CSE145/237D - Milestone Report

Doheon Lee, Gabriel Marcano

May 18, 2021

1 Summary of adjustments to milestones

Our initial project specification assumed that we already had an implementation of Skipper from SkipTrim for Keras, so we scoped the amount of testing around that. In reality, we did not have a Keras implementation of Skipper readily available, so we are developing one as a part of our minimum viable product. This has caused us to shift the amount of testing that we can realistically get done, as implementing and debugging Skip will take some time. We are shifting our focus to running tests primarily with ResNet20, with some additional testing with ResNet56 and ResNet110. All other proposed tests from the project specification have been moved beyond the minimum viable product (with one exception for ResNet32, see Section 2.1). Also, due to the 30+ hours model to FPGA bitstream conversion requires per model, we will be pushing back the ResNet110 conversions until after the MVP is completed.

2 MVP milestones

2.1 Milestone 1: Reproducing existing work

Table 1 shows the milestone task allocation and current status for our first milestone.

The majority of milestone 1 is complete. Most of the Skipper implementation is also done, specifically the knowledge distillation component of Skipper, but the iterative approach to removing skip connections while training is still in development, but should be completed this week. As soon as Skipper is completed we should be able to collect the data for the last two rows of Table 1. This is in our critical path, as every other task in all remaining MVP milestones require Skipper.

Gabriel collected some preliminary data on ResNet32 at the behest of one of the PIs, which was not included in our original project specification. We have also collected additional accuracy data using using a traditional knowledge distillation training approach for ResNet20 and ResNet32, to serve as an additional point of comparison to regular training and Skipper.

Tasks	Status	Owner
Source code implementing Skipper in Keras	In progress	Gabriel
Accuracy data and graphs for ResNet20	Done	Gabriel
Accuracy data and graphs for ResNet32	Done	Gabriel
Accuracy data and graphs for ResNet56	In progress	Doheon
Accuracy data and graphs for ResNet110	In progress	Gabriel
Accuracy data and graphs for ResNet20 with Skipper when quantized to <16, 4>	Pending	Doheon
Accuracy data and graphs for ResNet20 with Skipper when quantized to <8, 3>	Pending	Gabriel

Table 1: Milestone 1 tasks.

We intended to have completed milestone 1 by May 13, and thus we are slightly behind schedule. We are hoping to have it completed by May 21 to give us enough time to complete all remaining tasks for the remaining milestones.

2.1.1 Initial results

Each data collection consists of training each model a total of five times, and averaging the maximum accuracy across the five trials, as well as computing the standard deviations of the maximum accuracy. We have collected some preliminary accuracy comparisons between ResNets trained normally, ResNet models with skips removed trained via normal knowledge distillation, and ResNet models with skips removed trained normally. Tables 2 and 3 show the summary of the data.

For the ResNet20 experiments in Table 2, it appears that knowledge distillation alone is no better than training the network with skips removed from scratch.

Model	Accuracy mean	Accuracy standard deviation
ResNet20	0.9068	0.07296
ResNet20 with no skips (KD)	0.8982	0.0705
ResNet20 no skips	0.9001	0.0784

Table 2: ResNet20 initial training accuracy comparison.

For the ResNet32 experiments in Table 3, it appears that knowledge distillation alone is no better than training the network with skips removed from scratch, just as with ResNet20.

Model	Accuracy mean	Accuracy standard deviation
ResNet32	0.9161	0.0784
ResNet32 with no skips (KD)	0.8997	0.0825
ResNet32 no skips	0.8959	0.1003

Table 3: ResNet20 initial training accuracy comparison.

We are actively collecting similar data for ResNet56 and ResNet110 while we wait for Skipper to be implemented, and should have this data ready in the next few days.

Figures 1, 2, and 3 show representative samples of the accuracy graphs generated per trials for each of the ResNet20 tests represented in Table 2.

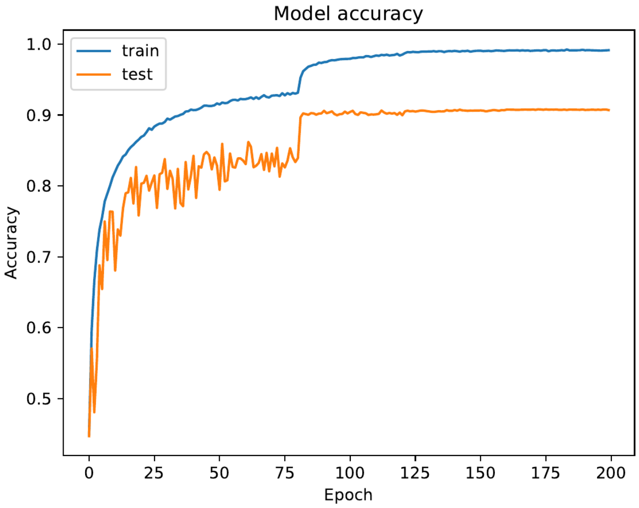

Figure 1: ResNet20 accuracy plot, showing the accuracy against the testing/validation set and against the training set. The major jumps seen in the graph correspond to learning rate adjustments, such as the large jump at epoch 80. ResNet32 yields slightly higher accuracy compared to ResNet20, as expected.

Similarly, Figures 4, 5, and 6 do the same for the ResNet32 trials represented in Table 3.

We have shared a Google Drive folder with the instructor and one of the PIs, containing our source code along with our current data and graphs.

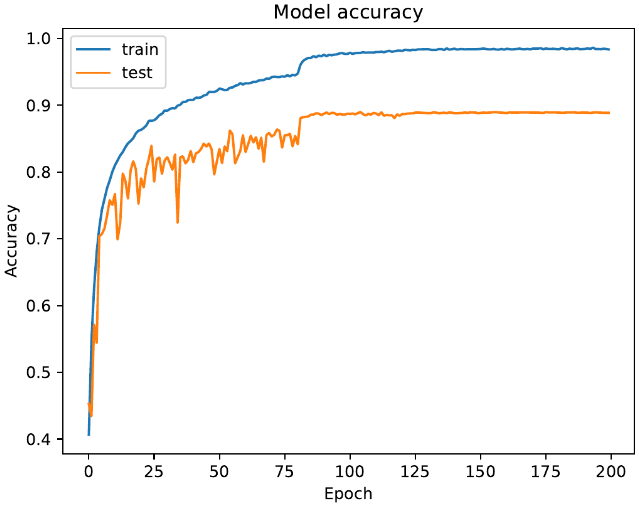

Figure 2: Plot of ResNet20 with skip connections removed trained by knowledge distillation accuracy plot, showing the accuracy against the testing/validation set and against the training set. The major jumps seen in the graph correspond to learning rate adjustments, such as the large jump at epoch 80. Notably, there is a slightly lower increase in accuracy around epoch 80 compared to Figure 1.

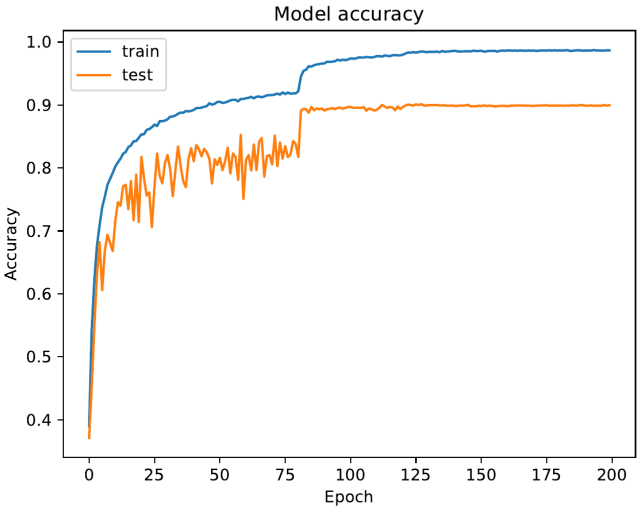

Figure 3: Plot of ResNet20 with skip connections removed trained normally, showing the accuracy against the testing/validation set and against the training set. The major jumps seen in the graph correspond to learning rate adjustments, such as the large jump at epoch 80.

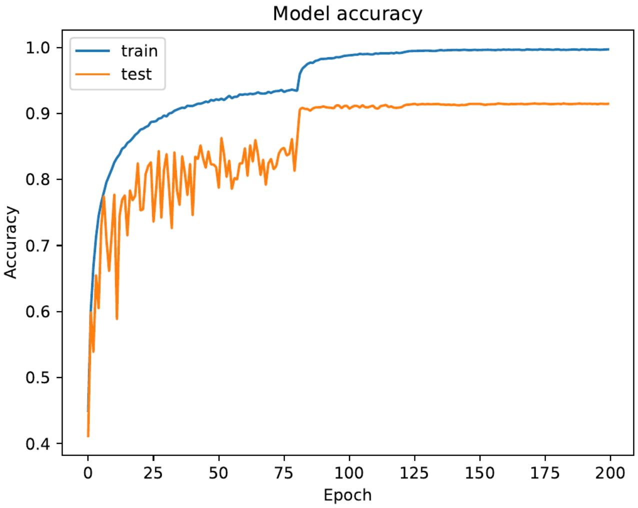

Figure 4: ResNet32 accuracy plot, showing the accuracy against the testing/validation set and against the training set. The major jumps seen in the graph correspond to learning rate adjustments, such as the large jump at epoch 80.

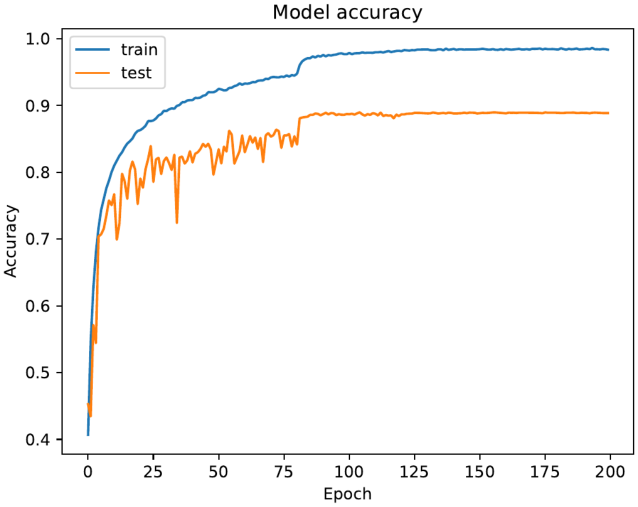

Figure 5: Plot of ResNet32 with skip connections removed trained by knowledge distillation accuracy plot, showing the accuracy against the testing/validation set and against the training set. The major jumps seen in the graph correspond to learning rate adjustments, such as the large jump at epoch 80. Notably, there is a slightly lower increase in accuracy around epoch 80 compared to Figure 4.

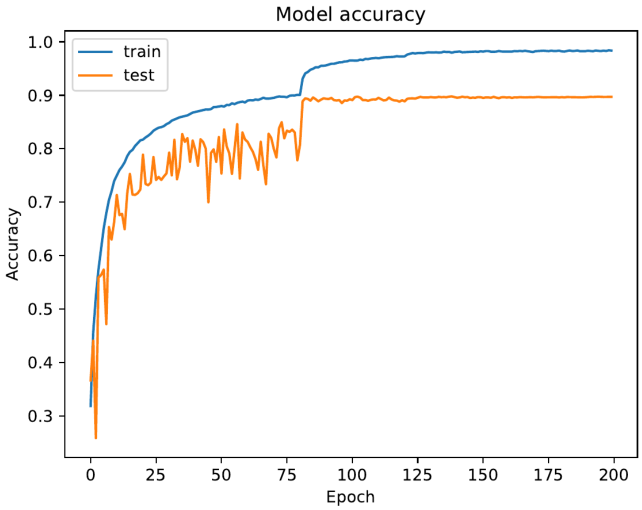

Figure 6: Plot of ResNet32 with skip connections removed trained normally, showing the accuracy against the testing/validation set and against the training set. The major jumps seen in the graph correspond to learning rate adjustments, such as the large jump at epoch 80.

2.2 Milestone 2: New work with ResNet56 and ResNet110

Tasks	Status	Owner
Accuracy data and graphs for ResNet56 with Skipper when quantized to <16, 4>	Pending	Gabriel
Accuracy data and graphs for ResNet56 with Skipper when quantized to <8, 3>	Pending	Gabriel
Accuracy data and graphs for ResNet110 with Skipper when quantized to <16, 4>	Pending	Doheon
Accuracy data and graphs for ResNet110 with Skipper when quantized to <8, 3>	Pending	Doheon
FPGA utilization report for Skipper ResNet56 quantized <16,4> model	Pending	Doheon
FPGA utilization report for Skipper ResNet56 quantized <8,3> model	Pending	Gabriel

Table 4: Milestone 2 tasks.

Milestone 2 depends on Skipper being finished. As soon as Skipper is completed, we should be able to begin collecting the necessary accuracy data in parallel.

We decided to focus on converting ResNet56 to the FPGA bytecode as it takes 30 hours to convert a single model. We have moved converting ResNet110 to FPGA bytecode as part of tasks in Table 9 for after we have completed our MVP.

As we need milestone 1 to be able to collect the data for milestone 2, the earliest we can begin to collect data is May 21. We expect this collection to take 1 week, so it should be completed by May 28.

2.3 Milestone 3: New quantization work with ResNet20

Tasks	Status	Owner
Accuracy data and graphs for ResNet20 with Skipper when quantized to <8, 2>	Pending	Gabriel
Accuracy data and graphs for ResNet20 with Skipper when quantized to <16, 3>	Pending	Doheon
Accuracy data and graphs for ResNet20 with Skipper when quantized to <16, 5>	Pending	Gabriel
FPGA utilization report for Skipper ResNet20 quantized model <8, 2>	Pending	Doheon
FPGA utilization report for Skipper ResNet20 quantized model <16, 3>	Pending	Gabriel
FPGA utilization report for Skipper ResNet20 quantized model <16, 5>	Pending	Doheon

Table 5: Milestone 3 tasks.

We believe we should be able to complete the tasks as currently outlined for milestone 3. These collections should be very similar to the ones done for milestone 2, so all we really need is time in order to complete them.

We may be able to begin some data collection before milestone 2 is completed, but we expect to be done with milestone 3 by June 4.

3 Above and beyond

These are tasks that we hope we may be able to complete once we have completed our MVP. The tables are ordered in approximate order of priority. All of these tasks are not particularly difficult to perform (with the exception of implementing Trimmer) once all prior milestones have been implemented.

Tasks	Status	Owner
Accuracy data and graphs for ResNet56 with Skipper when quantized to <2, 1>	Pending	
Accuracy data and graphs for ResNet56 with Skipper when quantized to <4, 2>	Pending	
Accuracy data and graphs for ResNet56 with Skipper when quantized to <8, 2>	Pending	
Accuracy data and graphs for ResNet56 with Skipper when quantized to <16, 3>	Pending	
Accuracy data and graphs for ResNet56 with Skipper when quantized to <16, 5>	Pending	

Table 6: Task for collecting accuracy of ResNet56 with Skipper and when quantized.

Tasks	Status	Owner
Accuracy data and graphs for ResNet110 with Skipper when quantized to <2, 1>	Pending	
Accuracy data and graphs for ResNet110 with Skipper when quantized to <4, 2>	Pending	
Accuracy data and graphs for ResNet110 with Skipper when quantized to <8, 2>	Pending	
Accuracy data and graphs for ResNet110 with Skipper when quantized to <16, 3>	Pending	
Accuracy data and graphs for ResNet110 with Skipper when quantized to <16, 5>	Pending	

Table 7: Task for collecting accuracy of ResNet110 with Skipper and when quantized.

Tasks	Status	Owner
FPGA utilization report for Skipper ResNet56 quantized model <2, 1>	Pending	
FPGA utilization report for Skipper ResNet56 quantized model <4, 2>	Pending	
FPGA utilization report for Skipper ResNet56 quantized model <8, 2>	Pending	
FPGA utilization report for Skipper ResNet56 quantized model <16, 3>	Pending	
FPGA utilization report for Skipper ResNet56 quantized model <16, 5>	Pending	

Table 8: Task for collecting FPGA resource utilization information of ResNet56 with Skipper and when quantized.

Tasks	Status	Owner
FPGA utilization report for Skipper ResNet110 quantized <16,4> model	Pending	
FPGA utilization report for Skipper ResNet110 quantized <8,3> model	Pending	
FPGA utilization report for Skipper ResNet110 quantized model <2, 1>	Pending	
FPGA utilization report for Skipper ResNet110 quantized model <4, 2>	Pending	
FPGA utilization report for Skipper ResNet110 quantized model <8, 2>	Pending	
FPGA utilization report for Skipper ResNet110 quantized model <16, 3>	Pending	
FPGA utilization report for Skipper ResNet110 quantized model <16, 5>	Pending	

Table 9: Task for collecting FPGA resource utilization information of ResNet110 with Skipper and when quantized.

Tasks	Status	Owner
Accuracy data and graphs for ResNet44	Pending	
Repeat prior experiments with a different dataset	Pending	
Implement Trimmer	Pending	
Use Trimmer with ResNet56 and ResNet110	Pending	
Use Trimmer with quantized versions of ResNet56 and ResNet110	Pending	
Run simulations of FPGA bitstreams	Pending	

Table 10: Additional tasks.