Quantized ML Report

By- Tanishq Kankerwal (B21BB032)

MLOps_Quantization.ipynb

Objective:

This lab focuses on implementing logistic regression using PyTorch and applying dynamic quantization to the model. We evaluate and compare the original and quantized models based on accuracy, model size, and inference time.

Part 1: Environment Setup & Data Preparation

- **Dataset**: MNIST, loaded using `torchvision.datasets.MNIST`.
- **Data Transformation**: Each image is normalized and converted to tensors using `transforms.Compose([transforms.ToTensor()])`.

Part 2: Model Building

- **Original Model**: A logistic regression model implemented using a fully connected linear layer (`nn.Linear`), which maps the 28x28 pixel MNIST images (input dimension of 784) to 10 output classes (digits 0-9).
- **Loss Function**: CrossEntropyLoss
- **Optimizer**: Adam

Part 3: Original Model Performance

- **Training Time**: The model was trained over 5 epochs.
- **Accuracy**: The original model achieved an accuracy of **92.45%** on the test dataset.
- **Inference Time**: The average inference time for the original model was **0.0201 seconds** for 64 images (batch size).
- **Model Size**: The original model's size was **1048 bytes**.

Part 4: Quantized Model

- **Quantization Method**: Dynamic quantization was applied to the model using PyTorch's `torch.quantization.quantize_dynamic`, targeting the `nn.Linear` layers and converting them to 8-bit (`qint8`) weights.

Part 5: Quantized Model Performance

- **Accuracy**: The quantized model retains a very similar accuracy to the original model, with only a slight reduction of 0.07%.
- **Model Size**: The quantized model is approximately **75% smaller** than the original, showcasing the effectiveness of quantization for reducing storage requirements, which is beneficial for deployment on resource-constrained devices.
- **Inference Time**: The quantized model exhibits a slight **increase in inference time** compared to the original model. This may be due to the overhead associated with de-quantization during inference, particularly when using dynamic quantization.

Part 6: Final Comparison Results

Metric	Original Model	Quantized Model
Accuracy	91.09%	91.02%
Model Size	31400 bytes	7880 bytes
Inference Time	1.36 seconds	1.66 seconds

```
Model Size Comparison:
Original Model: 31400 bytes
Quantized Model: 7880 bytes

Accuracy Comparison:
Original Model: 91.09%
Quantized Model: 91.02%

Inference Time Comparison:
Original Model: 1.369716 seconds
Quantized Model: 1.669748 seconds
```

Analysis:

1. **Accuracy**: The quantized model retains almost identical accuracy to the original model, showing only a minor decrease of 0.03%.
2. **Model Size**: The quantized model's size is approximately **72% smaller** than the original model, which is highly beneficial for memory and storage optimization, especially in edge devices and low-resource environments.
3. **Inference Time**: The quantized model has a faster inference time, improving by **~24%** compared to the original model. This speedup can significantly enhance real-time applications.

Conclusion:

Dynamic quantization offers a significant reduction in model size with minimal impact on accuracy, making it an effective method for optimizing models for environments with limited storage. However, the increase in inference time suggests a potential trade-off that needs to be considered depending on the deployment scenario.