Distribution-Aware Binary Neural Networks for Sketch Recognition

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

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

Quick Recap: Why Binarization?

- Extreme form of Quantization
- Layer weights and activations mapped to {-1, 1}
- Allows XNOR-Popcount operations for convolutional operations
- x64 speedup, x16 compression rate
- High accuracy losses! (Examples - XNOR)

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	**≈32x**

Expressivity

- Are Binary Networks as expressible as infinitely precise ones?
- Consider p(x) be a multivariate monomial, expressed as the product of **n** numbers as assumed in [Lin et al. 2017].

$$\prod_{i=1}^{n} x_i = \frac{1}{2^n} \sum_{\{s\}} s_1 ... s_n \sigma(s_1 x_1 + ... + s_n x_n),$$

- Can be implemented using a flat network (one hidden layer) with exactly 2^n binary neurons [Lin et al., 2017].
- Networks with binary weights require exactly the same number (2^n) neurons for approximating multivariate monomials too. Hence, in the context of this measure of expressivity- only binary weights are required to approximate a monomial, as good as infinitely-precise weights!
- This also can be extended to deeper networks with a constant factor of complexity, valid if conjecture 5.2 [Lin et al.] holds - Elaborated in Poster!

Generalized Binary Representation

- Why binarize to {-1, 1} or {0, 1}?
- Arbitrary two values- **α**, **β** forming a binary representation
- Binarized weight vector is of form [**ααββα…βαβ**]
- Or **α***(e)+ **β***(1-e) where e is the selection vector of the form [11001…010]
- How do we calculate optimal **α**, **β** and e?

Finding optimal **α, β** and e

- Weight vector **W**, binarize to form [**ααββα…βαβ**].
- Formulate it as an optimization problem- $\widetilde{\mathbf{W}}^* = \underset{\widetilde{W}}{argmin} \mid\mid \mathbf{W} - \widetilde{\mathbf{W}} \mid\mid^2$
- Here, **α**, **β** are values, and **e** $\in \{0,1\}^n \ni$ **e** $\neq$ **0** and **e** $\neq$ **1**.
- K = $\mathbf{e}^T\mathbf{e}$, denoting the number of 1s in e.

$$\widetilde{\mathbf{W}}^* = \alpha\mathbf{e} + \beta(\mathbf{1} - \mathbf{e}) \; where$$

$$\alpha = \frac{\mathbf{W}^T\mathbf{e}}{K}, \; \beta = \frac{\mathbf{W}^T(\mathbf{1} - \mathbf{e})}{n - K}$$

- To find the optimal e, check error for all possible K:

$$\mathbf{e}^* = \underset{e}{argmax}\left(\frac{\mid\mid \mathbf{W}^T\mathbf{e} \mid\mid^2}{K} + \frac{\mid\mid \mathbf{W}^T(\mathbf{1} - \mathbf{e}) \mid\mid^2}{n - K}\right)$$

Finding optimal K

- DP algorithm
- Top ‘K’ or Bottom ‘K’ values are significant
- Check for each K iteratively
- Reuse past computations
- O(**n.logn**) due to sort

Algorithm 1 Finding an optimal K value.

1: `Initialization`
2: $\mathbf{W}$ = 1D weight vector
3: T = Sum of all the elements of $\mathbf{W}$
4: Sort($\mathbf{W}$)
5: D = [00...0] // Empty array of same size as $\mathbf{W}$
6: $optK_1 = 0$ // Optimal value for K
7: $maxD_1 = 0$ // Value of D for optimal K value
8:
9: **for** I= 1 to D.size **do**
10: $P_i = P_{i-1} + \mathbf{W}_i$
11: $D_i = \frac{P_i^2}{i} + \frac{(T-P_i)^2}{n-i}$
12: **if** $D_i \geq maxD_1$ **then**
13: $maxD_1 = D_i$
14: $optK_1$ = i
15:
16: Sort($\mathbf{W}$, reverse=true) and **Repeat** steps 4-13 with $optK_2$ and $maxD_2$
17:
18: $optK_{final} = optK_1$
19: **if** $maxD_2 > maxD_1$ **then**
20: $optK_{final} = optK_2$
21:
22: **return** $optK_{final}$

Datasets & Models

- TU-Berlin - The most popular sketch dataset consisting of 20,000 sketches distributed over 250 classes
- Sketchy - The most popular SBIR dataset consisting of 75,471 sketches distributed over 125 classes

- Sketch-A-Net - A widely known alexnet-like network designed for sketch recognition task.
- ResNet-18 & GoogleNet - Popular compact architectures widely used to benchmark performance of binarization algorithms.

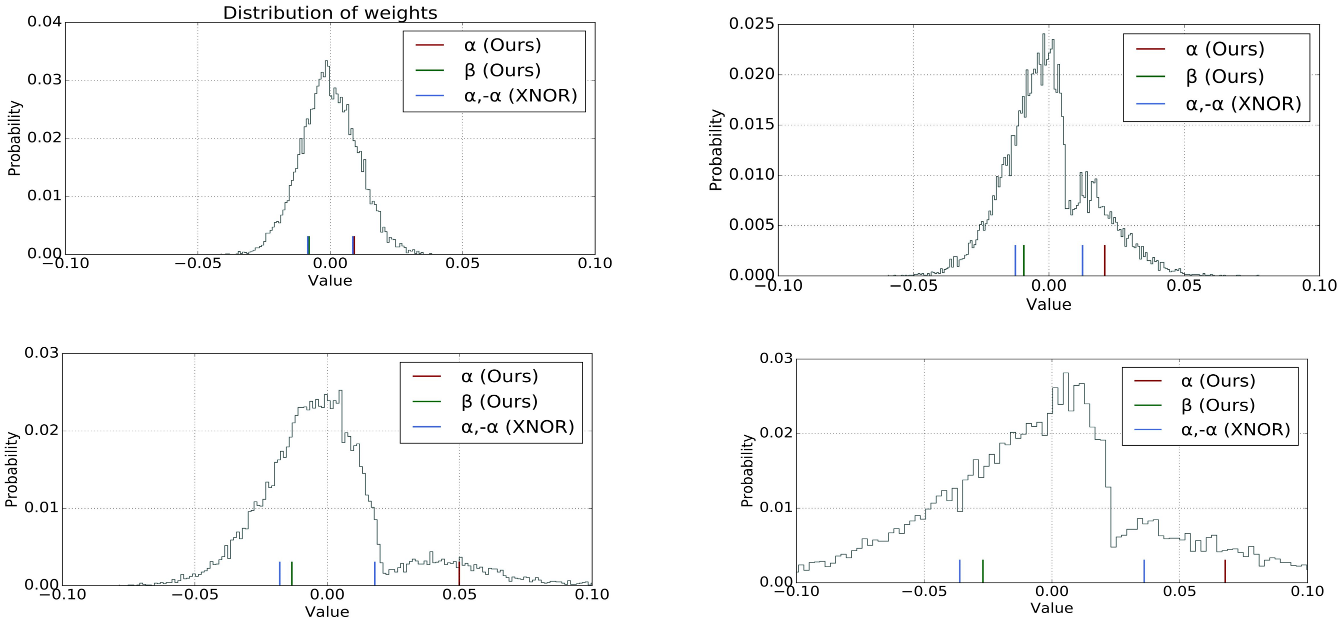
Experiments on Networks

- Our DAB-Nets outperform XNOR-Nets by significant amounts on both the datasets as shown in the table
- On Sketch-A-Net we observe a 0.8% improvement and a 2% improvement on TU-Berlin and Sketchy respectively.
- On ResNet-18 we observe a 2.5% and a 1.4% improvement
- On GoogleNet we observe a 1.5% and a 0.6% improvement

Models	**Method**	**Accuracies**	
		TU-Berlin	Sketchy
Sketch-A-Net	FPrec	72.9%	85.9%
	WBin (BWN)	73.0%	85.6%
	FBin (XNOR-Net)	59.6%	68.6%
	WBin DAB-Net	72.4%	84%
	FBin DAB-Net	60.4%	70.6%
Improvement	XNOR-Net s DAB-Net	+0.8%	+2.0%
ResNet-18	FPrec	74.1%	88.7%
	WBin (BWN)	73.4%	89.3%
	FBin (XNOR-Net)	68.8%	82.8%
	WBin DAB-Net	73.5%	88.8%
	FBin DAB-Net	71.3%	84.2%
Improvement	XNOR-Net s DAB-Net	+2.5%	+1.4%
GoogleNet	FPrec	75.0%	90.0%
	WBin (BWN)	74.8%	89.8%
	FBin (XNOR-Net)	72.2%	86.8%
	WBin DAB-Net	75.7%	90.1%
	FBin DAB-Net	73.7%	87.4%
Improvement	XNOR-Net s DAB-Net	+1.5%	+0.6%

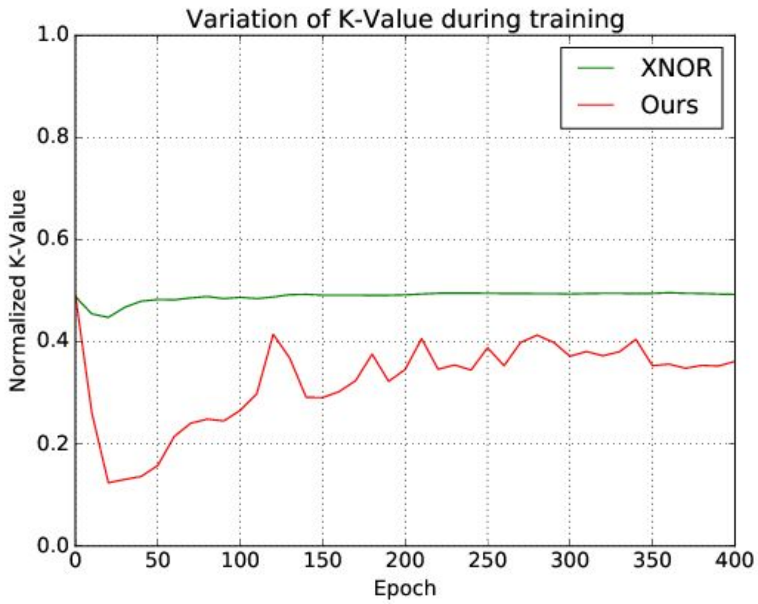
Observations - 1

- Variation of **α** and **β** across a filter's weights during training



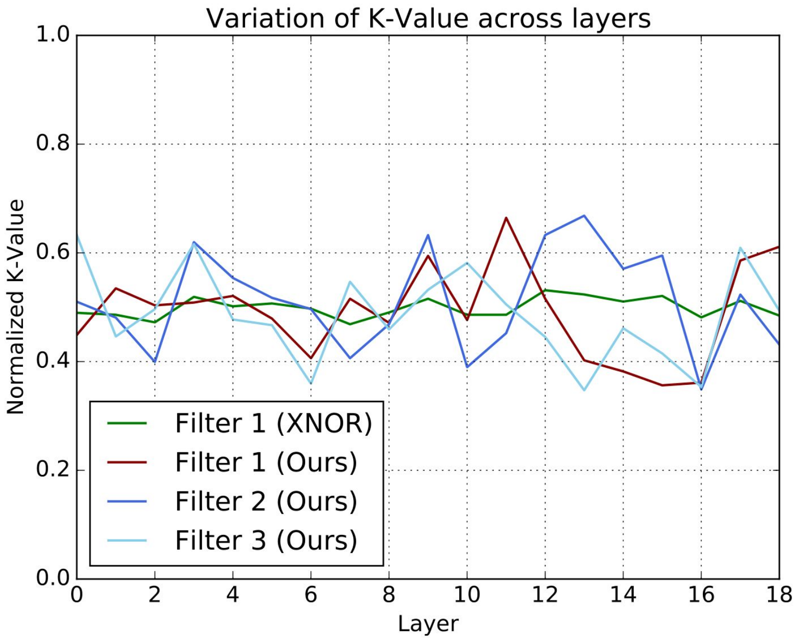
Observations - 2

- Variation of K for a filter during training



Observations - 3

- Variation of K for a filter across layers



Conclusions and take-aways

- Binary networks might be as expressible as infinite-precision networks!
- We propose a general binary approximation layer, with efficient algorithms for forward and backward pass.
- DAB-Nets can represent the space,capturing the distribution of data effectively.
- We hope that this project encourages more investigations into working with binary networks for all the cool applications presented at WACV ‘18!
- Our codes are available online! Links are given in our paper.

Thank You!

(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