Image Classification on FPGA

Honors Project Phase II Review I

Abhinav
Roll.No: 2020BEC0037
Guided by: **Dr.Kala S**

Department of Electronics and Communication Engineering
Indian Institute of Information Technology Kottayam

November 3, 2023

OUTLINE

Introduction

- ATM $\rightarrow$ Approximate convolution $\rightarrow$ CNN convolution layer
- CNN architecture $\rightarrow$ approximated convolution layer 1, pooling layer 1, approximated convolution layer 2, pooling layer 2, fully connected layer
- CNN trained on MNIST dataset
- Evaluation $\rightarrow$ python
- Implemented and synthesized using Vitis HLS

Implementation and Results

- Proposed CNN architecture uses approximate convolution which is implemented using approximate Toom-Cook multiplier with base size 3 [1]
- A pixel of convolved image represents the convolution of subsection of input image with size that of kernel's size and kernel
- In this convolution the multiplication arithmetic operation is evaluated using Approximate Toom-Cook multiplier

Implementation and Results

- Approximate Convolution Layer used in the proposed architecture is shown in Fig. 1

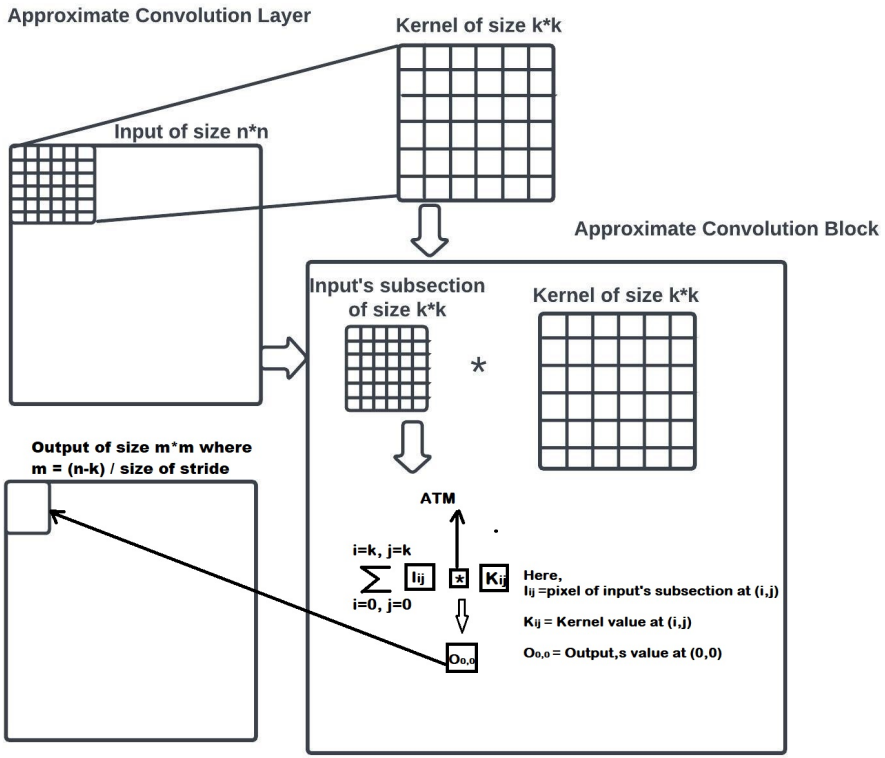

Figure 1: Approximate convolution layer

Implementation and Results

- Dataset $\rightarrow$ 60,000 training images and 10,000 testing images
- The images are grayscale images
- Proposed CNN architecture has been programmed in C++ and synthesized using Xilinx Vivado HLS tool.
- To check the accuracy of approximate CNN, the algorithm is simulated in python.

Implementation and Results

- Proposed CNN architecture takes an image of size 32 * 32 as input
- Architecture consists of two convolution layers and two pooling layers
- 10 kernels and 20 kernels of size seven and stride one is used for the first and second convolutional layer respectively
- 10 kernels of size two, and stride two is used for both of the pooling layers
- These are followed by fully connected layers as shown in Fig.2
- Output layer consists of 10 neurons, each representing a digit

Implementation and Results

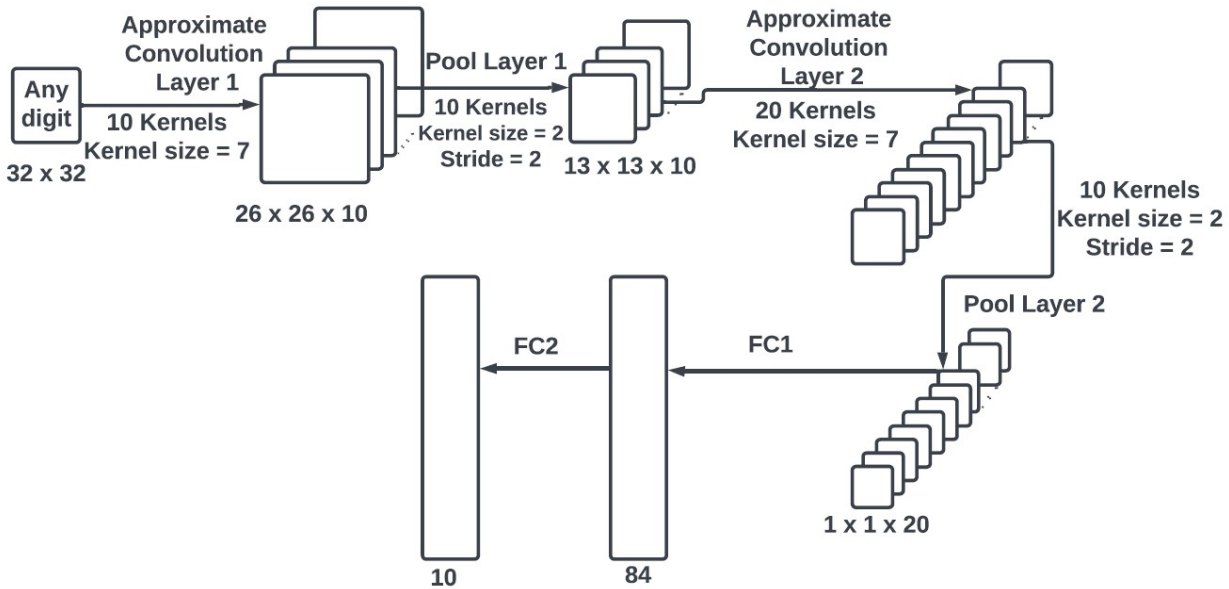

Figure 2: Proposed CNN architecture

Implementation and Results

- Two variants of approximate CNN $\rightarrow$ 8-bit, 16-bit
- Resource utilization of CNN using 8-bit ATM and 16-bit ATM are shown in Table 1 and Table 2 respectively with operating frequency at 150 MHz.

Table 1: Resource Utilization of CNN using 8-bit ATM

Resources	Used	Available	Utilization (%)
LUT	90471	1296000	6.9
FF	41846	2592000	1.2
BRAM	62	2016	3.1
DSP	175	9216	1.9

Implementation and Results

Table 2: Resource Utilization of CNN using 16-bit ATM

Resources	Used	Available	Utilization (%)
LUT	89276	1296000	6.9
FF	32381	2592000	1.6
BRAM	84	2016	4.1
DSP	330	9216	3.5

Implementation and Results

Table 3: Classification report of the proposed architecture

Class	Precision	Recall	F1 Score
0	0.97	0.97	0.97
1	0.99	0.99	0.99
2	0.98	0.99	0.99
3	0.97	0.93	0.95
4	0.93	0.99	0.96
5	0.99	0.96	0.97
6	0.91	0.94	0.93
7	0.93	0.92	0.92
8	0.97	0.99	0.98
9	0.97	0.95	0.96
Accuracy			0.96

Implementation and Results

- Comparision results with existing works are shown in Table 4

Table 4: Comparison with Existing Works

	Our work	[3]	[2]
Approximation	YES	NO	NO
FPGA	Virtex UltraScale+	Zedboard	Virtex-7 xc7vx980
Freq (MHz)	150	330	225
BRAM	84	257	-
DSP	330	188	937
Accuracy (%)	96	97	88.9

Conclusion

- Savings in hardware $\rightarrow$ with penalty in accuracy
- Applications $\rightarrow$ improving the efficiency of the system, where precise computations are not required
- Computational complexity without much compromise in the overall performance
- Proposed CNN architecture has been implemented on Xilinx Virtex UltraScale+ FPGA with an operating frequency of 150 MHz and gave comparable performance with existing Implementation and Results.
- An accuracy of 96% has been reported by the proposed architecture.

Publication

Paper titled ”Approximate CNN on FPGA Using Toom-Cook Multiplier” is accepted at IEEE International Symposium on Smart Electronic Systems (IEEE – iSES)

REFERENCE

[1] Salman Ahmed. “Approximate Computing Architectures and Algorithms for Error Resilient Applications”. In: (July 2021).

[2] Safa Bouguezzi et al. “An efficient fpga-based convolutional neural network for classification: Ad-mobilenet”. In: *Electronics* 10.18 (2021), p. 2272.

[3] Sheping Zhai et al. “Design of convolutional neural network based on fpga”. In: *Journal of Physics: Conference Series*. Vol. 1168. 6. IOP Publishing. 2019, p. 062016.