1 Energy, Latency, Area in BNN Vs. CNN

Total Decrease of Energy/Latency/Power in VGG3 Architecture resulting from Binarizing CNN

Energy: $\frac{Energy_{BNN}-Energy_{CNN}}{Energy_{CNN}}$
$\rightarrow \frac{713803-943524}{943524} = -0.2434712842 \approx -24.343\%$

Latency: $\frac{LatencyBNN-Latency_{CNN}}{Latency_{CNN}}$
$\rightarrow \frac{0.000717605-1.6491568}{1.6491568} = -0.9995648 \approx -99.956\%$

Area: $\frac{Area_{BNN}-Area_{CNN}}{Area_{CNN}}$
$\rightarrow \frac{39063763456-5.18221947*10^{10}}{5.18221947*10^{10}} = -0.2461962731 \approx -24.619\%$

Total Decrease of Energy/Latency/Power in VGG7 Architecture resulting from Binarizing CNN

Energy: $\frac{Energy_{BNN}-Energy_{CNN}}{Energy_{CNN}}$
$\rightarrow \frac{19434565-37468115}{37468115} = -0.4813039033 \approx -48.130\%$

Latency: $\frac{LatencyBNN-Latency_{CNN}}{Latency_{CNN}}$
$\rightarrow \frac{0.019605149088-65.48329719}{65.48329719} = -0.99970 \approx -99.970\%$

Area: $\frac{Area_{BNN}-Area_{CNN}}{Area_{CNN}}$
$\rightarrow \frac{1061099905408-2.057844566*10^{12}}{2.057844566*10^{12}} = -0.484363434 \approx -48.436\%$

2 number of operations

Approximation of the number of Operations for Convolutional Neural Networks (CNN) VGG3 Architecture				
Operation	**Addition**	**Subtraction**	**Multiplication**	**Comparison**
Forward Propagation	8897536	0	8897536	55488
Backward Propagation	26956096	291979	27535637	0
Total	35853632	291979	36433173	55488

Approximation of the number of Operations for Convolutional Neural Networks (CNN) VGG7 Architecture				
Operation	**Addition**	**Subtraction**	**Multiplication**	**Comparison**
Forward Propagation	697411584	0	697411584	211584
Backward Propagation	736967936	6190091	744149013	0
Total	1434379520	6190091	1441560597	211584

3 Energy, Latency, Area in VGG3 Architecture

Approximation of Energy for (CNN) VGG3 Architecture (32-bit)				
Operation	**Addition**	**Subtraction**	**Multiplication**	**Comparison**
Forward Propagation	115667 J	0	115667 J	110 J
Total FP	231335 J			
Backward Propagation	350428 J	3793 J	357961 J	0
Total BP	712188 J			
Total	943524 J			

Approximation of the Latency (CNN) VGG3 Architecture				
Operation	**Addition**	**Subtraction**	**Multiplication**	**Comparison**
Forward Propagation	0.2021609155	0	0.2021609155	$1.278 * 10^{-4}$
Total FP	0.404449631			
Backward Propagation	0.6124694572	0.0066340549	0.6256372083	0
Total BP	1.2447407204			
Total	1.649156835			

Approximation of the Area (CNN) VGG3 Architecture	
Forward Propagation	$1.27066 * 10^{10}$
Backward Propagation	$3.911557 * 10^{10}$
Total	$5.18221947 * 10^{10}$

4 Energy, Latency, Area in VGG7 Architecture

Approximation of the Energy (CNN) VGG7 Architecture				
Operation	**Addition**	**Subtraction**	**Multiplication**	**Comparison**
Forward Propagation	9066350 J	0	9066350 J	211 J
Total FP	18132912 J			
Backward Propagation	9580583 J	80471 J	9673937 J	211 J
Total BP	19335203 J			
Total	37468115 J			

Approximation of the Latency (CNN) VGG7 Architecture				
Operation	**Addition**	**Subtraction**	**Multiplication**	**Comparison**
Forward Propagation	15.8458886	0	15.8458886	$4.874895 * 10^{-4}$
Total FP	31.69226469			
Backward Propagation	16.74464847	0.1406450576	16.90780972	0
Total BP	33.79310325			
Total	65.48329719			

Approximation of the Area (CNN) VGG7 Architecture	
Forward Propagation	$9.959073389 * 10^{10}$
Backward Propagation	$1.061937227 * 10^{10}$
Total	$2.057844566 * 10^{10}$

5 Evaluating BNN

Energy Power Latency for Binarized VGG3 architecture:
FP latency = 6.360788000000001e-06
FP Energy = 2558.9759999999997
FP Area = 12352
total latency of model:= 0.000717605556
total Energy of model:= 713803.744
total Area of model:= 39063763456 LUT's

Energy Power Latency for Binarized VGG7 architecture:
FP latency = 0.00028540313600000004 FP Energy = 114819.072
FP Area = 12352
total latency of model:= 0.019605149088
total Energy of model:= 19434565.024
total Area of model:= 1061099905408 LUT's

We have a workload given and the workload is in matrix form, there are alpha entries(rows) and beta columns where alpha is the number of neurons while beta is the number of weights which describe the weight matrix. As for the input matrix which has beta elements as rows and delta elements as columns.
In the case of BNN's we take the design(from somar. Q: How can we link it to the paper?) as a reference to look at a BNN implementation on an FPGA.
We consider only one column with 64 Xnor gates. And push the workload to be accelerated on the crossbar array. This would mean that each row in our workload goes to one crossbar column and that is why we need to multiply everything by alpha.

Suppose we are looking at energy:

$$\text{E} = \sum_{n=0}^{n}(\alpha * \lceil \frac{\beta}{64} \rceil * \delta * E_1) \text{ where n is the number of layers in CNN}$$

After analyzing the energy E of one crossbar column we get $E_1 = 0.012$(from vivado). it is clear that if we have alpha neurons we have to use alpha columns because the rule is that each neuron is always mapped to one column and this is why we have alpha as a factor. As for beta which is the number of weights and since the crossbar is limited (we can only load 64

weights at once) and If we suppose beta was 128 bits we then need to load beta twice since the number of Xnor gates is limited to 64. Which is why beta is divided by 64 in the equation due to our architecture assumption (crossbar is 64). Afterwards we need to multiply by delta because each input is applied and to take every input into consideration.

Suppose we want to Analyze Latency of the model:

$$\text{L} = \sum_{n=0}^{n}(\lceil \frac{\alpha}{64} \rceil * \lceil \frac{\beta}{64} \rceil * \delta * L_1) \text{ where n is the number of layers in CNN}$$

it is dissimilar to how we calculated total Energy of the model. For the Latency we need a different assumption where we have (64 columns X 64 rows) but the speed is actually the same since they should be able to run in parallel (analysis of Latency using vivado $L_1 = 1.909 * (10^{-9})$). We also need to be careful with alpha, if we suppose we have 128 neurons then not all of them can fit in the crossbar since we can only work with 64 and so we need to use the crossbar 2 times. And that is why we divide alpha by the number of columns(which due to our assumption is 64).

As for the Area:

$$\text{A} = A_1 * 64$$

It is quite simply just the analyzed area of one column (193 LUT's) multiplied by 64 (once again due to our assumption). Since the number of columns and rows is fixed.

The above calculations can only be applied to the forward pass of the model since Binarization cannot be applied to the Backward Propagation process. in order to include those numbers within our results we simply used the python tool to calculate the Energy, Latency and Area consumed by the Backward propagation separately and added them together to get the results at the beginning of the section.