True Gradient-Based Training of Deep Binary Activated Neural Networks via Continuous Binarization

Charbel Sakr[*#], *Jungwook Choi*[+], *Zhuo Wang*[+], *Kailash Gopalakrishnan*[+], *Naresh Shanbhag*[*]

[*] University of Illinois at Urbana-Champaign

[+] IBM T.J. Watson Research Center

[#] work done at IBM

Acknowledgment:

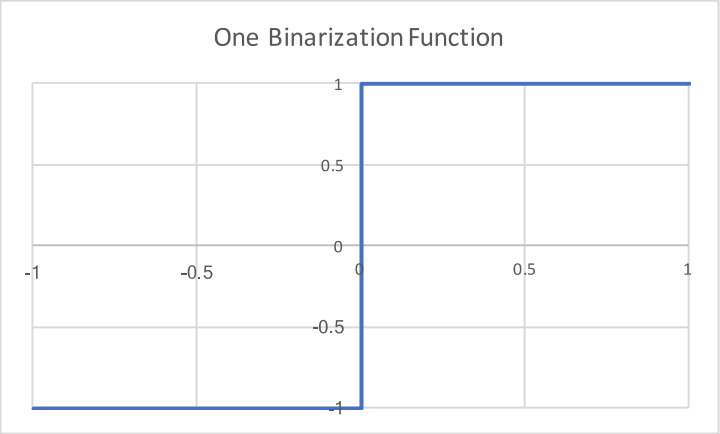
- This work was supported in part by Systems on Nanoscale Information fabriCs (SONIC), one of the six SRC STARnet Centers, sponsored by MARCO and DARPA.
- This work is supported in part by IBM-ILLINOIS Center for Cognitive Computing Systems Research (C3SR) - a research collaboration as part of the IBM AI Horizons Network.

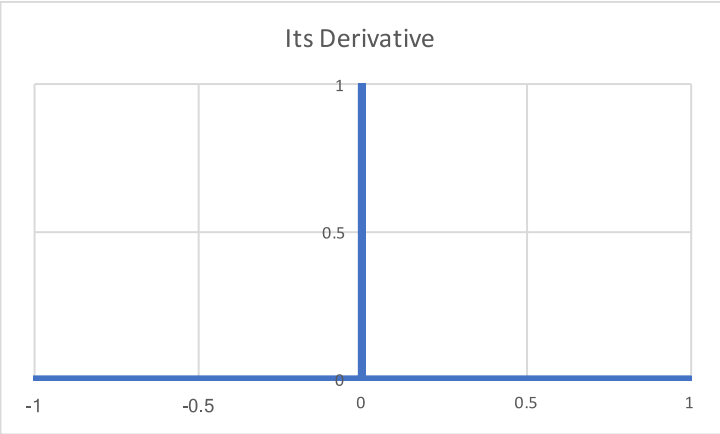
ILLINOIS
Electrical & Computer Engineering
COLLEGE OF ENGINEERING

IBM

The Binarization Problem

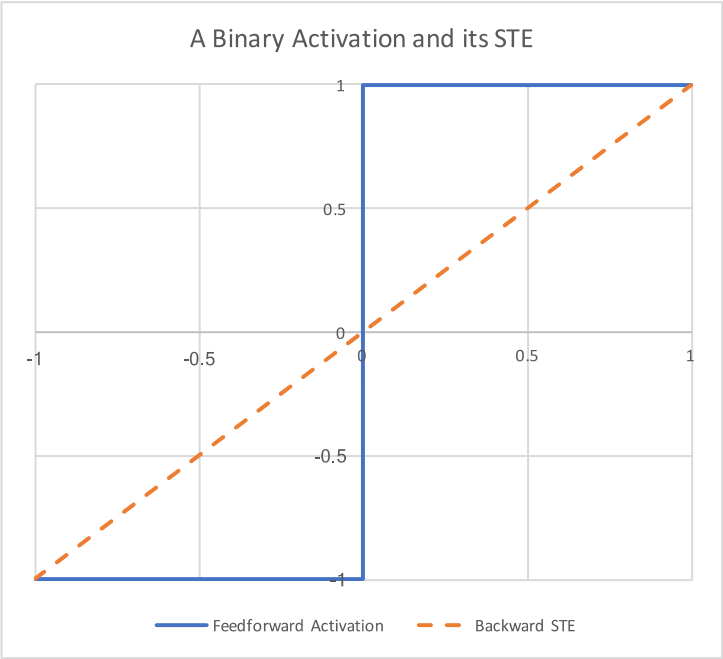
- Binarization of Neural Networks is an a promising direction to complexity reduction
- Binary activation functions are unfortunately non-continuous
- Training networks with binary activations cannot use gradient-based learning





Current Approach

- Treat binary activations as stochastic units (Bengio, 2013).
- Use a straight through estimator (STE) of the gradient.
- Was shown to enable training of binary networks (Hubara et al., Rastegari et al., etc…).
- Often comes at the cost of accuracy loss compared to floating point.

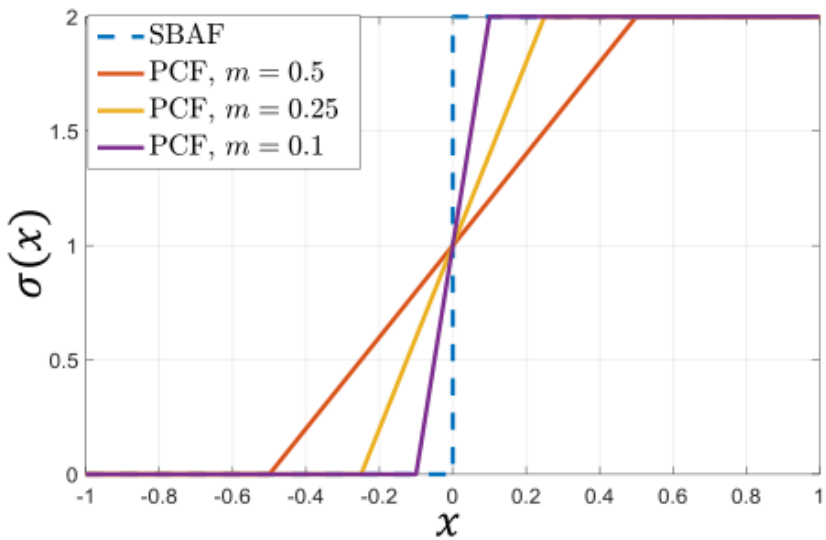


Proposed Method

- Start with a *clipping* activation function and *learn* it to become binary.

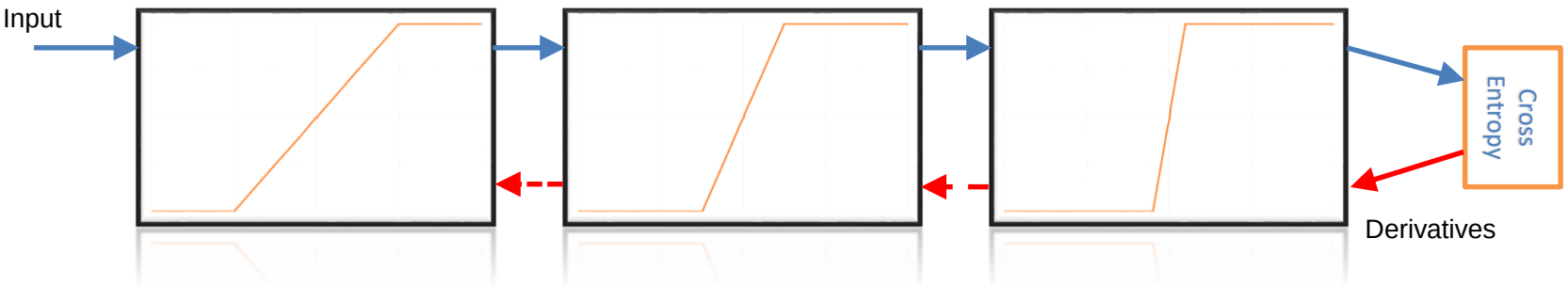
$$actFn(x) = Clip\left(\frac{x}{m} + \frac{\alpha}{2}, 0, \alpha\right)$$

- Smaller m means steeper slope. The activation function naturally approaches a binarization function.



Caveat: Bottleneck Effect

- Activations cannot be learned simultaneously due to the *bottleneck effect* in backward computations



- Hence we learn slopes *one layer at a time*.

Justification via induction

Step 1: The base case, a baseline network with clipping activation function

The arrows mean layer-wise operation of the input

Justification via induction

Step 1: The base case, a baseline network with clipping activation function

Step 2: Replace first layer activation with binarization and stop learning first layer

The arrows mean layer-wise operation of the input

Justification via induction

Step 1: The base case, a baseline network with clipping activation function

Step 2: Replace first layer activation with binarization and stop learning first layer

Equivalently, we have a new network with binary inputs

The arrows mean layer-wise operation of the input

Justification via induction

Step 1: The base case, a baseline network with clipping activation function

Step 2: Replace first layer activation with binarization and stop learning first layer

Step $L-1$: Binary inputs – Even shorter network

The arrows mean layer-wise operation of the input

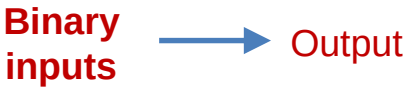
Justification via induction

Step 1: The base case, a baseline network with clipping activation function

Step 2: Replace first layer activation with binarization and stop learning first layer

Step $L-1$: Binary inputs – Even shorter network

Step L: Binary inputs – Very short network



The arrows mean layer-wise operation of the input

Justification via induction

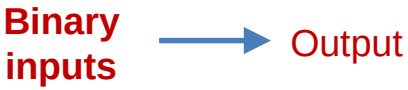
Further analysis required!

Step 1: The base case, a baseline network with clipping activation function

Step 2: Replace first layer activation with binarization and stop learning first layer

Step $L-1$: Binary inputs – Even shorter network

Step L: Binary inputs – Very short network



The arrows mean layer-wise operation of the input

Analysis

- The mean squared error of approximating the PCF by and SBAF decreases linearly in m

$$MSE = k \int_{-\frac{m\alpha}{2}}^{-\frac{m\alpha}{2}} \left(\frac{x}{m} + \frac{\alpha}{2} - \alpha \cdot \mathbb{1}_{x>0} \right)^2 dx = c \cdot m$$

Hence, perturbation magnitude decreases with m

PCF: Parametric Clipping Function
SBAF: Scaled Binary Activation Funciton

Analysis

- The mean squared error of approximating the PCF by and SBAF decreases linearly in m

$$MSE = k \int_{-\frac{m\alpha}{2}}^{-\frac{m\alpha}{2}} \left(\frac{x}{m} + \frac{\alpha}{2} - \alpha \cdot \mathbb{1}_{x>0} \right)^2 dx = c \cdot m$$

Hence, perturbation magnitude decreases with m

- There is no mismatch when using the SBAF or the PCF provided a bounded perturbation magnitude

$$\|\mathbf{q_a}\| < \min_{j=1\ldots M,\ j\neq i} \frac{f_i(\mathbf{a}_o) - f_j(\mathbf{a}_o)}{\|\nabla_{\mathbf{a}_o} f_j(\mathbf{a}_o) - \nabla_{\mathbf{a}_o} f_i(\mathbf{a}_o)\|}$$

Backtracking: small m → small perturbation → less mismatch

PCF: Parametric Clipping Function
SBAF: Scaled Binary Activation Funciton

Analysis

- The mean squared error of approximating the PCF by and SBAF decreases linearly in m

$$MSE = k \int_{-\frac{m\alpha}{2}}^{-\frac{m\alpha}{2}} \left(\frac{x}{m} + \frac{\alpha}{2} - \alpha \cdot \mathbb{1}_{x>0} \right)^2 dx = c \cdot m$$

Hence, perturbation magnitude decreases with m

- There is no mismatch when using the SBAF or the PCF provided a bounded perturbation magnitude

$$\|\mathbf{q_a}\| < \min_{j=1\ldots M,\ j\neq i} \frac{f_i(\mathbf{a}_o) - f_j(\mathbf{a}_o)}{\|\nabla_{\mathbf{a}_o} f_j(\mathbf{a}_o) - \nabla_{\mathbf{a}_o} f_i(\mathbf{a}_o)\|}$$

Backtracking: small m → small perturbation → less mismatch

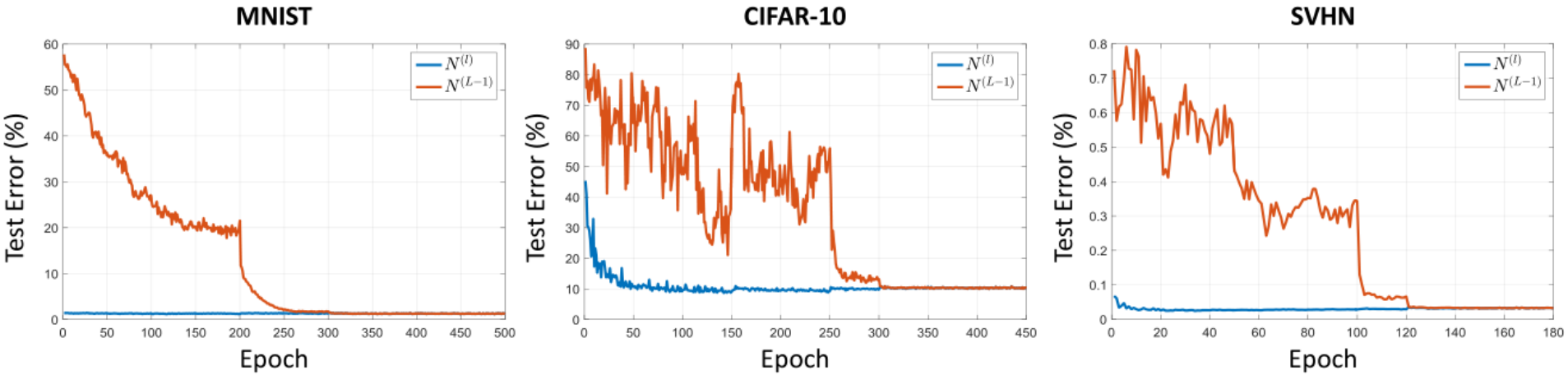
PCF: Parametric Clipping Function
SBAF: Scaled Binary Activation Funciton

How to make m small?

Regularization

- We add a regularization term λ to m when learning it (L_2 and/or L_1).
- The optimal λ value is usually found by tuning (common problem with regularization).
- We have some empirical guidelines from our experiments:
  - Layer type 1: fully connected
  - Layer type 2: convolution preceding convolution
  - Layer type 3: convolution preceding pooling
  - We have observed that the following is a good strategy
    - $\lambda_1 > \lambda_2 > \lambda_3$

Convergence



- Blue curve: obtained network by binarizing up to layer l
- Orange curve: completely binary activated network
- As training evolves, the network becomes completely binary and the two curves meet
- The accuracy is very close to the initial one which is the baseline

Comparison with STE

	MNIST	CIFAR-10	SVHN
Full-precision Baseline	1.45%	9.04%	2.53%
Binarization via STE	1.54%	14.80%	4.05%
Continuous Binarization	1.27%	10.41%	3.20%

summary of test errors

- Our method consistently outperforms binarization via STE

Conclusion & Future Work

- We presented a novel method for binarizing the activations of deep neural networks
  → The method leverages true gradient based learning
  Consequently, the obtained results consistently outperform conventional binarization via STE

- Future work
  → Experimentations on larger datasets
  → Combining the proposed activation binarization to weight binarization
  → Extension to multi-bit activations

Thank you!