Homework Assignment #1

Submission Due: 2021/03/21 23:59

Objective

1. In this homework assignment, you will learn to train a simple CNN with post-training quantization.
2. You will need to get familiar with the PyTorch code. Any discussions are welcome on the forum.

Action Items

1. In this assignment you need to handle with PyTorch using Google Colaboratory. Set up your own notebook copy from our template:
   - ✓ Click the link: https://drive.google.com/file/d/1A5MrBJ8ViQ0ENrBKPRQZfgNQXUTwgGu7/view?usp=sharing and select "Open with Google Colaboratory.
   - ✓ Select File -> Save a copy in Drive
   - ✓ Select Runtime -> Change runtime type
   - ✓ Select Hardware Accelerator pull-down menu -> Select GPU
2. The template trains a simple LeNet for CIFAR10 dataset. Take a look at the CNN structure. What is the type (convolution, pooling, fully-connected layer, etc.), input activation size, output activation size, and activation function (if any) of each layer?
3. Visualizing weights:
   a. Plot histograms of the weights for every convolutional and fully-connected layer. Discuss any observations you make about the data distribution.
   b. Record the range of the weights, as well as their 3-sigma range (the difference between μ + 3σ and μ − 3σ). For which layers is the 3-sigma range larger or smaller than the actual range? Explain which kind of range you would prefer to quantize the weights for each layer.
4. Quantizing weights:
   Computation of convolution or fully-connected layer can be expressed as
   $$W * I = O$$
   where W is the weight tensor, I is the input tensor, and O is the output tensor. Let n_w is the scaling factor. The quantized computation can be expressed as
   $$n_w W * I = n_w O.$$
   Note that we only quantize weights and did not consider biases in this step.
   a. Fill in the `quantized_weights` function. The template code will call this function to lower the floating-point weights of every layer into 8-bit signed integer precision.
   b. Record the accuracy degradation of the CNN after quantization. If you've done everything correctly, the accuracy degradation should be negligible.
5. Visualizing activations:
   The template code will record every pixel's values when running on a subset of the training set.
   a. Plot histograms of the input images and the output activations of every convolutional and fully-connected layer. Discuss any observations about the value distribution.

b. Record the range of the values, as well as their 3-sigma range (the difference between μ + 3σ and μ − 3σ). For which layers is the 3-sigma range larger or smaller than the actual range? Explain which kind of range you would prefer for quantization.

6. Quantizing Activations:
   a. The output of `conv1` can be described as
   $$W_{\text{conv1}} * I = O_{\text{conv1}}.$$
   Let the scaling factor of the input matrix (I) be n_{I}, the scaling factor of the weight matrix (W_{conv1}) be $n_{W_{\text{conv1}}}$, and the scaling factor of the output matrix (O_{conv1}) be $n_{O_{\text{conv1}}}$. Derive an equation for the output of the `conv1` layer.
   b. Derive an equation for the `conv2` layer with I, W_{conv1}, W_{conv2}, O_{conv1}, O_{conv2}, n_{I}, $n_{W_{\text{conv1}}}$, $n_{W_{\text{conv2}}}$, $n_{O_{\text{conv1}}}$, and $n_{O_{\text{conv2}}}$. You may assume that pooling layers do not exist.
   c. Complete the `quantize_initial_input` and `quantize_activations` functions to calculate the scaling factors for the initial input image to the CNN, and the outputs of each layer, respectively.
   d. Complete the `forward` function for the `NetQuantized` class. You will have to add code here to scale the outputs of each layer, and then to clamp the outputs of each layer to integers between -128 and 127 afterwards.
   e. Record the accuracy of your CNN with quantized weights and activations. If you've done everything right, you should still find that the accuracy degradation is negligible.
7. Quantizing Biases:
   a. In this step, we include `net_with_bias` to create and train a new CNN with bias in its final fc3 layer. For a layer before quantization:
   $$W * I + \beta = O$$
   where β is the bias. Derive the equation of a quantized layer with a bias.
   b. Complete `quantized_bias` function in the `NetQuantizedWithBias` class. This function is to quantize the bias on the final layer. Note that biases are commonly quantized to 32-bits. Therefore your bias values are not necessary between -128 and 127.
   c. What is your accuracy before and after quantizing CNN with the bias? The accuracy degradation should be negligible.
8. Submission:
   a. Submit the PDF report according to the questions. Include the plots in your writeup. Use the following file name:
   `hw1_YourStudentID.pdf`
   The report template consists of Design Concept, Simulation and Discussion, and Summary. Note that for this homework assignment, you may describe the CNN architecture from the code in the section of Design Concept.
   b. Also, hand in your source code with the following file name:
   `hw1_YourStudentID.ipynb`