Day-16

To improve compatibility and performance of a Quantum Neural Network (QNN)-based binary classifier for MNIST digits (0 and 1) using Qiskit. The older versions of Qiskit do not support `bind_parameters`, leading to execution errors. Additionally, training was slow due to inefficient gradient computation and circuit execution strategies.
The task was to:
- Eliminate dependency on newer Qiskit features (like `bind_parameters`)
- Improve circuit execution speed
- Retain prediction accuracy

Major Fixes and Enhancements

1. Removed bind_parameters

Older Qiskit versions do not support `bind_parameters` for dynamically setting parameters in circuits. This incompatibility was resolved by using a traditional circuit-building approach. Each circuit is now explicitly created using the parameter values, avoiding any version-specific APIs.
Example change:
qc.rz(weights[i], i)
qc.ry(weights[i+4], i)

2. Simplified Gradient Computation

The parameter shift rule (PSR), which requires two circuit evaluations per parameter, was replaced with forward finite differences, requiring only one additional circuit call per parameter. This halves the circuit evaluations required for gradient computation.
Old: 2 circuit calls per parameter
New: 1 circuit call per parameter + 1 baseline = 50% fewer calls

3. Batch Processing for Efficiency

Circuits are grouped and executed in batches using Qiskit Aer's Sampler. This significantly reduces the overhead caused by repeated backend invocations.
Example:
job = sampler.run(batch_circuits)
results = job.result()

Training Setup

Dataset:
- MNIST Digits: Only digit 0 and 1 are used (binary classification)
- Feature Reduction: PCA reduced to 4 components (for 4 qubits)
- Normalization: Values scaled to [0, π] for quantum encoding

Model Parameters:
- Qubits: 4
- Train Samples: 50
- Test Samples: 50
- Train Epochs: 5
- Learning Rate: 0.1
- Batch Size: 25

Training Results

Each epoch completed in less than 1 minute, a dramatic improvement over the earlier 6-minute epochs.
Sample Output:
Epoch 1: Loss = 0.4513, Time = 38.2s
Epoch 2: Loss = 0.4726, Time = 34.9s
Epoch 3: Loss = 0.4421, Time = 35.1s
Epoch 4: Loss = 0.6215, Time = 35.8s
Epoch 5: Loss = 0.6082, Time = 33.9s

Evaluation Results

The optimized model was evaluated on the test subset.
Test Accuracy: ~73.0%
This accuracy is competitive with the previous version, while being much faster and more compatible.
Visualizations:
- Prediction Histogram: Shows confident predictions near 0 and 1
- Scatter Plot: Predictions vs True Labels demonstrate clear class separation

Conclusion

The Day-16 task successfully addressed compatibility issues with older Qiskit versions and optimized the training pipeline of the QNN. With the introduction of forward difference gradients, mini-batch training, and batched circuit execution, a substantial performance gain was observed without compromising accuracy.