
Convolution Optimization

— with *Zynq* FPGA —

Michael Losh, Oluwakemi Adabonyan
ECE 495/595 Fall 2016 Final Project Presentation

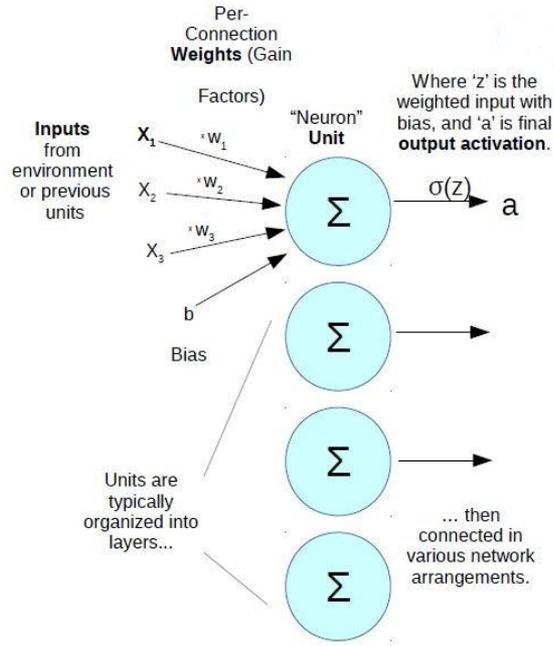
December 2016

Introduction

Background
Objectives



Background



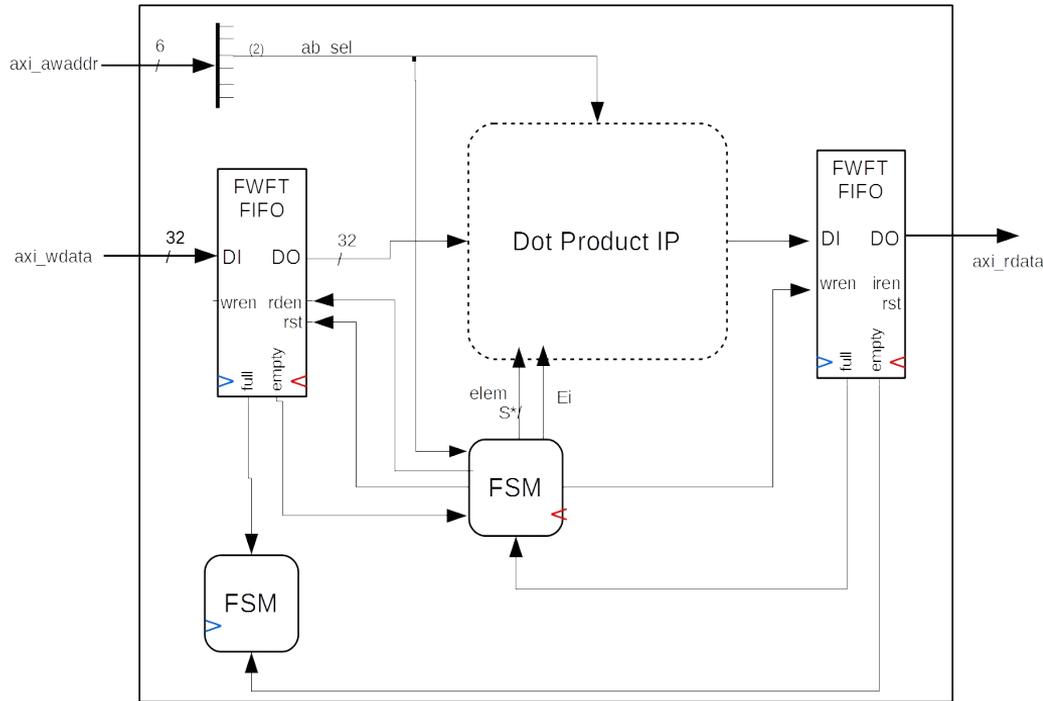
- Inspired by living Organisms
- Model for Signal Processing and Pattern recognition
- Convolutional Neural Networks inspired by visual cortex of Animals.

Objectives

- ☁ Design a Zynq AXI Full Interface peripheral to calculate vector dot products
- ☁ Instantiate a parallel operation in hardware and applied to Convolutional operations
- ☁ Dramatically improve Performance of Neural Network handwriting recognition tasks
- ☁ Determine likely enhancements

Hardware Design

Dot Product Accelerator AXI Full Peripheral IP

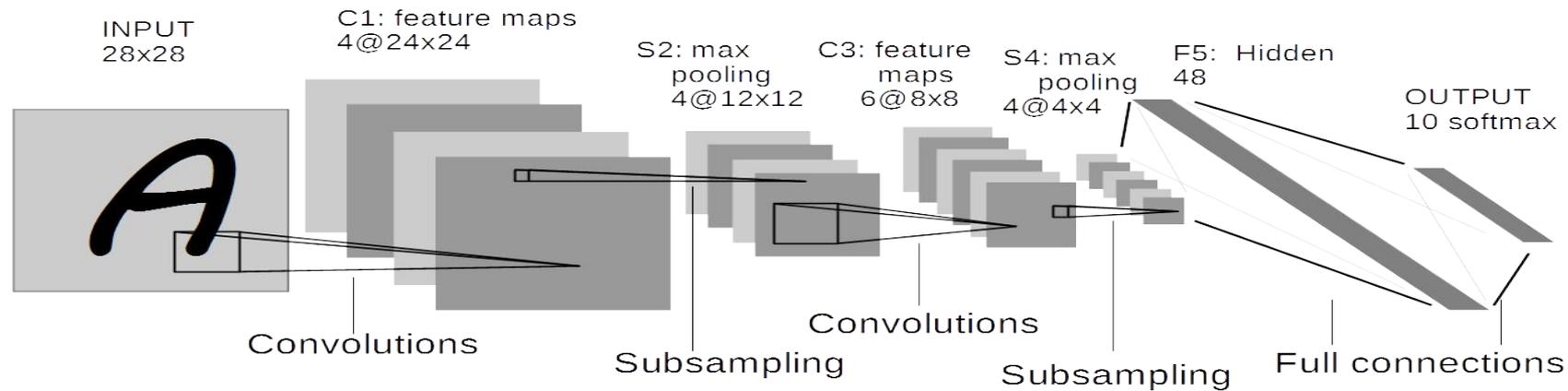


☞ AXI Full interface of the Zynq chip family, interconnecting between the Memory Mapped AXI Masters.

☞ The Dot Product IP, interfaced with the input and output fifos, and two Finite state machines.

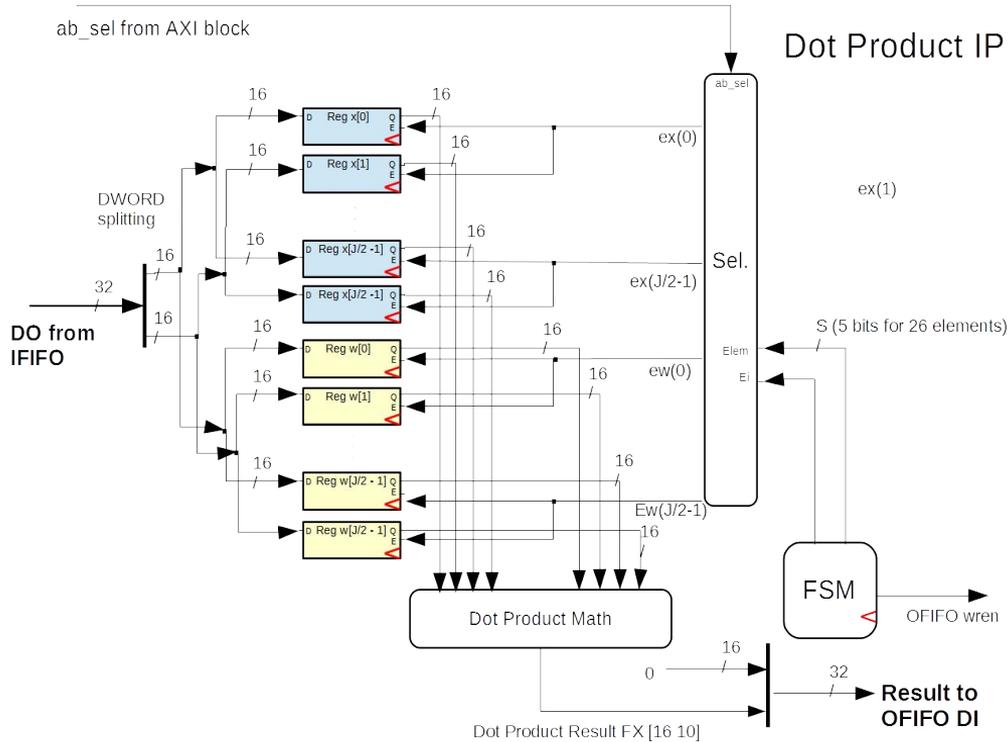
* Using 5 lines for 'elem' ($\text{ceil}(\log_2(26))$)

Design



Simplified CNN for Handwritten Digit Classification

Hardware Design



☞ Produces the total weighted input (Z) to the non-linear function.

☞ Input from IFIFO, output of Registers to Dot Product Math Circuit to OFIFO.

☞ Red FSM provides the 5 bit elem and E_i for the Sel Circuit (A 1-to- J -1 Demux).

☞ Based on the enable from the SEL, the registers are accessed and sent to the Dot Product Math.

☞ Dot Product Math multiplies the weight w and the inputs x which is then summed and outputted to the OFIFO.

Testing and Results

Implementation	Accuracy (out of 10,000 images)	Execution Time
Baseline performance for “random” memory-access pattern version neural network (software only)	98.32%	190.191sec
Performance for AXI dot product hardware-utilizing neural network in convolutional layers	98.32%	183.854 sec
Performance for streamlined dot-product structured neural network (software only)	98.32%	128.728 sec

Conclusion

Results

- ☞ Designed a Zynq AXI Full Interface peripheral to calculate vector dot products
- ☞ Instantiate a parallel operation in hardware and applied to Convolutional operations
- ☞ Hardware Co-processor for Parallel Processing,
- ☞ Faster Hardware than some software implementations

Improvements

- ☞ DMA
- ☞ Increased Parallelization: multiple feature maps (multiple weight vectors) at the same time
- ☞ Reconfigurability- hard-coded weights (distributed arithmetic LUTs?)
- ☞ Parts of the Circuit are generic - make it completely generic

**THANK YOU FOR
LISTENING**



ANY QUESTIONS