Efficient Layout Hotspot Detection via Binarized Residual Neural Network

Yiyang Jiang[1], Fan Yang[1*], Hengliang Zhu[1], Bei Yu[3], Dian Zhou[2], Xuan Zeng[1*]

[1]State Key Lab of ASIC & System, Microelectronics Department, Fudan University
[2]University of Texas at Dallas
[3]Chinese University of Hong Kong

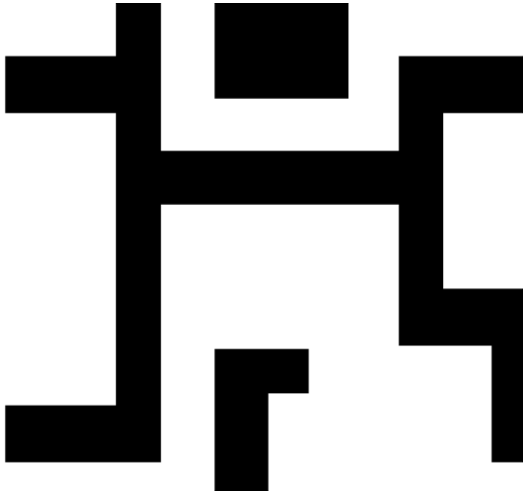
Outline

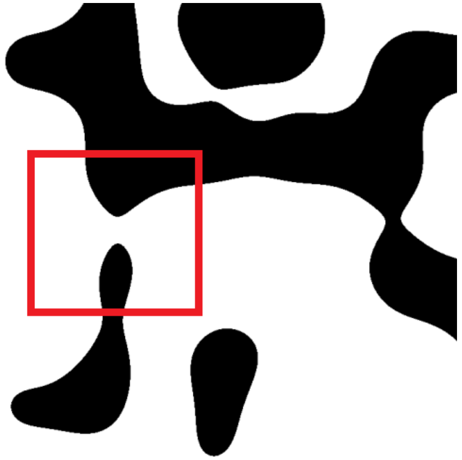
- Introduction
- Proposed Binarized Neural Network-based Hotspot Detector
- Experimental Results

Outline

- Introduction
- Proposed Binarized Neural Network-based Hotspot Detector
- Experimental Results

Lithography Proximity Effect





- What you see ≠ what you get
- RETs: OPC, SRAF, MPL
- Still exists hotspots: low fidelity patterns
- Lithography simulation: time consuming

Hotspot Detection Problem

Definition: Accuracy

The ratio of correctly predicted hotspots among the set of actual hotspots.

$$Accuracy = \frac{\#TP}{\#TP + \#FN}$$

Definition: False Alarm

The number of incorrectly predicted non-hotspots.

$$False\ Alarm = \#FP$$

Problem: Hotspot Detection

Given a dataset that contains hotspot and non-hotspot instances, train a classifier that can maximize the $accuracy$ and minimize the $false\ alarm$.

Hotspot Detection Methods

Two Classes:

- Pattern matching-based
- Machine learning-based

Pattern Matching-based Hotspot Detection

- Characterize the hotspots as explicit patterns and identify the hotspots by matching these patterns

- [Yu+,ICCAD’14] [Nosato+,JM3’14] [Kahng+,SPIE’06] [Su+,TCAD’15] [Wen+,TCAD’14] [Yang+,TCAD’17]

- Fast but hard to detect unseen patterns

Machine Learning-based Hotspot Detection

- Build implicit models by learning from existing training data
  - SVM, Bayesian, Decision-tree, Boosting, NN, ...

- [Ding+,ASPDAC'11] [Yu+,DAC'13] [Matsunawa+,SPIE'15] [Zhang+,ICCAD'16] [Wen+,TCAD'14]

- Possible to detect the unseen hotspots but may cause false alarm issues

Deep Learning-based Hotspot Detection

- Belongs to ML-based hotspot detection but different from conventional ML models:
  - Feature Crafting v.s. Feature Learning
  - Stronger scalability

- [Yang+,DAC'17]

- Drawback: not storage and computational efficient

Outline

- Introduction
- Proposed Binarized Neural Network-based Hotspot Detector
- Experimental Results

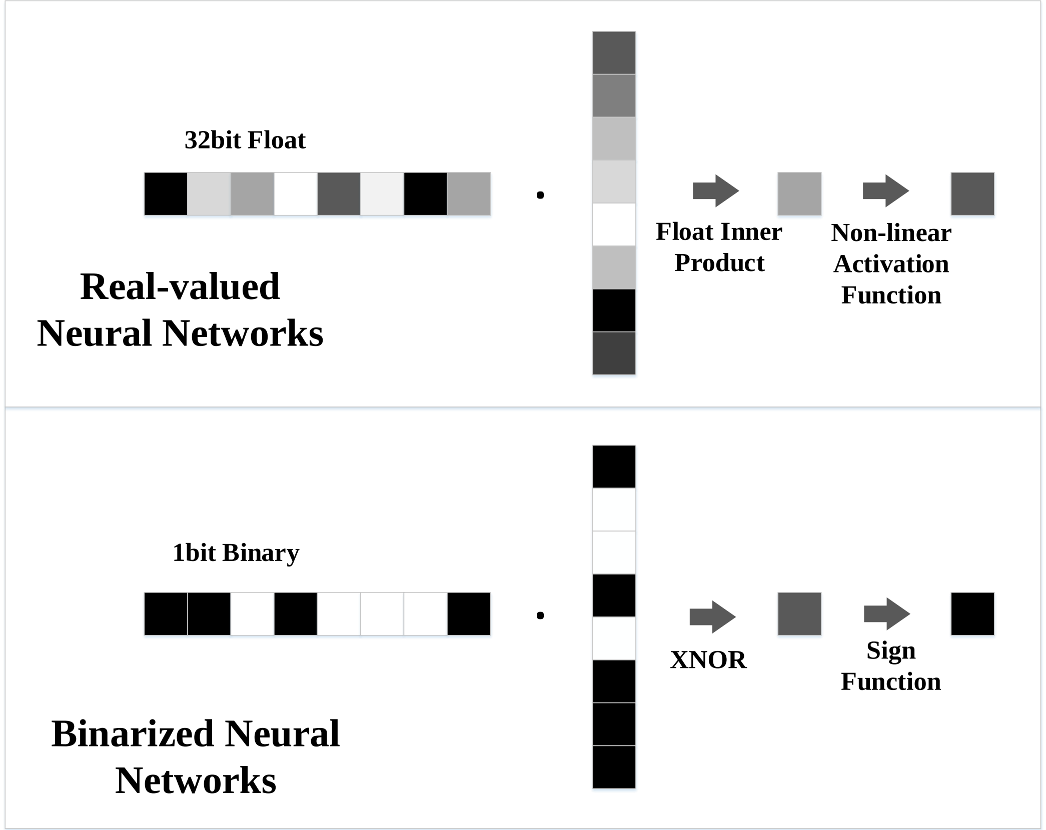
Parameter Quantization

- Problem with deep neural networks:
  - Enormous computational and storage consumption

- To alleviate this problem:
  - Parameter Quantization
  - 32-bit floating-point weights not necessary: quantized to fixed-point of 8-bit, 3-bit, 1-bit…
  - [Arora+,ICML'14] [Hwang+,SiPS'14] [Soudry+,ANIPS'14] [Rastegari+,ECCV'16]

Binarized Neural Network

- Binarized neural network (BNN):
  - Extremely quantized to 1 bit
  - Inherently suitable for hardware implementation
- Layout patterns are binary images
  - BNN might be suitable for that



Binarization Approach

Definition

Let W be the kernel which is an n-element vector and X be the vector of the corresponding block in the input tensor, $n = w_k \times h_k$. Let W_B, X_B be the binarized kernel and input vector and α_W, α_X be the corresponding scaling factors. Here $W, X \in \mathbb{R}^n$, $W_B, X_B \in \{-1, +1\}^n$ and $\alpha_W, \alpha_X \in \mathbb{R}^+$.

Problem: Binarization

Given the kernel and input vector W, X, find best $W_B, X_B, \alpha_W, \alpha_X$ that minimizes the binarization loss L_i. $L_i(W_B, X_B, \alpha_W, \alpha_X) = \|W \odot X - \alpha_W W_B \odot \alpha_X X_B\|^2$ where $\odot$ means inner product.

Binarization Approach

- Solving the minimization problem:

$$W_B^* = sign(W),\ X_B^* = sign(X)$$

$$\alpha_W^* = \frac{1}{n}\|W\|_{l1},\quad \alpha_X^* = \frac{1}{n}\|X\|_{l1}$$

- The estimated weight and corresponding input vector $\widetilde{W}, \tilde{X}$ are:

$$\widetilde{W} = \frac{1}{n} sign(W)\|W\|_{l1}$$

$$\tilde{X} = \frac{1}{n} sign(X)\|X\|_{l1}$$

Training BNN

- Gradient for $sign$ function [Hubara, 2016]

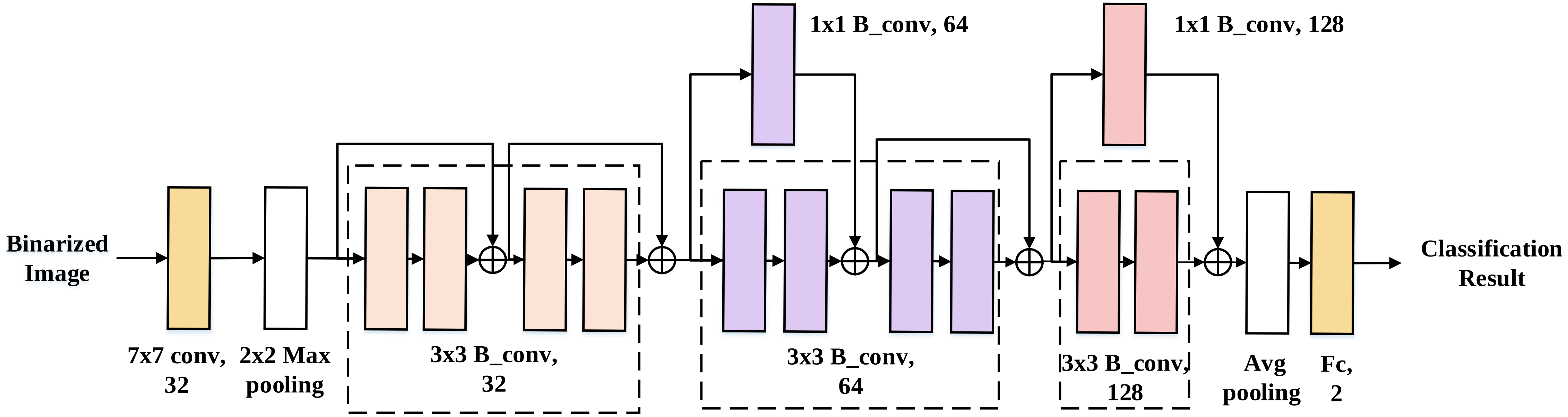
$$\frac{\partial sign(x)}{\partial x} = \mathbf{1}_{\|W\|<1}$$

- Back propagation through the Binarizing Layer

$$\begin{aligned}
\frac{\partial l}{\partial W} &= \frac{\partial l}{\partial \widetilde{W}} \frac{\partial \widetilde{W}}{\partial W} \\
&= \frac{\partial l}{\partial \widetilde{W}} \frac{\partial (\frac{1}{n}\|W\|_{l1} sign(\mathrm{W}))}{\partial W} \\
&= \frac{\partial l}{\partial \widetilde{W}} (\frac{1}{n} + \alpha_W^* \mathbf{1}_{\|W\|<1})
\end{aligned}$$

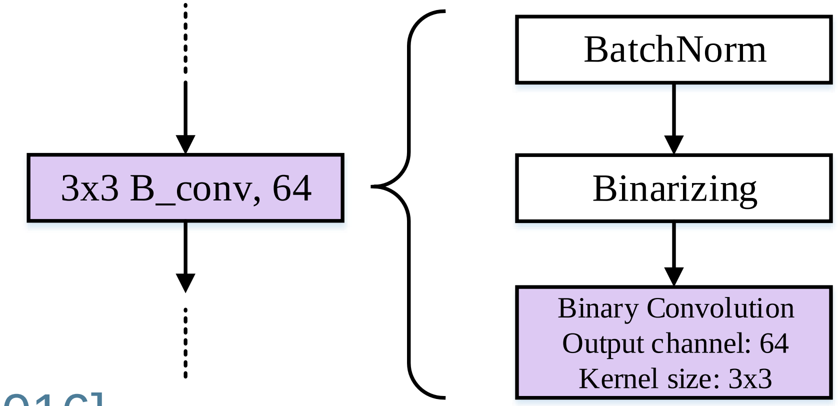
Network Architecture

- Information loss caused by binarization: need a stronger network
- Residual block-based architecture

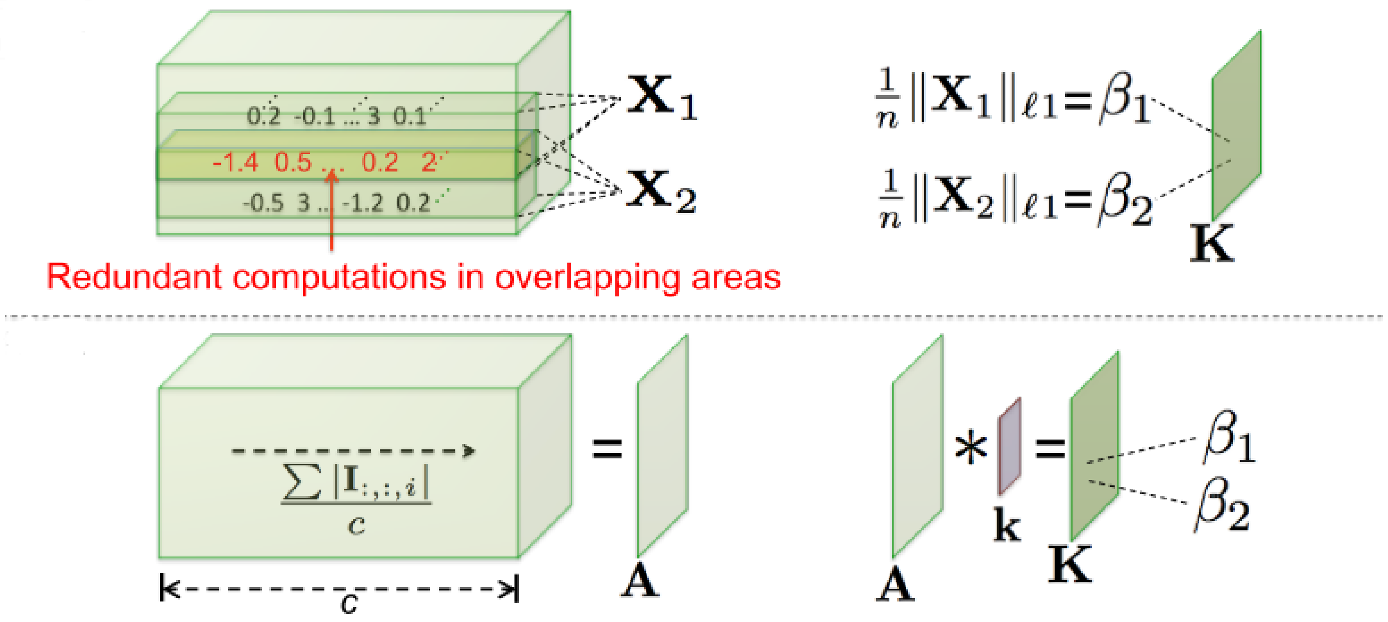


Implementation Details

- Typical BNN block structure



- Speedup scaling factor calculation [Rastegari, 2016]



Implementation Details

- Biased Learning [Yang, 2017]
  - Loss function: Softmax cross entropy
  - Trained with hotspot's label $y_h^* = [0,1]$ and non-hotspot's label $y_n^* = [1, 0]$
  - Trained model is fine-tuned with non-hotspot's label changed to $y_n^* = [1 - \epsilon, \epsilon]$ and hotspot's label keeps the same. ϵ is set to 0.2.
- Data preprocessing
  - Down-sampled to 128×128
- Training hyperparameters
  - Batch size:128
  - Learning rate: Initial 0.15, exponentially decay each time loss plateaus
  - Optimizer: NAdam optimizer [Dozat, 2016]
  - Initializer: Xavier initializer [Glorot, 2010]

Outline

- Introduction
- Proposed Binarized Neural Network-based Hotspot Detector
- Experimental Results

Performance Comparisons with Previous Hotspot Detectors

- Benchmark: ICCAD 2012 Contest

Method	Accuracy (%)	False Alarm #	Runtime (s)
SPIE’15	84.2	2919	2672
ICCAD’16	97.7	4497	1052
DAC’17	98.2	3413	482
Ours	**99.2**	**2787**	**60**

- Accuracy improved from 84.2% to 99.2%
- Fewest False Alarms: 2787
- Lowest Runtime: 60s, 8x faster

Thank You