{I} - \mathbf{I_B} \|^2}{n}$ tells us how much informational loss on binarization
- Higher $\mathbf{E}$ -> Binarization there becomes less useful
- $\mathbf{NF}$ (Number of Flops) indicates computational contribution of layer
- Higher $\mathbf{NF}$ -> Binarization becomes more useful
- All weights are binarized, we choose where to binarize inputs. Weight binarization does not affect accuracies, experimentally

# Network Conversion Algorithm

- Graphical Flow:



# Results on Imagenet

- We perform experiments on Imagenet using AlexNet and ResNet-18 architectures, which are popularly used to benchmark performance of binary networks.
- We see a 4.9% and 3.6% increase in accuracy of our proposed models over the traditional FBin counterparts with a negligible increase in the number of FLOPs.

| **Model** | **Method** | **Accuracy** | | **Memory Savings** | **FLOPs** |
|---|---|---|---|---|---|
| | | Top-1 | Top-5 | | |
| AlexNet | FPrec | 57.1% | 80.2% | 1x | 1135 (9.4x) |
| | WBin (BWN) | 56.8% | 79.4% | 10.4x | 780 (6.4x) |
| | FBin (XNOR) | 43.3% | 68.4% | 10.4x | **121 (1x)** |
| | Hybrid-1 | 48.6% | 72.1% | 10.4x | 174 (1.4x) |
| | Hybrid-2 | **48.2%** | **71.9%** | **31.6x** | 174 (1.4x) |
| Increase | Hybrid vs FBin | +4.9% | +3.5% | +21.2x | +53 (+0.4x) |
| ResNet-18 | FPrec | 69.3% | 89.2% | 1x | 1814 (13.5x) |
| | WBin (BWN) | 60.8% | 83.0% | 13.4x | 1030 (7.7x) |
| | FBin (XNOR) | 51.2% | 73.2% | 13.4x | **134 (1x)** |
| | Hybrid-1 | 54.9% | 77.9% | 13.4x | 359 (2.7x) |
| | Hybrid-2 | **54.8%** | **77.7%** | **31.2x** | 359 (2.7x) |
| Increase | Hybrid vs FBin | +3.6% | +4.5% | +17.8x | +225 (+1.7x) |

# Comparison with other approaches

- Outperforms nearly every binary/ternary network, while preserving almost maximal compression.
- Hybrid-2 records 47.4% accuracy on AlexNet, 4.9% higher than XNOR-Net, the algorithm that we build upon.
- A 1.2% increase in accuracy as DoReFa-Net (2-bit activations), with hybrid 1-bit activations, with increased compression, similarly 1.6% higher accuracy than HTCBN with similar compression rates.
- Hybrid-2 records 54.8% accuracy on ResNet-18, 3.6% higher than XNOR-Net, 1.2% higher than HTCBN (2-bit activations).

| **Technique** | **Acc-Top1** | **Acc-Top5** | **W/I** | **Mem** | **FLOPs** |
|---|---|---|---|---|---|
| **AlexNet** | | | | | |
| BNN | 39.5% | 63.6% | 1/1 | 32x | 121 (1x) |
| XNOR | 43.3% | 68.4% | 1/1 | 10.4x | **121 (1x)** |
| Hybrid-1 | 48.6% | 71.7% | 1/1 | 10.4x | 174 (1.4x) |
| Hybrid-2 | **48.2%** | **71.5%** | 1/1 | **31.6x** | 174 (1.4x) |
| HTCBN | 46.6% | 71.1% | 1/2 | 31.6x | 780 (6.4x) |
| DoReFa-Net | 47.7% | - | 1/2 | 10.4x | 780 (6.4x) |
| **Res-Net 18** | | | | | |
| BNN | 42.1% | 67.1% | 1/1 | 32x | 134 (1x) |
| XNOR | 51.2% | 73.2% | 1/1 | 13.4x | **134 (1x)** |
| Hybrid-1 | 54.9% | 77.9% | 1/1 | 13.4x | 359 (2.7x) |
| Hybrid-2 | **54.8%** | **77.7%** | 1/1 | **31.2x** | 359 (2.7x) |
| HTCBN | 53.6% | - | 1/2 | 31.2x | 1030 (7.7x) |

# Experiments on Sketch datasets

- We also perform experiments on TU-Berlin and Sketchy datasets (one of the largest and most popular sketch datasets) and show significant improvements
- On Sketch-A-Net we observe a 13.5% and 15% improvement on TU-Berlin and Sketchy respectively.
- On ResNet-18 we observe a 5% and 5.1% improvement.

| Model | Method | Accuracy | | Memory Savings | FLOPs (in M) |
|---|---|---|---|---|---|
| | | TU-Berlin | Sketchy | | |
| Sketch-A-Net | FPrec | 72.9% | 85.9% | 1x | 608 (7.8x) |
| | WBin (BWN) | 73% | 85.6% | 29.2x | 406 (5.2x) |
| | FBin (XNOR) | 59.6% | 68.6% | 19.7x | **78 (1x)** |
| | Hybrid | **73.1%** | **83.6%** | **29.2x** | **85 (1.1x)** |
| Increase | Hybrid vs FBin | +13.5% | +15.0% | +9.5% | +7 (+0.1x) |
| ResNet-18 | FPrec | 74.1% | 88.7% | 1x | 1814 (13.5x) |
| | WBin (BWN) | 73.4% | 89.3% | 31.2x | 1030 (7.7x) |
| | FBin (XNOR) | 68.8% | 82.8% | 31.2x | **134 (1x)** |
| | Hybrid | **73.8%** | **87.9%** | **31.2x** | 359 (2.7x) |
| Increase | Hybrid vs FBin | +5.0% | +5.1% | - | +225 (+1.7x) |
| Sketch-A-Net | FPrec | 72.9% | 85.9% | 1x | 1135 (12.3x) |
| Squeezenet | FPrec | 71.2% | 86.5% | 8x | 610 (6.6x) |
| Squeezenet | WBin | 66.7% | 81.1% | 23.7x | 412 (4.5x) |
| Squeezenet | FBin | 56.8% | 66.0% | 23.7x | **92 (1x)** |
| Squeezenet | Hybrid | **64.8%** | **79.6%** | **23.7x** | 164 (1.8x) |
| Improvement | Hybrid vs FBin | +8.0% | +13.6% | - | +72 (+0.8x) |

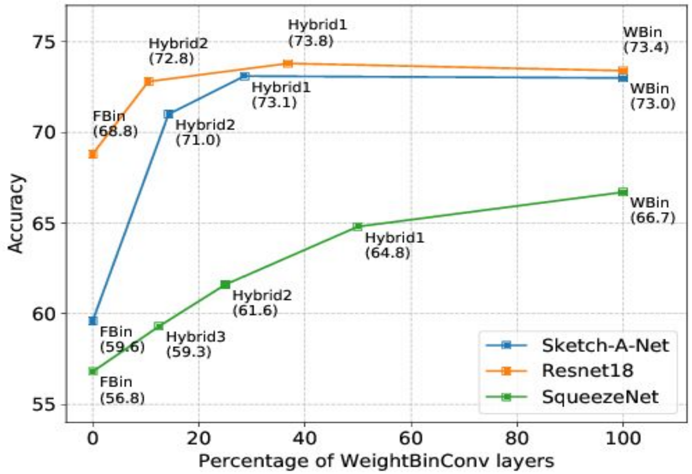
# Effects of last layer weight binarization

- XNOR-Nets lose ~10% accuracy on last-layer weight binarization on Sketch-A-Net
- Other approaches too generally avoid last-layer binarization due to accuracy drops
- Our network loses only 1% accuracy on last-layer binarization

| Model | BinType | Last Bin? | Acc | Mem |
|---|---|---|---|---|
| Sketch-A-Net | FBin (XNOR) | No | 59.6% | 19.7x |
| | | Yes | 48.3% | **29.2x** |
| Sketch-A-Net | Hybrid | No | 73.1% | 19.7x |
| | | Yes | 72.0% | **29.2x** |
| Resnet-18 | FBin (XNOR) | No | 69.9% | 13.4x |
| | | Yes | 68.8% | **31.2x** |
| Resnet-18 | Hybrid | No | 73.9% | 13.4x |
| | | Yes | 73.8% | **31.2x** |

# Why hybrid models?

- Initially, small increases in the percentage of WeightBinConv layers improves the accuracy significantly in all models without much of an increase in the number of FLOPs.



# Conclusions and Take-aways

- Selective binarization by our proposed metric strikes balance between performance, memory-savings and accuracy.
- This also gives a simple and successful way to binarize last-layer weights without significant accuracy drops enabling compression rates of nearly 32x!
- This approach can be used along with other compression techniques, e.g. SqueezeNet.
- We hope that this project encourages more investigations into working with binary networks for all the cool applications presented at WACV '18!

# Thanks!

(Looking for a 6-12 month RAship/Internship!
Please let me know there are any openings)

I’m available in the poster session.
ameya.prabhu@research.iiit.ac.in