ASIC-based Accelerator for CNN Application in Handwritten number Recognition

Saravanan, Shanmathi (MS Electrical Engineering)
Thatikayala, Tejaswi (MS Electrical Engineering)

Introduction
- CNN is one of the most used algorithms for a varied range of visual processing tasks
- Convolution layer has the maximum number of computations among all the layers.
- While performing inference it contributes the maximum to the computation delay.

Proposal
- In this project we implement the convolution layer in CNN on an ASIC based hardware accelerator.
- Parallel computations are implemented to reduce the computational time of convolution layers.
- Higher throughput is achieved by not giving zero-pixel values to multiplication and accumulation units.

Techniques
Null-Hop Technique:
- This technique exploits the sparsity of input feature map to compress the inputs.
- Input features are given to the accelerator in a compressed form.
- The compressed inputs are decoded and are given to MAC units.
- This technique reduces the memory overhead the computation time as the multiplications for the zero-pixel values are avoided.

Architecture
- Block diagram of the design:

Implementation
The accelerator architecture is composed of 2 major blocks:
A. INPUT DATA PROCESSOR
B. MAC Unit

A. INPUT DATA PROCESSOR
- The input is given through a 32-bit bus
- Outputs of this block are pixel value and the coordinates of the pixel.
- This block decodes the compressed inputs and maps the pixel values to its coordinates

B. MAC UNIT
- The prediction output of the inference engine is as follows:

Analysis and Results
- The inputs of convolution layers of CNN are saved as text files and are given to the hardware accelerator.
- The module is verified by comparing the convolution output to the actual output of the convolution layer read from inference model.
- The above is the GTK waveform generated by the MAC unit

Software implementation
- A CNN based model was built for handwritten number recognition and was trained using the MNIST dataset using python.
- The accuracy of the trained model is 99.13%
- The inference engine was run both on a TPU and CPU and the TPU proved to be 12.768 % more time efficient.
- The convolution layer output is as below:

Conclusion
- The inputs to this unit are pixel values and the coordinates.
- The MAC controller receives the input pixel values along with their X-coordinate and Y-coordinate values and passes them to the multipliers.
- The MAC controller keeps track of the multipliers which are free and assigns the incoming value to it accordingly.
- Based on the controller output the appropriate kernel values and the pixel values are sent to multiplier and then accumulated.
- By using the Null Hop method we make sure that the MAC cycles which would generally be wasted on multiplication operation of zero matrix elements are saved.

Acknowledgements
We would like to thank Prof. Thuy Le for his guidance throughout the project. We are grateful for all the suggestions and help that professor provided us throughout our project development and during all our lab work.

Key References