Indian Institute of Science, Bengaluru

CSA

Spring 2023

Project #2 Machine Learning Optimization Evaluation,

Due: March 20th 2023

Overview:

In this model, you will learn a Machine Learning model quantization algorithm and evaluate its effectiveness in 4 dimensions: 1) Runtime Improvement, 2) Memory Usage Improvement, 3) Model Compression Ratio, and 4) Accuracy Impact.

ML Model Quantization Algorithm:

Researchers and companies, who are trying their best to get the best model accuracy, are making Machine Learning models more pertinent to our daily life. Moreover, state-of-art Machine Learning models are becoming bigger and more complex. Bigger and more complex models can lead to better model accuracy because they tend to capture more information about given inputs. However, in certain scenarios where operations are run on edge devices such as mobile phones and IoT devices, whose memory and computational resources are limited, it is preferable to use models whose computational requirements are less intensive than full-scale deep models. Model quantization is one of the techniques that would reduce the complexity of a given model while having little model accuracy degradation. Quantization techniques convert floating point operands (4 bytes and complex to perform operations on) to integers with a few bits (1 byte to 4 bytes). Thus, edge devices could perform efficient ML model inference. In this project, you are asked to evaluate one of the popular quantization techniques provided in PyTorch in 4 dimensions by completing the tasks mentioned below.

Starter Code:

Get your starter code here (proj2).

Task 1 Train a small model (GPU or CPU):

In this task, you should first train a model to classify the MNIST data set. The data set is included in the proj2 folder, and you only need to use the “load_data” function to get dataloaders for both training and testing. The model you are required to use is called SmallNet in smallNet.py. The testing function is provided in utils.py. Please use the hyperparameters below to train your model.

Training batch	64
Epoch	10
Optimizer	SGD
Learning rate	0.01

After 10 epochs, your model should have an accuracy of more than 90%. Be sure to test your model using the "test" function and save the model weights in your local machine. The starter code is in train.py.

Task 2 Using Quantize the SmallNet (CPU only):

Follow this tutorial section PTSQ API Example to quantize your model and save the quantized model on your local computer. Please note that the PyTorch quantization backend only works on CPUs. Please do not use GPUs for this task. Be sure to check your model accuracy after quantization. The starter code is in convertQ.py.

Task 3 Evaluation (CPU only):

In this task, you are required to evaluate the quantization technique's impact on model performance. In your project report, include all the measurements and generate your conclusion on quantization. Measurements include:

- Model Accuracy before and after quantization
- Model runtime before and after quantization
- Model compression ratio
- Inference Peak memory usage before and after quantization (batch size 64)

The starter code is in measureFloat.py and mesureInt8.py.

Delivery:

Your code in proj2 compressed into a zip.
Your final report containing all the measurements and your conclusion in task 3.