CSE145/237D - Project Specification

Doheon Lee, Gabriel Marcano

April 27, 2021

1 Project specification

1.1 Overview

Artificial neural networks (ANNs) are being deployed to solve difficult real-world problems, including object identification in images and video, voice recognition, and autonomous driving. The problem with typical ANNs is that they require a lot of computational power to run, limiting their deployment to systems that can host power-hungry graphic processing units (GPUs). Field programmable gate arrays (FPGAs) could be used as an alternative to GPUs for running ANNs with lower power requirements, but FPGAs tend to be resource constrained, making it difficult to port ANNs to them. This project will build off existing work currently under review that was performed at UC San Diego in optimizing ANNs for FPGAs to investigate what else could be done to further reduce FPGA resource utilization while minimizing accuracy loss. The current work explored optimizations for ANNs with skip connections, or connections between layers skipping a few in between, through two approaches: quantizing the ANN weights, and by using a novel algorithms to reduce the number of skip connections required. This project will continue the research and investigate the effects of different quantization parameters on model accuracy and its impact on said novel algorithm. If time allows, we hope to also explore the impact on accuracy of knowledge distillation (KD), which can further reduce the resource requirement of ANNs on FPGAs.

1.2 Approach

Existing work currently under review examined the effects of the novel SkipTrim algorithm on ResNet of various sizes, but only applied quantization to one of the SkipTrim optimized networks. We will continue where the work left off and explore the effects of the quantization parameters used in the previous work on more ResNet models. After this is achieved we will examine the effects of different, currently not examined quantization parameters on the accuracy of all of the models. If time allows, we will then implement knowledge distillation and re-do all previous experiments with these newly compressed neural networks.

Integer bits	Decimal bits	Total bits
3	5	8
4	12	16

Table 1: Quantization parameters used by existing work

1.2.1 Reproducing existing work

The paper in review describes two general experiments, namely the impact of SkipTrim on ResNet20, ResNet32, ResNet44, ResNet56, and ResNet110, and the effects of quantization on a SkipTrim processed ResNet20 model. We will form a baseline for this project by reproducing the existing work by using SkipTrim on ResNet20, ResNet56, and ResNet110, and then using both SkipTrim and quantization on ResNet20. We will then compare the results from both of our reproductions with the results in the paper. We will leverage existing code and run the training and simulations ourselves, and record and compare the accuracy percentage, the throughput frequency, the number of flip-flops, the number of look-up tables (LUTs), and the Block Random Access Memory (BRAMs) with the existing paper results.

1.2.2 Applying quantization to other ResNet models

After confirming that we are able to reproduce existing results, we will augment the existing code and configuration files to run SkipTrim and apply quantization to ResNet models that were not explored in the paper, specifically ResNet56 and ResNet110. We will gather data for both quantization parameters shown in Table 1 and record the accuracy percentage, the throughput frequency, the number of flip-flops, the number of look-up tables (LUTs), and the Block Random Access Memory (BRAM) utilization.

1.2.3 Experimenting with more quantization parameters

The current work limited quantization to the parameters shown in Table 1. We will test the effects of quantization on model accuracy by applying the parameters in Table 2 to ResNet56 and ResNet110. As before, we will record the accuracy percentage, the throughput frequency, the number of flip-flops, the number of look-up tables (LUTs), and the Block Random Access Memory (BRAM) of each model after it has been processed by SkipTrim and quantized.

1.2.4 Additional ResNet models

sec:additional

After we have completed running the experiments outlined in sections 1.2.2 and 1.2.3, we will repeat the experiments for a wider set of ResNet configurations. Specifically, we will run the experiment in section 1.2.2 for ResNet32

Integer bits	Decimal bits	Total bits
1	1	2
2	2	4
2	6	8
3	13	16
5	11	16
4	20	24
5	19	24
6	18	24

Table 2: Quantization parameters for new work

and ResNet44, and the experiment in section 1.2.3 for ResNet20, ResNet32, and ResNet44. As before, for both experiments we will record the accuracy percentage, the throughput frequency, the number of flip-flops, the number of look-up tables (LUTs), and the Block Random Access Memory (BRAM) utilization of each model after it has been processed by SkipTrim and quantized.

1.2.5 Knowledge distillation

If we succeed in doing the previous experiments and data collections, we plan on implementing knowledge distillation for all of the previously mentioned ResNet models. We will explore the impact of knowledge distillation when it is applied before quantization, and when it is applied afterwards. As with previous experiments, we will record the accuracy percentage, the throughput frequency, the number of flip-flops, the number of look-up tables (LUTs), and the Block Random Access Memory (BRAM).

1.2.6 Additional datasets

If there is still more time left, we plan on running all previous experiments with the CIFAR-100 dataset, and record the same information as before in order to examine the effects of using a different dataset (as every prior experiment has been done with CIFAR-10).

1.2.7 Technologies and libraries

The majority of the work we will be doing will be working with Keras to modify the existing code for handling SkipTrim and related testing. We will also leverage hls4ml along with Vivado HLS to convert and synthesize the FPGA bitstream resulting from all of our models. We will use CIFAR-10 as the dataset for training our models and running all of our experiments, in order to use the same data used by the prior work.

1.3 Objectives

Our minimum viable product is the collection of all of the artifacts described in sections 1.2.1, 1.2.2, and 1.2.3. Specifically, we aim to have a collection of tables listing accuracy and related parameters, a set of all optimized neural networks, a collection of bitstreams based off the optimized neural networks for all experiments, and finally source code and configuration files for all of the experiments.

Should time permit, we plan on completing the experiments described in sections **??** (repeating experiments with more ResNet models), 1.2.5 (knowledge distillation), and 1.2.6 (running prior experiments with different datasets). These new experiments will build off the minimum viable product by reusing the code infrastructure and experimental setup. We will re-running the prior experiments with the new optimization pass, resulting in more tables, more optimized neural networks, more bitstreams, and more source code for the new experiments.

The baseline objective of all of this data collection is to do a more in-depth analysis of the impact on accuracy of SkipTrim with quantization on more than just ResNet20, and to compare those results with non-quantized versions of ResNet. All of this data will help identify further advantages and/or limitations of SkipTrim, as well as help identify the trade-off between some neural network optimization and accuracy.

1.4 Constraints, risk, and feasibility

The biggest risk in this project is that we do not have a lot of experience with neural networks and high-level synthesis (HIL), and as such it is difficult for us to estimate the time requirement of all of the experiments. We have conferred with some of the individuals that have worked on the prior work for guidance, and they seem confident that we will be able to attain our minimum viable product by the end of the quarter.

Another major risk is that we are basing our work off existing code, which means that we will have to spend time learning the codebase to be able to extend it for our work. We have begun to read and watch media about Keras in order to familiarize ourselves with the concepts behind the code. Our first goal is to recreate the existing experiments as a way to familiarize us with the process we will use to collect data for the rest of the experiments. In addition, the existing code was mostly run on Linux, and while it should run on Windows, this has not been tested yet.

In terms of time management, we expect setting up the previously done work will take a non-insignificant amount of time, but it is not clear to us, due to our lack of experience, just how long this will be. Also, all of the training and simulations required to collect data take significant amounts of time, from hours to possibly days. It is in our best interest to begin collecting data as soon as possible, and to distribute the load equally among us, to maximize the amount of training and simulations we can complete in the remaining time.

To counteract a lot of this uncertainty we have limited the number of ResNet models we will analyze to ResNet56 and ResNet110 for our minimum viable product experiments. If we have more time we will repeat the experiments with more ResNet models.

Beyond our minimum viable product, we plan on implementing and testing the effects of knowledge distillation on model accuracy, and testing what effects changing the dataset to train and test the models, but due to our inexperience we suspect doing both experiments might take a considerable amount of time, likely a week or more. This is why, although we believe these experiments will yield useful results for the overarching research, we are not including these in our minimum viable product.

In summary, most of our risks derive from the fact that we are inexperienced in this area of study. We are mitigating risks by running our ideas past others with more experience with machine learning and that have worked with this, as well as by scheduling our time to get a sense of the amount of time we have available and how much time the experiments will require. The simulation and training of the models, in particular, will take a long time just to run and have to be planned for in advance.

2 Group management

2.1 Roles

Since there are only two group members, we will be splitting the tasks rather than having major roles for the group's management. We will both take turns presenting in the in-class presentations.

2.2 Decisions and Designated Communication

Decisions will be made by consensus. Communication will be done mostly over Discord, and some over email. Any deviation from the proposed plan will be presented to Olivia Weng, Alireza Khodamoradi, and Ryan Kastner for confirmation over email. In order to help remain on task, we will meet at least once a week on Friday at 2 PM PST to review our progress that week, take inventory of roadblocks or issues, and make plans or adjustments to our plans to be able to meet all of our deadlines.

2.3 Scheduling and Division of Work

We are using a Gantt chart (see Figure 1) to illustrate our proposed schedule and project scope. Each major goal/milestone is color coded, and subtasks are listed, along with an estimated amount of time required to complete each task. We anticipate that most major milestones will take a week to complete, with the experiments beyond the minimal viable product possibly taking a little longer due to the greater amount of work to achieve those.

Work will be divided to maximize the amount of concurrent tests happening in a single week. Initially, everyone on the team will replicate the existing experiments, as described in section 1.2.1, in order to make sure everyone is able to set up the software and simulations on their own computer systems. For the other experiments, the tasks will be split equally so the experiments can be performed in parallel, as training and simulating each model is expected to take hours, if not days.

The responsibilities are described in the Gantt chart in Figure 1. Each person assigned under each task will be responsible for their own work. As stated previously, we will meet on Fridays to take inventory of progress and complications.

3 Project development

We are responsible for running the experiment outlined in section 1.2.1 to reproduce the existing work, so that both can use that experiment as a baseline for all further work. Most of the subsequent experiments can be split up amongst the team as individual tests are independent from each other.

We will use our own computers with Nvidia GPUs to accelerate the training of models through the use of GPU-accelerated Tensorflow functionality. Simulations of the generated bitstreams will be executed on the same computers on CPU, which will take long amounts of time. See section 1.2.7 for more details on the software technologies being leveraged.

No additional hardware is necessary, as both team members have computers that are able to train and simulate the neural networks, and the FPGA bitstreams are being executed in simulations and not on physical hardware.

Documentation and project reports will be written in LaTeX in Overleaf. Presentation slides will be made in Google Slides. Currently, we have been told that the existing code is not open source, and as such we will not be making it publicly available, beyond deliverables required for this project.

4 Project milestones and schedule

The proposed schedule for managing the primary milestones and sub-components are detailed in Figure 1. The following list shows the main milestones along with the kinds of artifacts generated when achieving each milestone.

4.1 Milestones

- (MVP) Replicate existing work (section 1.2.1)
  - Train neural networks from prior work
  - Convert model to bitstream
  - Run simulation, confirm accuracy match prior work

  - Artifacts: source code, tables, models, simulation results, bitstream

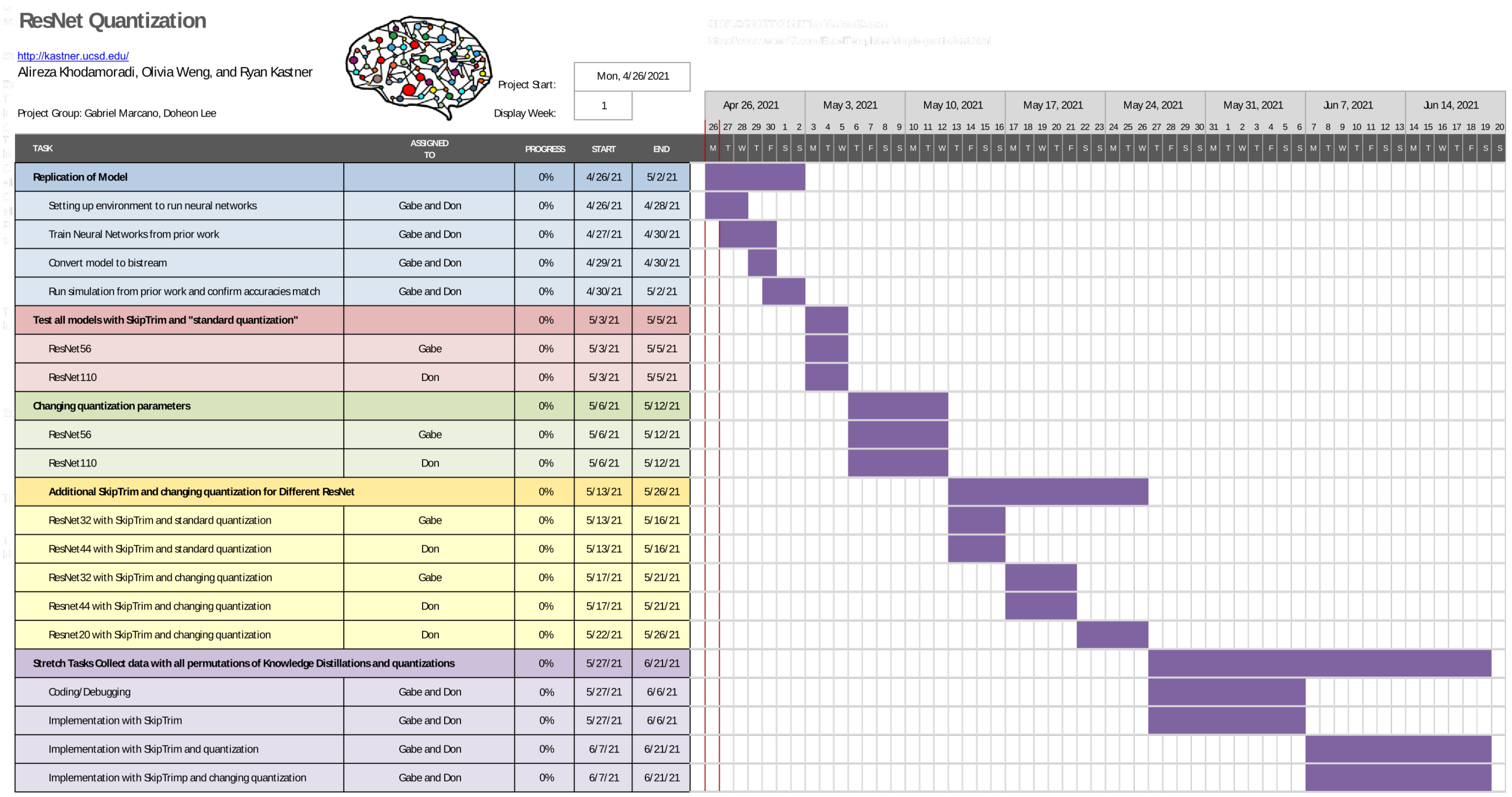
- (MVP) Test ResNet56 and ResNet110 models with SkipTrim and "standard" quantization (section 1.2.2)
  - Train neural networks
  - Convert model to bitstream
  - Run simulation
  - Artifacts: source code, tables, models, simulation results, bitstream
- (MVP) Test ResNet56 and ResNet110 models with SkipTrim and custom quantization (section 1.2.3)
  - Train neural networks
  - Convert model to bitstream
  - Run simulation
  - Artifacts: source code, tables, models, simulation results, bitstream
- Repeat experiments from sections 1.2.2 for ResNet32 and ResNet44
- Repeat experiments from sections 1.2.3 for ResNet20, ResNet32, and ResNet44
- Implement knowledge distillation and simulate its effects on model accuracy for ResNet20, ResNet32, ResNet44, ResNet56, and ResNet110 (section 1.2.5)
  - Implement knowledge distillation
  - Train neural networks, quantize before knowledge distillation
  - Convert model to bitstream
  - Run simulation
  - Train neural networks, quantize after knowledge distillation
  - Convert model to bitstream
  - Run simulation
  - Artifacts: source code, tables, models, simulation results, bitstream for both sets of experiments
- Run previous experiments with new dataset (section 1.2.6)

4.2 Additional tasks

These are tasks required by the class. There is approximately one major task per week from now until the project due date, and these will overlap with the work being done on the experiments.

- Oral project update (due May 4)

- Milestone report (due May 13)
- Project web-presence (due May 20)
- Final Oral presentation (due May 27)
- Final project video (due June 7)
- Final report (due June 10)
- Group evaluation (due June 10)



ResNet Quantization

http://kastner.ucsd.edu/

Alireza Khodamoradi, Olivia Weng, and Ryan Kastner

Project Group: Gabriel Marcano, Doheon Lee

Project Start: Mon, 4/26/2021

Display Week: 1

TASK	ASSIGNED TO	PROGRESS	START	END
Replication of Model		0%	4/26/21	5/2/21
Setting up environment to run neural networks	Gabe and Don	0%	4/26/21	4/28/21
Train Neural Networks from prior work	Gabe and Don	0%	4/27/21	4/30/21
Convert model to bistream	Gabe and Don	0%	4/29/21	4/30/21
Run simulation from prior work and confirm accuracies match	Gabe and Don	0%	4/30/21	5/2/21
Test all models with SkipTrim and "standard quantization"		0%	5/3/21	5/5/21
ResNet56	Gabe	0%	5/3/21	5/5/21
ResNet110	Don	0%	5/3/21	5/5/21
Changing quantization parameters		0%	5/6/21	5/12/21
ResNet56	Gabe	0%	5/6/21	5/12/21
ResNet110	Don	0%	5/6/21	5/12/21
Additional SkipTrim and changing quantization for Different ResNet		0%	5/13/21	5/26/21
ResNet32 with SkipTrim and standard quantization	Gabe	0%	5/13/21	5/16/21
ResNet44 with SkipTrim and standard quantization	Don	0%	5/13/21	5/16/21
ResNet32 with SkipTrim and changing quantization	Gabe	0%	5/17/21	5/21/21
Resnet44 with SkipTrim and changing quantization	Don	0%	5/17/21	5/21/21
Resnet20 with SkipTrim and changing quantization	Don	0%	5/22/21	5/26/21
Stretch Tasks Collect data with all permutations of Knowledge Distillations and quantizations		0%	5/27/21	6/21/21
Coding/Debugging	Gabe and Don	0%	5/27/21	6/6/21
Implementation with SkipTrim	Gabe and Don	0%	5/27/21	6/6/21
Implementation with SkipTrim and quantization	Gabe and Don	0%	6/7/21	6/21/21
Implementation with SkipTrimp and changing quantization	Gabe and Don	0%	6/7/21	6/21/21

Figure 1: Proposed project schedule