ECE 661: Homework #4

Pruning and Fixed-point Quantization

NetID: wh162

1. True/False Questions
   1. Problem 1.1: True, weight pruning is a technique that discard unnecessary data in a model without affecting the accuracy, whereas weight optimization is how the weight is optimize for reducing the bit representative. So these two method doesn’t intervene with each other.
   2. Problem 1.2: False, even though the parameters are pruned to zero, the GPU still will compute the data when feed into it.

* 1. Problem 1.3: False, though pruning will reduce the number of weight, the bit representative for a weight is still in 32/64 bit, which would take Huffman encoding more time to compute.
  2. Problem 1.4: False, pruning is to remove the weights that would less affect the accuracy of the model, it is not guarantee that the value of the weight would be exact zero, so pruning is still necessary.
  3. Problem 1.5: False, as from the slides, thought soft thresholding reveals the bias problem of L1, the issue was solved by SCAD and MCP.
  4. Problem 1.6: True, refer to lec13 page 21.
  5. Problem 1.7: True, refer to lec 14 page 16.
  6. Problem 1.8: True, refer to lec 14 page 27-28
  7. Problem 1.9: True, refer to lec15 page 10.
  8. Problem 1.10: True, refer to Lec 15 page 28.

1. Lab1: Sparse optimization of linear model
   1. Problem 2.1:

Text, letter

Description automatically generated

* 1. Problem 2.2:

From the plot below, we can see that W is converging to a value after several steps are executed. Yes, W is converging to a optimal solution but not a sparse solution.

Graphical user interface

Description automatically generated with low confidence

* 1. Problem 2.3:

From the plot below, we can see that W is converging to a value after several steps are executed. Yes, W is converging to an optimal solution and a sparse solution.

Diagram

Description automatically generated with medium confidence

* 1. Problem 2.4:
     1. Alpha = 0.2, from the plot we can see that loss converged to -3 after some steps and W hasn’t reach a steady point after 200 steps

Graphical user interface

Description automatically generated with medium confidence

* + 1. Alpha = 0.5, we can see that the loss is much lower the alpha=0.2, however, the weight is still dropping and not yet reach a steady state.

Line chart

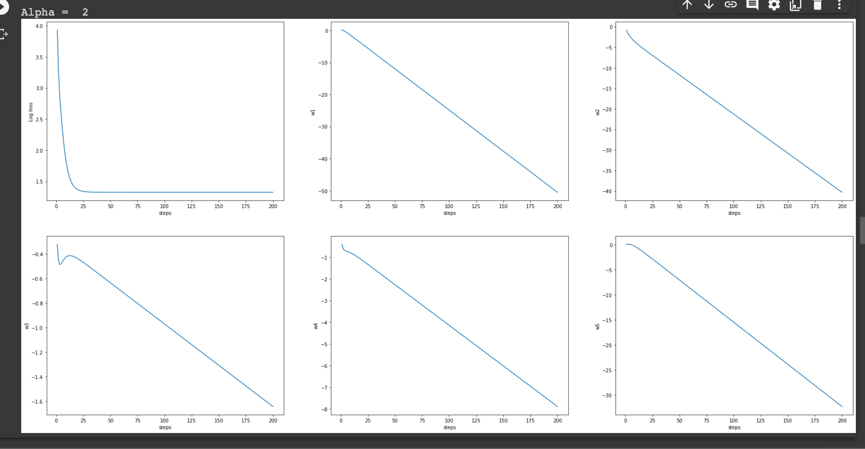
Description automatically generated with medium confidence

* + 1. Alpha = 1, loss converge at 0 and this might the best case, weight has been dropping drastically.

Line chart

Description automatically generated

* + 1. Alpha = 2, loss was greater than the previous alpha values. However, the weight turns into a steady drop, unlike the previous ones, which increases and decreases, when alpha = 2, the weight dropped consistency.



* 1. Problem 2.5:
     1. Threshold = 0.004, we can see that comparing with 2.4, the loss is dropping consistency, whereas the weight is converging, though some of the data have a bump.

Diagram

Description automatically generated with medium confidence

* + 1. Threshold = 0.01, we can see that all the weight converges when reached step 200, however the loss is not converging.

Graphical user interface

Description automatically generated with low confidence

* + 1. Threshold = 0.02, all the weight had converged, and so does the loss, the performance was better than 2.4

A picture containing shape

Description automatically generated

* + 1. Threshold = 0.04, we can see that the convergence of the weight reaches their steady point much faster than the previous ones. Loss, on the other hand, doesn’t converge yet after 200 steps.

Graphical user interface

Description automatically generated with medium confidence

* 1. Problem 2.6:

1. Lab2: Pruning ResNet-20 model
   1. Problem 3.1:

The accuracy of the floating-point pretrained model is 0.9151

Graphical user interface, text

Description automatically generated

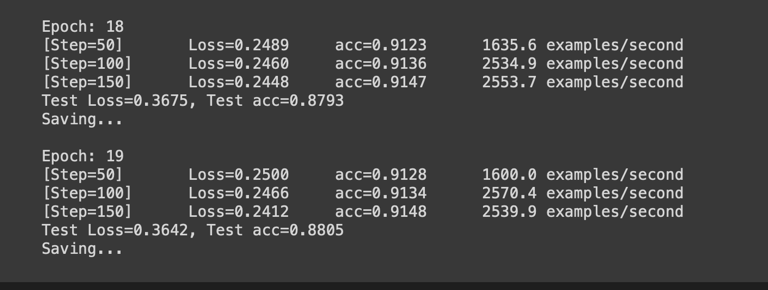
* 1. Problem 3.2:

Text

Description automatically generated

* 1. Problem 3.3:

From the result shown in the image, the best accuracy is 88.05% after 20 epochs of training.



* 1. Problem 3.4:

From the testing result, the best accuracy occurs at the first two epochs with accuracy = 91.55%, which is different with the previous result, which increases gradually.

Graphical user interface

Description automatically generated

* 1. Problem 3.5:

The result shows the same which the best accuracy occurs during the first couple epochs. The total sparsity was 79%.

A picture containing graphical user interface

Description automatically generated

Text

Description automatically generated

1. Lab3: Fixed-point quantization and fine-tuning
   1. Problem 4.1:

Text

Description automatically generated

* 1. Problem 4.2:

From the result, we can see that when Nbits = 6, the accuracy is the best, and it decreases as we lower the Nbit value.

Text

Description automatically generated

* 1. Problem 4.3:

As we can see from the following results, when the value of Nbit is greater, the accuracy is better, and so does the quantize results.

* + 1. Nbits = 4

Calendar

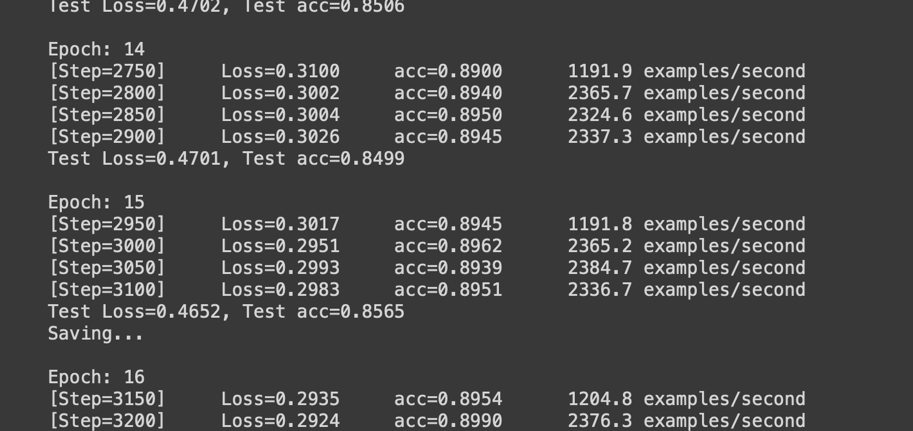
Description automatically generated

* + 1. Nbits = 3

Calendar

Description automatically generated

* + 1. Nbits = 2



* 1. Problem 4.4:
  2. Problem 4.5: