Self-Evalution of the Project Process

James Erik Groving Meade

Advisor: Jens Sparsø

At the beginning of this project, I set out a goal to investigate the potential for designing hardware to meaningfully accelerate the training process of neural networks. While my understanding of the topic has substantially deepened throughout the project, the overall goal has maintained constant throughout.

Eventually, by combing through the literature and thinking critically, I managed to stumble upon a domain in which hardware accelerators could have a great impact: the online training of neural networks. In a metaphoric sense, my original goal at the beginning of the project process was simply a direction toward a hazy horizon. At the end of the process, I had arrived at my destination with a much clearer understanding of the path I had taken and the significance of what I had accomplished. Being able to do go through this process and conduct research in this way has helped me not only to achieve a further understanding of my thesis' subject matter, but also to grow as a critical thinker and engineer.

I believe that having weekly meetings helped drive me to have new content every week. They were helpful to motivate me to consistently develop and deliver. These meetings added a benign pressure and encouraged me to maintain a consistent work ethic throughout the entirety of the process.

Regarding work ethic, I believe I was able to find a good balance through effective time management. However, it was not always easy, especially during the beginning of the semester when I had been interviewing for jobs and teaching several classes.

As is shown in the original and revised project plans, the original schedule changed slightly, as there was more overlap of tasks than originally planned. Regardless, I believe that I have successfully accomplished the goal I set out to achieve when I originally embarked on this thesis. Ultimately, I have experienced a large amount of personal and academic growth throughout this 5-month thesis process that will surely be an invaluable asset to me and to those around me in the future.

Hardware Accelerator for Training Neural Networks

Original Project Plan

James Erik Groving Meade

Advisor: Jens Sparsø

This document outlines the current tentative plan for James Erik Groving Meade's Master's thesis. The objective of my Master's thesis is to design and implement a FPGA accelerator for improving energy and time requirements for training a neural network.

The past few years have been witness to a massive rise in interest regarding machine learning. In this new wave of AI, computer vision has been completely dominated by a new architecture known as convolutional neural network. These networks often contain many hidden layers between input and output, hence the origin of the term "deep learning". These networks use convolutional filters to recognize features and classify images.

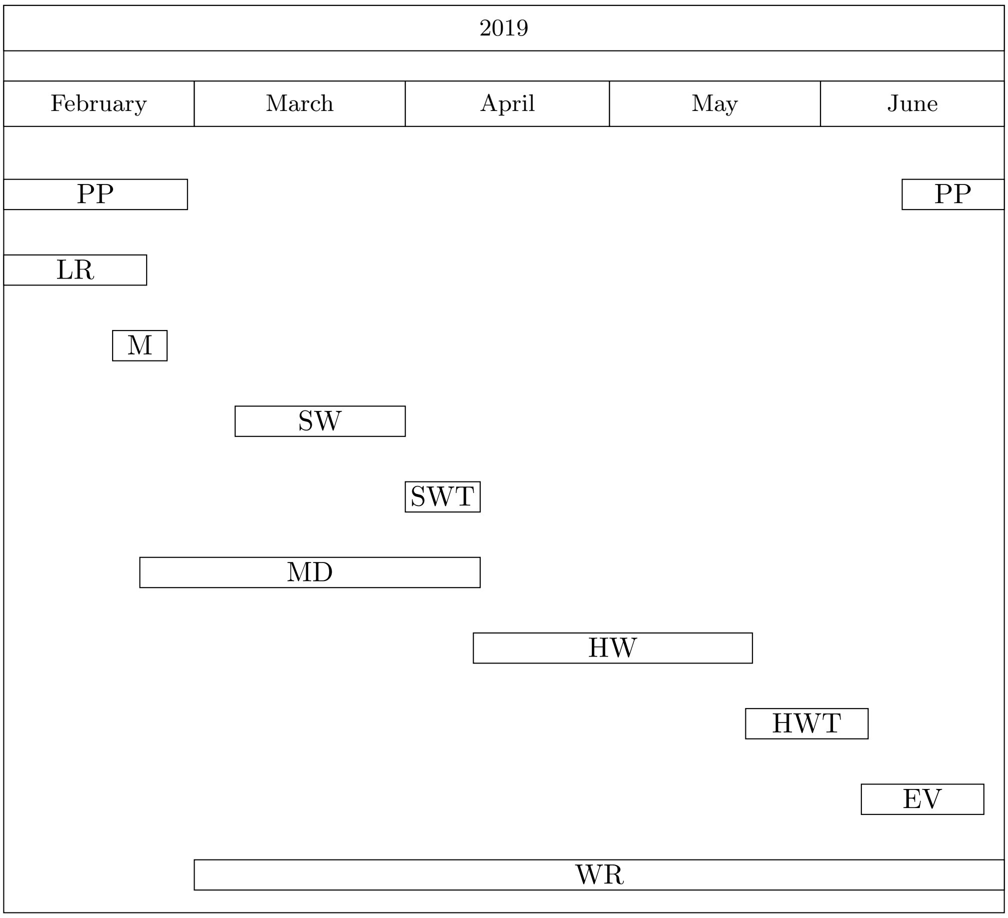
For these networks to operate, all the parameters and weights in the network must be trained using supervised learning and a large labeled training dataset. The training process in its current state can be rather slow and is often done using GPU's due to its massively parallel nature. We have only started to begin to see application-specific hardware being developed, and primarily in academia. Thus, the hardware side of neural networks is very much so in its infancy, as one must understand the intricacies of the high-level algorithms to be able to construct an efficient, targeted, low-level model on which to delegate the training workload.

While there are numerous chips that have been developed to perform inference such as the Eyeriss [1], Google TPU [2], and nn-X chips [3], not much focus has been given to performing the training of neural networks. There has been preliminary academic research such as the F-CNN FPGA model [4], but this work has not investigated reduced precision to improve speedup or energy efficiency. A paper by Courbariaux performed a software simulation of reduced precision and observed that low precision multiplications do not cause too much added error in many cases. *Therefore, the proposed work shall focus on developing an FPGA-model that uses reduced precision and a modular, flexible architecture to achieve efficient and fast neural network training while minimizing the final error loss of the trained network.*

Despite the dearth of research in application specific hardware for neural network training, there are grand use cases for this type of hardware. While the industry has defaulted to using GPU-based cloud computing centers, large energy savings could potentially be realized by switching to more specialized training hardware.

This would result in greener and more cost-effective training and any changes in speedup can be managed by adding more units due to the embarrassingly parallel nature of training neural networks.

Regarding project management, weekly meetings have been arranged between the student and the advisor. Furthermore, all relevant papers, research, work, and code are being maintained in a git repository. Lastly, a tentative schedule for the timely completion of the thesis report has been created and is visible in the below figure.



PP Project plan
LR Literature review
M Creating CPU/GPU models for benchmarking
MD Hardware model design
SW Software implementation of the accelerator model
SWT Testing of software implementation
HW Hardware implementation on FPGA of the accelerator
HWT Testing of hardware implementation
EV Evaluation of results/benchmarks and data gathering
WR Ongoing writing of report and reading literature

Figure 1: Current tentative schedule for completing the thesis

Hardware Accelerator for Training Neural Networks

Revised Project Plan

James Erik Groving Meade

Advisor: Jens Sparsø

This document outlines the results of planning for James Erik Groving Meade's Master's thesis and how it differed from the original plan. The objective of my Master's thesis was to design and implement a FPGA accelerator for improving energy and time requirements for training a neural network.

The past few years have been witness to a massive rise in interest regarding machine learning. In this new wave of AI, computer vision has been completely dominated by a new architecture known as convolutional neural network. These networks often contain many hidden layers between input and output, hence the origin of the term "deep learning". These networks use convolutional filters to recognize features and classify images.

For these networks to operate, all the parameters and weights in the network must be trained using supervised learning and a large labeled training dataset. The training process in its current state can be rather slow and is often done using GPU's due to its massively parallel nature. We have only started to begin to see application-specific hardware being developed, and primarily in academia. Thus, the hardware side of neural networks is very much so in its infancy, as one must understand the intricacies of the high-level algorithms to be able to construct an efficient, targeted, low-level model on which to delegate the training workload.

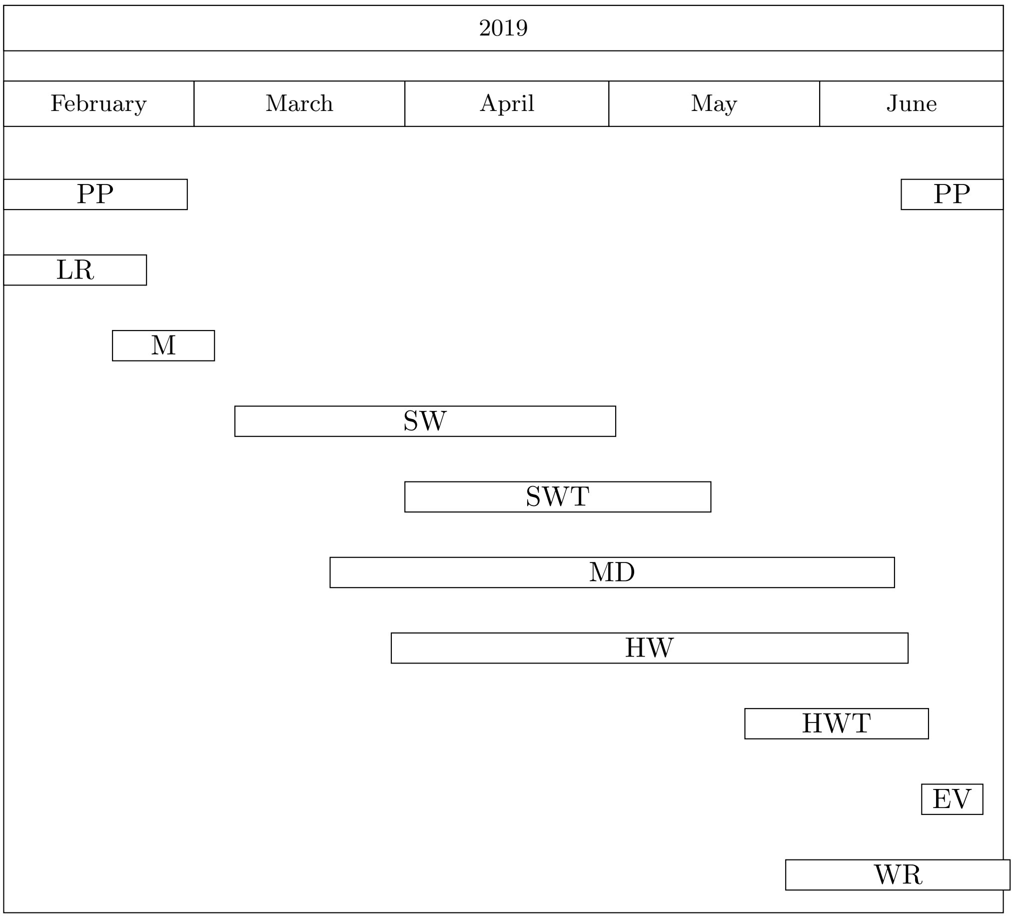
While there are numerous chips that have been developed to perform inference such as the Eyeriss [1], Google TPU [2], and nn-X chips [3], not much focus has been given to performing the training of neural networks. There has been preliminary academic research such as the F-CNN FPGA model [4], but this work has not investigated reduced precision to improve speedup or energy efficiency. A paper by Courbariaux performed a software simulation of reduced precision and observed that low precision multiplications do not cause too much added error in many cases. *Therefore, the proposed work shall focus on developing an FPGA-model that uses reduced precision and a modular, flexible architecture to achieve efficient and fast neural network training while minimizing the final error loss of the trained network.* **This did not change from the original project plan, as a modular and flexible architecture has been implemented. Regarding the original statement on reduced-precision, it was deemed that low precision is able to train to a certain optimum, but is unable to converge to a local optima**

as precision error accumulates through training. It should be noted that the paper by Courbariaux was for Maxout networks, which is a different network type that what was implemented for my thesis.

Despite the dearth of research in application specific hardware for neural network training, there are grand use cases for this type of hardware. While the industry has defaulted to using GPU-based cloud computing centers, large energy savings could potentially be realized by switching to more specialized training hardware. This would result in greener and more cost-effective training and any changes in speedup can be managed by adding more units due to the embarrassingly parallel nature of training neural networks.

Regarding project management, weekly meetings have been arranged between the student and the advisor. Furthermore, all relevant papers, research, work, and code are being maintained in a git repository. Lastly, a tentative schedule for the timely completion of the thesis report has been created and is visible in the below figure.

Differences between the original project plan and the final course of action taken throughout the thesis primarily lie in when tasks were done. In reality, there was a lot more overlap of work between tasks compared to the original project. In addition, the thesis report was written in the last month of the thesis, compared to how it was originally planned to have been written alongside completing the work for thesis.



PP Project plan

LR Literature review

M Creating CPU/GPU models for benchmarking

MD Hardware model design

SW Software implementation of the accelerator model

SWT Testing of software implementation

HW Hardware implementation on FPGA of the accelerator

HWT Testing of hardware implementation

EV Evaluation of results/benchmarks and data gathering

WR Ongoing writing of report and reading literature

Figure 2: Current tentative schedule for completing the thesis

References

[1] Yu-Hsin Chen, Joel Emer, and Vivienne Sze. Eyeriss: A spatial architecture for energy-efficient dataflow for convolutional neural networks. *SIGARCH Comput. Archit. News*, 44(3):367–379, June 2016.

[2] Norman P. Jouppi, Cliff Young, Nishant Patil, David Patterson, Gaurav Agrawal, Raminder Bajwa, Sarah Bates, Suresh Bhatia, Nan Boden, Al Borchers, Rick Boyle, Pierre luc Cantin, Clifford Chao, Chris Clark, Jeremy Coriell, Mike Daley, Matt Dau, Jeffrey Dean, Ben Gelb, Tara Vazir Ghaemmaghami, Rajendra Gottipati, William Gulland, Robert Hagmann, C. Richard Ho, Doug Hogberg, John Hu, Robert Hundt, Dan Hurt, Julian Ibarz, Aaron Jaffey, Alek Jaworski, Alexander Kaplan, Harshit Khaitan, Andy Koch, Naveen Kumar, Steve Lacy, James Laudon, James Law, Diemthu Le, Chris Leary, Zhuyuan Liu, Kyle Lucke, Alan Lundin, Gordon MacKean, Adriana Maggiore, Maire Mahony, Kieran Miller, Rahul Nagarajan, Ravi Narayanaswami, Ray Ni, Kathy Nix, Thomas Norrie, Mark Omernick, Narayana Penukonda, Andy Phelps, and Jonathan Ross. In-datacenter performance analysis of a tensor processing unit. 2017.

[3] Vinayak Gokhale. Nn-x a hardware accelerator for convolutional neural networks. 2017.

[4] Wenlai Zhao, Haohuan Fu, Wayne Luk, Teng Yu, Shaojun Wang, Bo Feng, Yuchun Ma, and Guangwen Yang. F-CNN: an fpga-based framework for training convolutional neural networks. In *27th IEEE International Conference on Application-specific Systems, Architectures and Processors, ASAP 2016, London, United Kingdom, July 6-8, 2016*, pages 107–114, 2016.