Weight Isolation-based Binarized Neural Networks Accelerator

Zhangkong Xian, Hongge Li*, Yuliang Li,
School of Electronic Information Engineering
Beihang University, Beijing, China

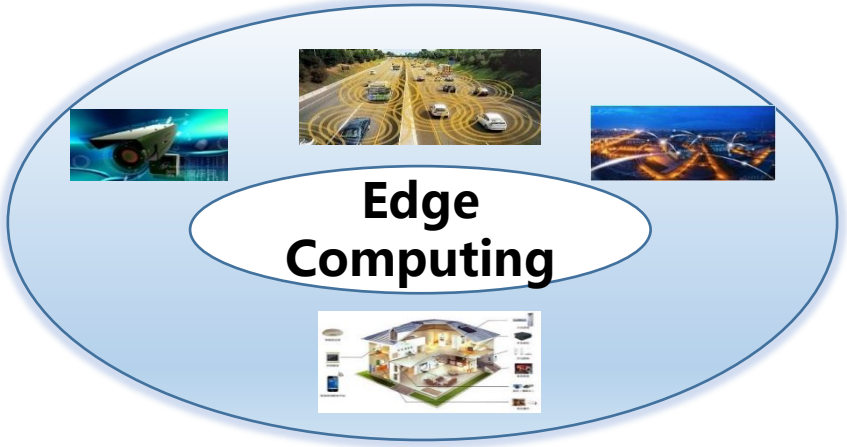
Outline

1. Background

- Convolutional neural networks （CNN）have demonstrated state-of-the-art results in many fields, such as image classification，object detection，and even artificial intelligence for games against human beings.
- CNN usually need large amounts of memory and computing power.

Networks	AlexNet	VGG16	VGG19	ResNet152
Number of convolutional layer operations	666 M	15.3 G	19.5 G	11.3 G
Number of convolutional layer parameters	2.33 M	14.7 M	20 M	58 M
Number of full connected layer operations	58.6 M	124 M	124 M	2.05 M
Number of full connected layer parameters	58.6 M	124 M	124 M	2.05 M
Total number of operations	724 M	15.5 G	19.6 G	11.3 G
Total number of parameters	61 M	138 M	144 M	60 M

1. Background



Resource efficiency

Energy efficiency

Power Consumption

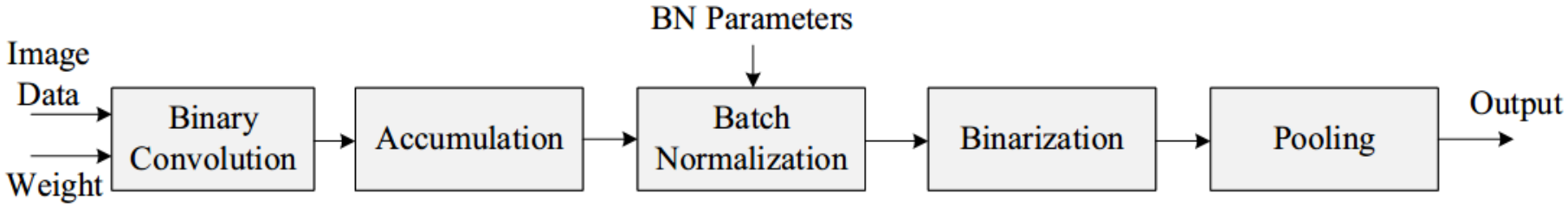
.....

- **Model compression and hardware implementation methods :**
  - Binarized Neural Network (BNN) based ASIC/FPGA
- **Some shortcoming of conventional BNN with (-1, +1) :**
  - Conventional BNN methods need to be more hardware-friendly
  - BNN accelerators need to be more energy efficiency
  - Existing hardware does not take full advantage of BNN

Outline

2.1 Binarization Method

- **BNN Convolutional Layer Operations :**



- **Traditional Neural Network Binarization Methods**[Courbariaux 2016]
  - $+1:\ x \geq 0;\ -1:\ x < 0$
  - Using bit operation instead of multiplier
  - Using counter instead of adder
- **Some shortcoming of conventional BNN with (-1, +1) :**
  - Requires subtraction in BNN circuit
  - If counter is used to sum, additional operation are required
  - Resource efficiency, Power Consumption

2.1 Binarization Method

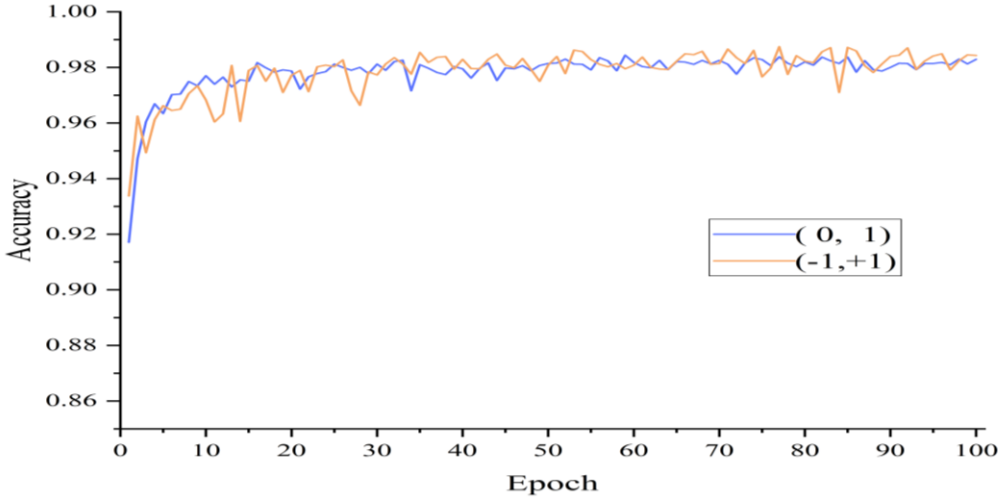
- **Binarization method proposed**

$$\mathbf{Binarize(x)} = \begin{cases} \mathbf{1} & \boldsymbol{x} > \mathbf{0} \\ \mathbf{0} & \boldsymbol{x} \leq \mathbf{0} \end{cases}$$

  - The data quantized as (0,1) is used in the circuit
  - Subtraction step is not required
  - Using bit operation and pop-counter

- **The precision of the two methods is simulated in the same condition.**

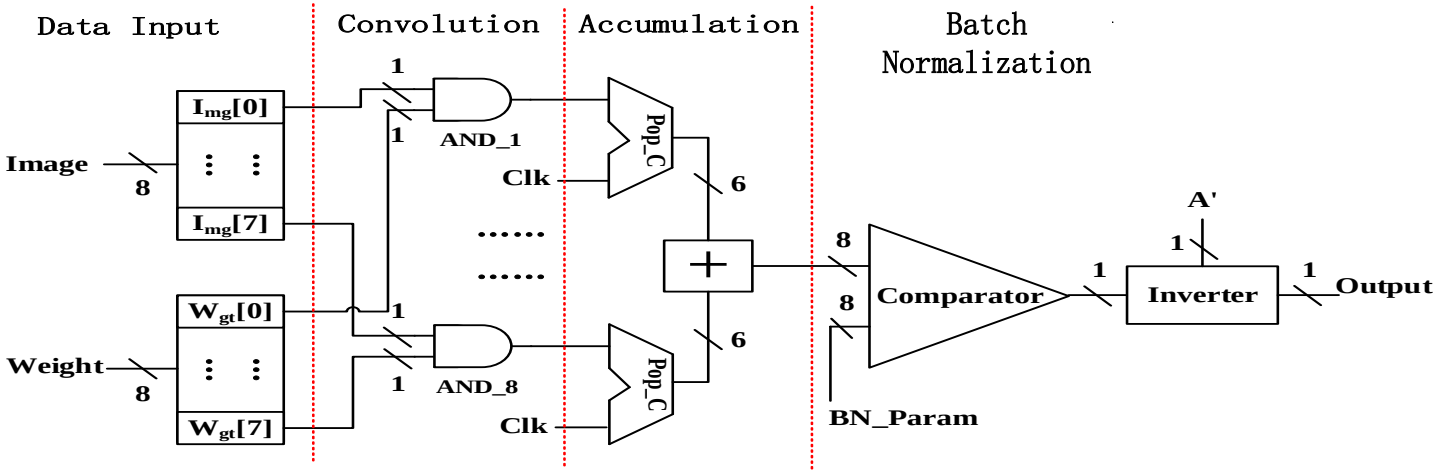
Dataset	MNIST
Formwork	Pytorch
Network	Lenet-5
Epoch	100



Outline

2.2 Computing Core

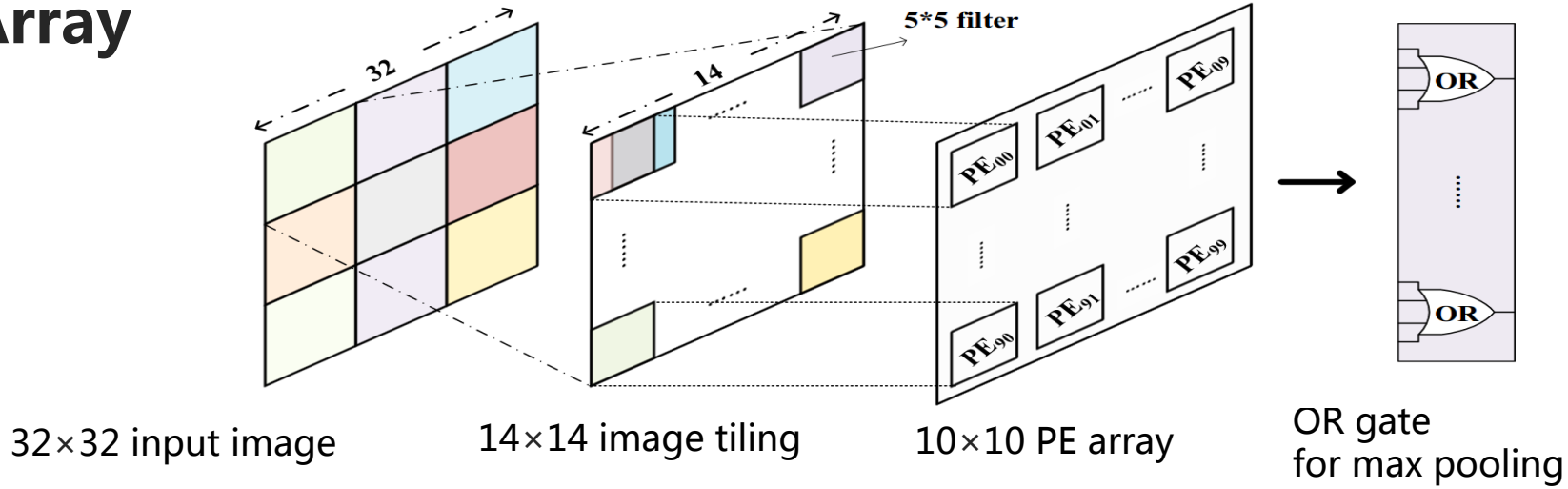
- **Process Element (PE)**



- To keep the data bit widths consistent with the 8 bits of the input layer, each computing unit uses 8 parallel AND gates
- Use AND gate instead of multiplier for the convolution operation
- Use pop-counter to sum the parts and then add up the final convolution
- BN layer include the comparator and inverter

2.2 Computing Core

- **PE Array**



32×32 input image　　14×14 image tiling　　10×10 PE array　　OR gate for max pooling

- **Image tiling**

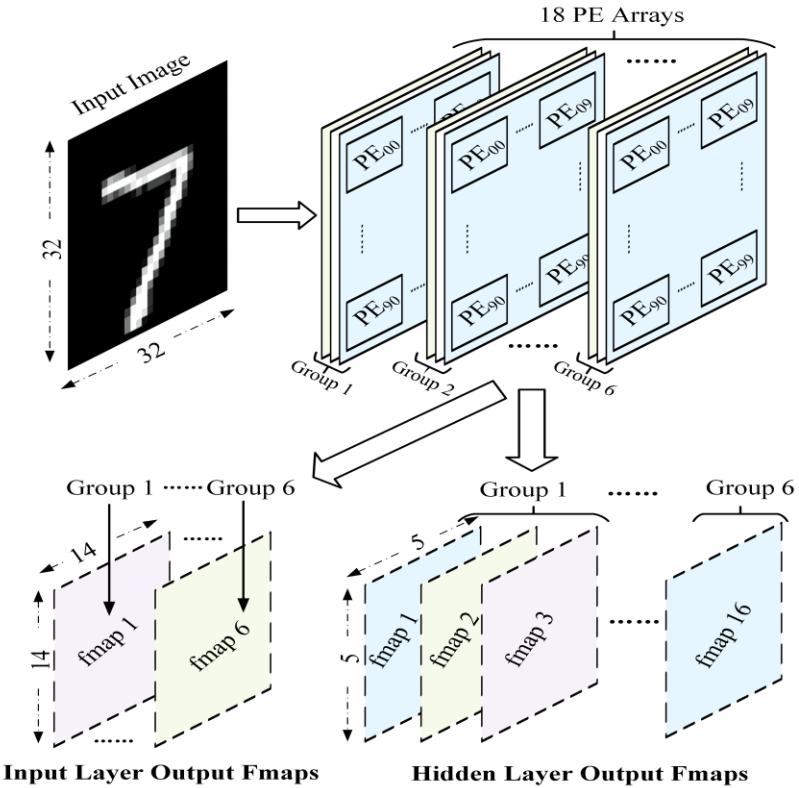
  32×32 input image is cut into (14×14) ×9 of sub-images to avoid large array design and reduce resource consumption.

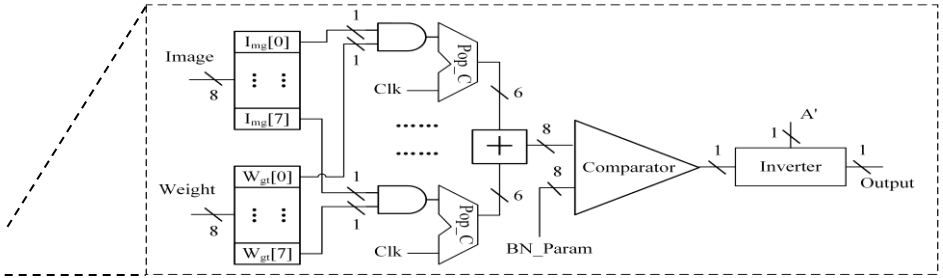
- **10×10 PE array**

  Calculate 14×14 image at one time, 9 cycles to complete a 32×32 image calculation.

2.2 Computing Core

- **Computing arrays :**



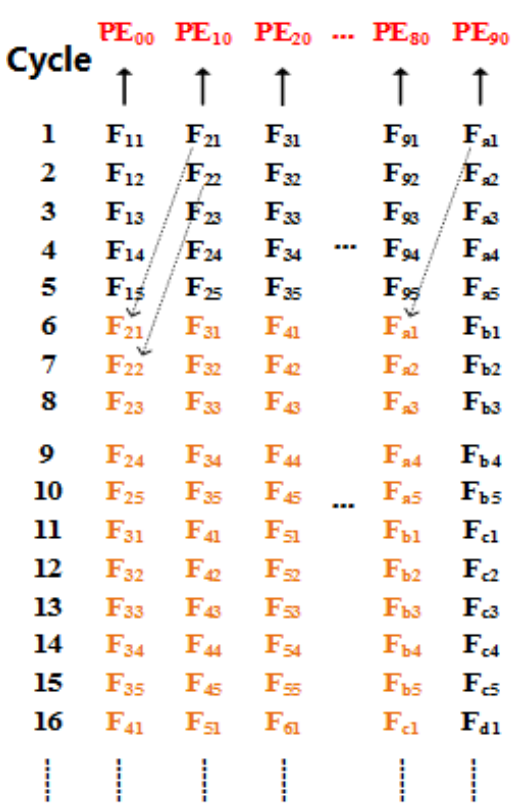


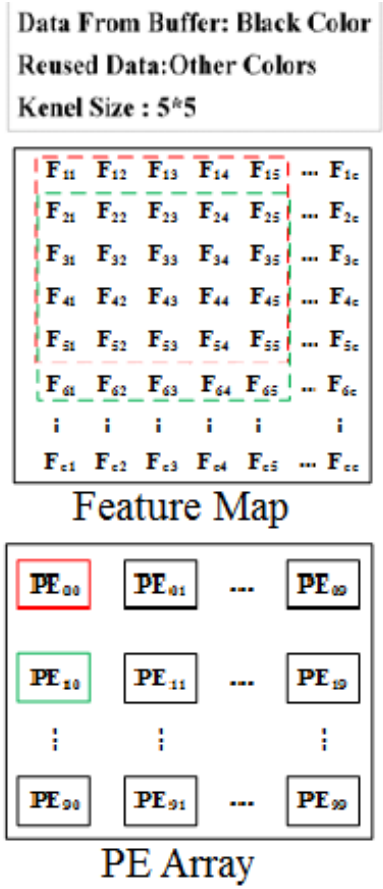
- **18 reconfigurable computing arrays**
- **Input layer** : The number of output features is 6, assign 3 arrays to calculate an output feature map.
- **Hidden layer** : The number of output features is 16, one output feature map is calculated for each array.

Outline

2.3 Data reuse

- **Input Layer**
  - Each feature map data is 8 bits
  - Each splicing module is used for one line of PE data transfer
  - FIFO is used for temporary storage of reuse data
  - For 5x5 kernel, data reuse rate :
  - $1 - \frac{(25+5*9)*10}{100*25} * 100\% = \mathbf{72}\%$





2.3 Data reuse

- **Hidden Layer**
  - Each feature map data is 1 bit
  - Each splicing module is used for one line of PE data transfer
  - No FIFO is used for temporary data storage
  - For 5x5 kernel, data reuse rate :

$$1 - \frac{(25+8\times9+7)\times10}{100\times25} \times 100\% = \mathbf{58.4\%}$$

Algorithm 1: Data Reuse

Note:

Clk: Calculation cycle
Ct: Concatenation modules
Ct_num: the number of data concatenation modules
t: for input layer, t is the tile number of input image; for hidden layer, t is the number of input fmaps
k: for input layer, k = kenel size; for hidden layer, k = 1
j: for input layer, j = k*k; for hidden layer, $j = \lceil k * k/8 \rceil$

```
for loop = 1 to t do
   for Clk = 1 to j do
      if Clk <= k then
         for i = 1 to Ct_num do
            Ct_i read data from DRAM
         end for
      else if Clk > k do
         if i = Ct_num do
            Ct_i read data from DRAM
         else if i = Ct_num-1 do
            input  layer: Ct_i read data from DRAM
                          hidden layer: Ct_i read data from DRAM and Ct_i+1
            else do
            for i = 1 to Ct_num-2 do
                          input  layer: Ct_i read data from FIFO
                          hidden layer: Ct_i read data from Ct_i+1, Ct_i+2
                          end for
            end if
         end if
   end for
end for
```

Outline

2.4 Weight Isolation Logic

- **If the weight is 0, the transmission of the image data is invalid**

  After the data in the CNN is binarized to (0,1), when the weight is 0, the result of this operation is also 0, where the image data has no effect on the convolutional result

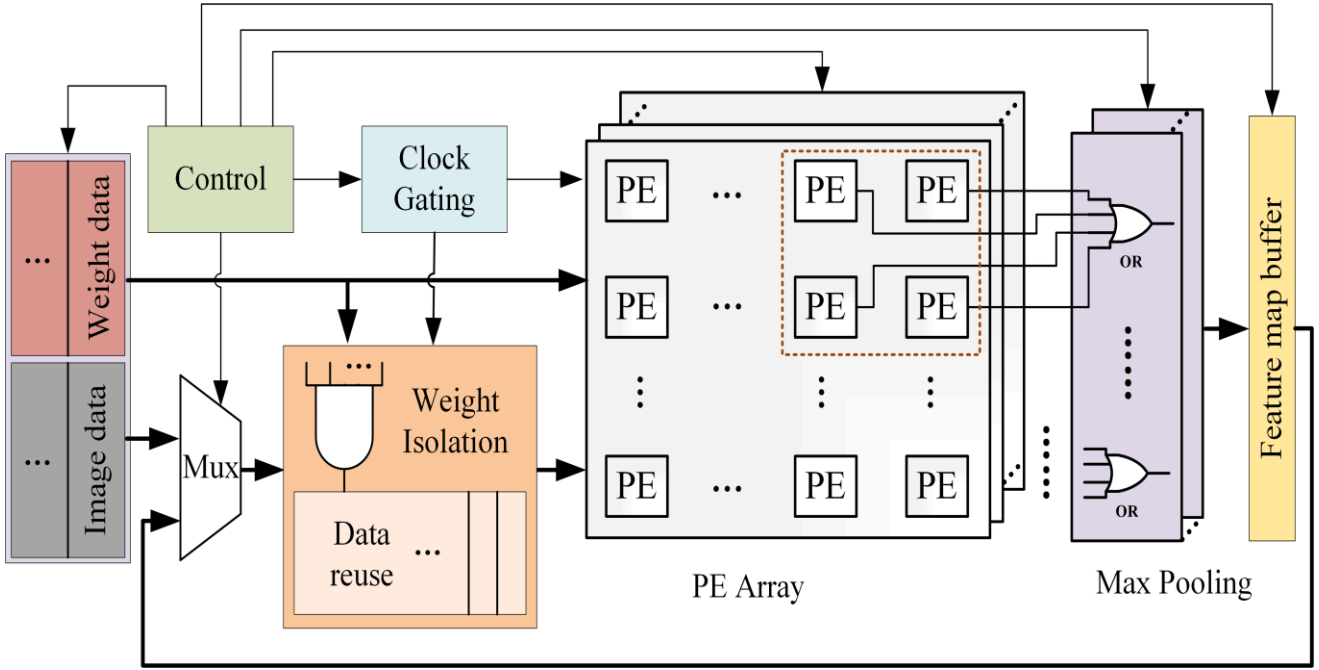
- **Percentage of weight 0 in each layer of the BNN**

Layers	Conv1	Conv2	FC1	FC2	FC3	Average
0-ratio	79.3%	81.1%	79.9%	81.2%	57.3%	75.8%

  - **With an average weight of 75.8% to 0, then, there are many invalid data transfers**

2.4 Weight Isolation Logic

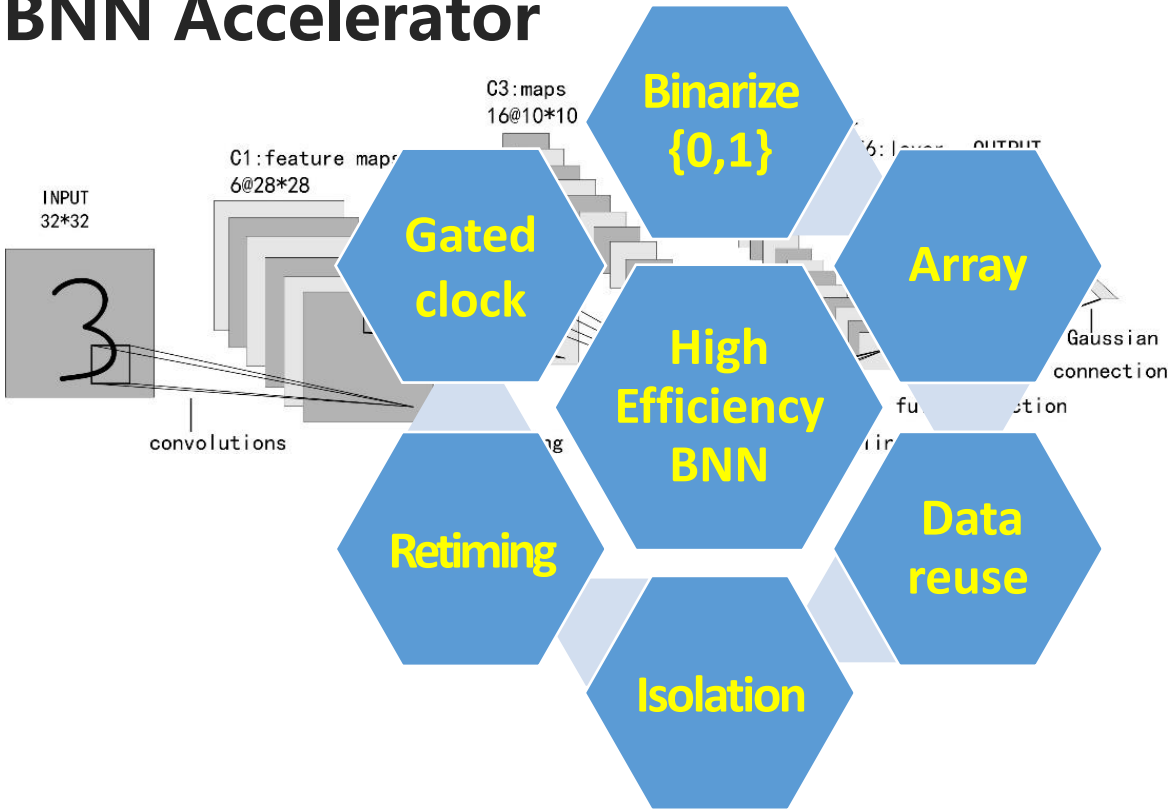
- The weights as control signal controls whether the image data to transmit.
- Reduced signal send rate to realize low power consumption



Outline

2.5 Hardware Implementation

- BNN Accelerator



Throughput≈3.4 TOPS@500MHz; Power=2.08W@FPGA

2.5 Hardware Implementation

- **COMPARISON OF PREVIOUS WORKS**

BNN	FPGA 2017	NC 2018	JCSC 2019	VLSI 2019	ISCAS 2020 This work
Platform	ZYNQ	Stratix-V	ZYNQ	Vertex-485T	**Vertex-690T**
Precision	1bit weight	1bit	1bit	1bit weight	**1bit**
Dataset	MNIST	MNIST	SVHN	PASCAL VOC	**MNIST**
Frequency (MHz)	200	150	-	200	**500**
DSP Used	-	384	-	272	**-**
BRAM Used (18Kb)	396	2210	103	1214	**23**
LUT Used	82988	-	29600	104700	**65413**
FF Used	-	-	-	140100	**54891**
Performance (GOPS)	9086	12219	2236	4420	**3378**
Power(W)	8.8	26.2	3.2	14.72	**2.08**
GOPS/KLUT	109.5	-	75.5	42.2	**51.7**
Efficiency (GOPS/W)	1033	466	699	300	**1624**

Outline

3. Summary

- Hardware sparse architecture based on the weight isolation is proposed for high efficiency binary neural network
- New algorithm of the data reuse to decrease the access to off-chip is proposed
- Weights and activations are constrained to either 0 or 1
- The accelerator realizes high throughput, 58.8% decrease of data access, and 1.624 TOPS/W high energy efficiency based on FPGA.

Thanks!

Q&A