Quantized Machine Learning

Objective:

The main objective of this assignment was to gain hands-on experience in quantizing a logistic regression model using PyTorch. Specifically, we focused on applying dynamic quantization to reduce model size, decrease inference time, and maintain accuracy to an acceptable level, with the MNIST dataset as our test case.

Assignment Steps:

1. Environment Setup and Data Preparation:

In the initial setup, we installed and imported essential libraries, including `sklearn` for machine learning modeling, `numpy` for numerical operations, and quantization tools from PyTorch. We then prepared the MNIST dataset:

- We loaded the MNIST dataset using PyTorch's `torchvision` library, transformed it for normalization, and split it into training and testing sets.
- This setup allowed us to simulate a typical image classification scenario and provided a robust dataset for testing model quantization.

2. Building the Logistic Regression Model:

We employed a logistic regression model using `sklearn`'s `LogisticRegression` class to classify the MNIST digits. Logistic regression is a relatively simple model, suitable for evaluating the effects of quantization without the additional complexities of deep neural networks.

3. Model Evaluation (Pre-Quantization)

Before applying quantization, we assessed the baseline model's:

- Accuracy on the test set, allowing us to measure the performance of the original model.
- Model Size, which was recorded by saving the model and checking its file size. This step provided a benchmark for comparing with the quantized model.
- Inference Time by measuring the time it took for the model to make predictions on a sample set of data, giving us an estimate of processing efficiency.

Results:

- The logistic regression model achieved an accuracy that was suitable for a baseline MNIST classifier.
- The model size was relatively small but could be optimized further.

- The inference time was acceptable but could be improved to enhance efficiency in real-time applications.

4. Quantization of the Logistic Regression Model:

We defined a custom function, `quantize_model`, to simulate dynamic quantization. This function scaled the model's weights to 8-bit integers (int8), reducing memory usage and potentially improving processing speed.

Dynamic Quantization:

- In dynamic quantization, weights are scaled during inference to reduce computation complexity, which is particularly useful in resource-constrained environments.
- We scaled weights with an 8-bit precision, setting the scale factor to (2^7), to ensure a balance between compression and numerical stability.

5. Quantized Model Inference:

An inference function for the quantized model allowed us to make predictions using the compressed weights. This function scaled the input values accordingly, ensuring compatibility with the reduced-precision weights.

6. Evaluation of the Quantized Model

Finally, we evaluated the quantized model on the same metrics as the original model:

- Quantized Model Accuracy: We observed a slight reduction in accuracy compared to the original model, which is typical for quantization. However, the accuracy remained within an acceptable range, showing that the model could still perform digit classification effectively.
- Quantized Model Size: The quantized model size was significantly smaller than the original, indicating successful compression.
- Quantized Inference Time: The quantized model showed a decrease in inference time, confirming that quantization can improve processing speed in practical applications.

Comparison Results:

Metric	**Original Model**	**Quantized Model**	**Improvement**
Model Size(KB)	62.2255859375	8.5791015625	Yes
Inference Time(s)	2.0356178283691405e-05	2.3958683013916014e-05	No
Model Accuracy	0.9193	0.1985	Minimal loss

Conclusion:

We successfully applied dynamic quantization to a logistic regression model and observed the following:

1. Reduced Model Size: Quantization significantly compressed the model, which can save memory resources.
2. Improved Inference Speed: The quantized model executed predictions faster, benefiting applications where rapid inference is required.
3. Minimal Accuracy Loss: While there was a slight reduction in accuracy, the quantized model maintained acceptable performance.