Trail 0:

[INFO] Accuracy (no compression): 0.8766

[INFO] Accuracy (compression, no post-training): 0.8358

Trial 0 finished with value: 0.86752 and parameters: {'num\_layers': 8, 'num\_heads': 2, 'hidden\_size': 192, 'intermediate\_size': 768, 'linear\_layer\_choices': 'linear'}. Best is trial 0 with value: 0.86752.

[INFO] Accuracy (compression + post-training): 0.8675

Trail 1:

[INFO] Accuracy (no compression): 0.8776

[INFO] Accuracy (compression, no post-training): 0.7969

Trial 1 finished with value: 0.86624 and parameters: {'num\_layers': 4, 'num\_heads': 8, 'hidden\_size': 192, 'intermediate\_size': 2048, 'linear\_layer\_choices': 'linear'}. Best is trial 0 with value: 0.86752.

[INFO] Accuracy (compression + post-training): 0.8662

Trail 2:

[INFO] Accuracy (no compression): 0.8710

[INFO] Accuracy (compression, no post-training): 0.8529

Trial 2 finished with value: 0.86584 and parameters: {'num\_layers': 8, 'num\_heads': 8, 'hidden\_size': 128, 'intermediate\_size': 512, 'linear\_layer\_choices': 'identity'}. Best is trial 0 with value: 0.86752.

[INFO] Accuracy (compression + post-training): 0.8658

Trail 3:

[INFO] Accuracy (no compression): 0.8776

[INFO] Accuracy (compression, no post-training): 0.8507

Trial 3 finished with value: 0.85988 and parameters: {'num\_layers': 8, 'num\_heads': 16, 'hidden\_size': 384, 'intermediate\_size': 512, 'linear\_layer\_choices': 'linear'}. Best is trial 0 with value: 0.86752.

[INFO] Accuracy (compression + post-training): 0.8599

Trail 4:

[INFO] Accuracy (no compression): 0.8778

[INFO] Accuracy (compression, no post-training): 0.8094

Trial 4 finished with value: 0.8728 and parameters: {'num\_layers': 4, 'num\_heads': 4, 'hidden\_size': 128, 'intermediate\_size': 768, 'linear\_layer\_choices': 'linear'}. Best is trial 4 with value: 0.8728.

[INFO] Accuracy (compression + post-training): 0.8728

Trail 5:

[INFO] Accuracy (no compression): 0.8571

[INFO] Accuracy (compression, no post-training): 0.8557

Trial 5 finished with value: 0.87156 and parameters: {'num\_layers': 4, 'num\_heads': 4, 'hidden\_size': 512, 'intermediate\_size': 512, 'linear\_layer\_choices': 'identity'}. Best is trial 4 with value: 0.8728.

[INFO] Accuracy (compression + post-training): 0.8716

Trail 6:

[INFO] Accuracy (no compression): 0.8764

[INFO] Accuracy (compression, no post-training): 0.8600

Trial 6 finished with value: 0.85704 and parameters: {'num\_layers': 4, 'num\_heads': 2, 'hidden\_size': 512, 'intermediate\_size': 768, 'linear\_layer\_choices': 'linear'}. Best is trial 4 with value: 0.8728.

[INFO] Accuracy (compression + post-training): 0.8570

Trail 7:

[INFO] Accuracy (no compression): 0.8734

[INFO] Accuracy (compression, no post-training): 0.8242

Trial 7 finished with value: 0.86104 and parameters: {'num\_layers': 4, 'num\_heads': 2, 'hidden\_size': 384, 'intermediate\_size': 2048, 'linear\_layer\_choices': 'identity'}. Best is trial 4 with value: 0.8728.

[INFO] Accuracy (compression + post-training): 0.8610

Trail 8:

MISSING ACCURACY NO COMPRESSION

[INFO] Accuracy (compression, no post-training): 0.8086

Trial 8 finished with value: 0.85836 and parameters: {'num\_layers': 8, 'num\_heads': 16, 'hidden\_size': 512, 'intermediate\_size': 1024, 'linear\_layer\_choices': 'linear'}. Best is trial 4 with value: 0.8728.

[INFO] Accuracy (compression + post-training): 0.8584

Trail 9:

[INFO] Accuracy (no compression): 0.8740

[INFO] Accuracy (compression, no post-training): 0.8562

Trial 9 finished with value: 0.87172 and parameters: {'num\_layers': 4, 'num\_heads': 8, 'hidden\_size': 256, 'intermediate\_size': 768, 'linear\_layer\_choices': 'identity'}. Best is trial 4 with value: 0.8728.

[INFO] Accuracy (compression + post-training): 0.8717

**Question 6 Results:**  
Trial 0 finished with value: **0.87348** and parameters: {'bert.encoder.layer.0.attention.self.query\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.encoder.layer.0.attention.self.key\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.0.attention.self.key\_data\_in\_width': 16, 'bert.encoder.layer.0.attention.self.key\_data\_in\_frac\_width': 2, 'bert.encoder.layer.0.attention.self.key\_weight\_width': 8, 'bert.encoder.layer.0.attention.self.key\_weight\_frac\_width': 4, 'bert.encoder.layer.0.attention.self.key\_bias\_width': 32, 'bert.encoder.layer.0.attention.self.key\_bias\_frac\_width': 2, 'bert.encoder.layer.0.attention.self.value\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.encoder.layer.0.attention.output.dense\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.0.attention.output.dense\_data\_in\_width': 32, 'bert.encoder.layer.0.attention.output.dense\_data\_in\_frac\_width': 8, 'bert.encoder.layer.0.attention.output.dense\_weight\_width': 32, 'bert.encoder.layer.0.attention.output.dense\_weight\_frac\_width': 4, 'bert.encoder.layer.0.attention.output.dense\_bias\_width': 32, 'bert.encoder.layer.0.attention.output.dense\_bias\_frac\_width': 2, 'bert.encoder.layer.0.intermediate.dense\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.encoder.layer.0.output.dense\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.0.output.dense\_data\_in\_width': 32, 'bert.encoder.layer.0.output.dense\_data\_in\_frac\_width': 8, 'bert.encoder.layer.0.output.dense\_weight\_width': 16, 'bert.encoder.layer.0.output.dense\_weight\_frac\_width': 2, 'bert.encoder.layer.0.output.dense\_bias\_width': 32, 'bert.encoder.layer.0.output.dense\_bias\_frac\_width': 2, 'bert.encoder.layer.1.attention.self.query\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.1.attention.self.query\_data\_in\_width': 32, 'bert.encoder.layer.1.attention.self.query\_data\_in\_frac\_width': 8, 'bert.encoder.layer.1.attention.self.query\_weight\_width': 32, 'bert.encoder.layer.1.attention.self.query\_weight\_frac\_width': 8, 'bert.encoder.layer.1.attention.self.query\_bias\_width': 32, 'bert.encoder.layer.1.attention.self.query\_bias\_frac\_width': 2, 'bert.encoder.layer.1.attention.self.key\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.encoder.layer.1.attention.self.value\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.encoder.layer.1.attention.output.dense\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.encoder.layer.1.intermediate.dense\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.1.intermediate.dense\_data\_in\_width': 32, 'bert.encoder.layer.1.intermediate.dense\_data\_in\_frac\_width': 8, 'bert.encoder.layer.1.intermediate.dense\_weight\_width': 16, 'bert.encoder.layer.1.intermediate.dense\_weight\_frac\_width': 4, 'bert.encoder.layer.1.intermediate.dense\_bias\_width': 16, 'bert.encoder.layer.1.intermediate.dense\_bias\_frac\_width': 8, 'bert.encoder.layer.1.output.dense\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.1.output.dense\_data\_in\_width': 32, 'bert.encoder.layer.1.output.dense\_data\_in\_frac\_width': 8, 'bert.encoder.layer.1.output.dense\_weight\_width': 8, 'bert.encoder.layer.1.output.dense\_weight\_frac\_width': 8, 'bert.encoder.layer.1.output.dense\_bias\_width': 8, 'bert.encoder.layer.1.output.dense\_bias\_frac\_width': 8, 'bert.pooler.dense\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.pooler.dense\_data\_in\_width': 8, 'bert.pooler.dense\_data\_in\_frac\_width': 2, 'bert.pooler.dense\_weight\_width': 16, 'bert.pooler.dense\_weight\_frac\_width': 2, 'bert.pooler.dense\_bias\_width': 16, 'bert.pooler.dense\_bias\_frac\_width': 8, 'classifier\_type': <class 'torch.nn.modules.linear.Linear'>}. Best is trial 0 with value: 0.87348.

Trial 1 finished with value: **0.85444** and parameters: {'bert.encoder.layer.0.attention.self.query\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.0.attention.self.query\_data\_in\_width': 8, 'bert.encoder.layer.0.attention.self.query\_data\_in\_frac\_width': 8, 'bert.encoder.layer.0.attention.self.query\_weight\_width': 16, 'bert.encoder.layer.0.attention.self.query\_weight\_frac\_width': 8, 'bert.encoder.layer.0.attention.self.query\_bias\_width': 32, 'bert.encoder.layer.0.attention.self.query\_bias\_frac\_width': 4, 'bert.encoder.layer.0.attention.self.key\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.0.attention.self.key\_data\_in\_width': 16, 'bert.encoder.layer.0.attention.self.key\_data\_in\_frac\_width': 8, 'bert.encoder.layer.0.attention.self.key\_weight\_width': 8, 'bert.encoder.layer.0.attention.self.key\_weight\_frac\_width': 2, 'bert.encoder.layer.0.attention.self.key\_bias\_width': 16, 'bert.encoder.layer.0.attention.self.key\_bias\_frac\_width': 4, 'bert.encoder.layer.0.attention.self.value\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.0.attention.self.value\_data\_in\_width': 16, 'bert.encoder.layer.0.attention.self.value\_data\_in\_frac\_width': 4, 'bert.encoder.layer.0.attention.self.value\_weight\_width': 16, 'bert.encoder.layer.0.attention.self.value\_weight\_frac\_width': 2, 'bert.encoder.layer.0.attention.self.value\_bias\_width': 32, 'bert.encoder.layer.0.attention.self.value\_bias\_frac\_width': 4, 'bert.encoder.layer.0.attention.output.dense\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.encoder.layer.0.intermediate.dense\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.0.intermediate.dense\_data\_in\_width': 8, 'bert.encoder.layer.0.intermediate.dense\_data\_in\_frac\_width': 4, 'bert.encoder.layer.0.intermediate.dense\_weight\_width': 8, 'bert.encoder.layer.0.intermediate.dense\_weight\_frac\_width': 4, 'bert.encoder.layer.0.intermediate.dense\_bias\_width': 32, 'bert.encoder.layer.0.intermediate.dense\_bias\_frac\_width': 8, 'bert.encoder.layer.0.output.dense\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.0.output.dense\_data\_in\_width': 8, 'bert.encoder.layer.0.output.dense\_data\_in\_frac\_width': 2, 'bert.encoder.layer.0.output.dense\_weight\_width': 32, 'bert.encoder.layer.0.output.dense\_weight\_frac\_width': 8, 'bert.encoder.layer.0.output.dense\_bias\_width': 8, 'bert.encoder.layer.0.output.dense\_bias\_frac\_width': 8, 'bert.encoder.layer.1.attention.self.query\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.1.attention.self.query\_data\_in\_width': 32, 'bert.encoder.layer.1.attention.self.query\_data\_in\_frac\_width': 8, 'bert.encoder.layer.1.attention.self.query\_weight\_width': 16, 'bert.encoder.layer.1.attention.self.query\_weight\_frac\_width': 4, 'bert.encoder.layer.1.attention.self.query\_bias\_width': 16, 'bert.encoder.layer.1.attention.self.query\_bias\_frac\_width': 4, 'bert.encoder.layer.1.attention.self.key\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.encoder.layer.1.attention.self.value\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.1.attention.self.value\_data\_in\_width': 16, 'bert.encoder.layer.1.attention.self.value\_data\_in\_frac\_width': 4, 'bert.encoder.layer.1.attention.self.value\_weight\_width': 32, 'bert.encoder.layer.1.attention.self.value\_weight\_frac\_width': 2, 'bert.encoder.layer.1.attention.self.value\_bias\_width': 16, 'bert.encoder.layer.1.attention.self.value\_bias\_frac\_width': 8, 'bert.encoder.layer.1.attention.output.dense\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.1.attention.output.dense\_data\_in\_width': 16, 'bert.encoder.layer.1.attention.output.dense\_data\_in\_frac\_width': 4, 'bert.encoder.layer.1.attention.output.dense\_weight\_width': 32, 'bert.encoder.layer.1.attention.output.dense\_weight\_frac\_width': 8, 'bert.encoder.layer.1.attention.output.dense\_bias\_width': 32, 'bert.encoder.layer.1.attention.output.dense\_bias\_frac\_width': 4, 'bert.encoder.layer.1.intermediate.dense\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.1.intermediate.dense\_data\_in\_width': 16, 'bert.encoder.layer.1.intermediate.dense\_data\_in\_frac\_width': 2, 'bert.encoder.layer.1.intermediate.dense\_weight\_width': 8, 'bert.encoder.layer.1.intermediate.dense\_weight\_frac\_width': 8, 'bert.encoder.layer.1.intermediate.dense\_bias\_width': 32, 'bert.encoder.layer.1.intermediate.dense\_bias\_frac\_width': 8, 'bert.encoder.layer.1.output.dense\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.pooler.dense\_type': <class 'torch.nn.modules.linear.Linear'>, 'classifier\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'classifier\_data\_in\_width': 8, 'classifier\_data\_in\_frac\_width': 8, 'classifier\_weight\_width': 32, 'classifier\_weight\_frac\_width': 4, 'classifier\_bias\_width': 8, 'classifier\_bias\_frac\_width': 4}. Best is trial 0 with value: 0.87348.

Trial 2 finished with value: **0.87484** and parameters: {'bert.encoder.layer.0.attention.self.query\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.0.attention.self.query\_data\_in\_width': 32, 'bert.encoder.layer.0.attention.self.query\_data\_in\_frac\_width': 2, 'bert.encoder.layer.0.attention.self.query\_weight\_width': 16, 'bert.encoder.layer.0.attention.self.query\_weight\_frac\_width': 8, 'bert.encoder.layer.0.attention.self.query\_bias\_width': 8, 'bert.encoder.layer.0.attention.self.query\_bias\_frac\_width': 8, 'bert.encoder.layer.0.attention.self.key\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.encoder.layer.0.attention.self.value\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.encoder.layer.0.attention.output.dense\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.encoder.layer.0.intermediate.dense\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.0.intermediate.dense\_data\_in\_width': 16, 'bert.encoder.layer.0.intermediate.dense\_data\_in\_frac\_width': 2, 'bert.encoder.layer.0.intermediate.dense\_weight\_width': 8, 'bert.encoder.layer.0.intermediate.dense\_weight\_frac\_width': 8, 'bert.encoder.layer.0.intermediate.dense\_bias\_width': 16, 'bert.encoder.layer.0.intermediate.dense\_bias\_frac\_width': 4, 'bert.encoder.layer.0.output.dense\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.encoder.layer.1.attention.self.query\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.1.attention.self.query\_data\_in\_width': 32, 'bert.encoder.layer.1.attention.self.query\_data\_in\_frac\_width': 8, 'bert.encoder.layer.1.attention.self.query\_weight\_width': 16, 'bert.encoder.layer.1.attention.self.query\_weight\_frac\_width': 2, 'bert.encoder.layer.1.attention.self.query\_bias\_width': 16, 'bert.encoder.layer.1.attention.self.query\_bias\_frac\_width': 2, 'bert.encoder.layer.1.attention.self.key\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.1.attention.self.key\_data\_in\_width': 16, 'bert.encoder.layer.1.attention.self.key\_data\_in\_frac\_width': 4, 'bert.encoder.layer.1.attention.self.key\_weight\_width': 32, 'bert.encoder.layer.1.attention.self.key\_weight\_frac\_width': 2, 'bert.encoder.layer.1.attention.self.key\_bias\_width': 32, 'bert.encoder.layer.1.attention.self.key\_bias\_frac\_width': 4, 'bert.encoder.layer.1.attention.self.value\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.1.attention.self.value\_data\_in\_width': 16, 'bert.encoder.layer.1.attention.self.value\_data\_in\_frac\_width': 4, 'bert.encoder.layer.1.attention.self.value\_weight\_width': 16, 'bert.encoder.layer.1.attention.self.value\_weight\_frac\_width': 4, 'bert.encoder.layer.1.attention.self.value\_bias\_width': 32, 'bert.encoder.layer.1.attention.self.value\_bias\_frac\_width': 8, 'bert.encoder.layer.1.attention.output.dense\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.encoder.layer.1.intermediate.dense\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.1.intermediate.dense\_data\_in\_width': 8, 'bert.encoder.layer.1.intermediate.dense\_data\_in\_frac\_width': 2, 'bert.encoder.layer.1.intermediate.dense\_weight\_width': 16, 'bert.encoder.layer.1.intermediate.dense\_weight\_frac\_width': 8, 'bert.encoder.layer.1.intermediate.dense\_bias\_width': 16, 'bert.encoder.layer.1.intermediate.dense\_bias\_frac\_width': 8, 'bert.encoder.layer.1.output.dense\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.1.output.dense\_data\_in\_width': 16, 'bert.encoder.layer.1.output.dense\_data\_in\_frac\_width': 2, 'bert.encoder.layer.1.output.dense\_weight\_width': 16, 'bert.encoder.layer.1.output.dense\_weight\_frac\_width': 8, 'bert.encoder.layer.1.output.dense\_bias\_width': 16, 'bert.encoder.layer.1.output.dense\_bias\_frac\_width': 4, 'bert.pooler.dense\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.pooler.dense\_data\_in\_width': 16, 'bert.pooler.dense\_data\_in\_frac\_width': 8, 'bert.pooler.dense\_weight\_width': 16, 'bert.pooler.dense\_weight\_frac\_width': 2, 'bert.pooler.dense\_bias\_width': 16, 'bert.pooler.dense\_bias\_frac\_width': 2, 'classifier\_type': <class 'torch.nn.modules.linear.Linear'>}. Best is trial 2 with value: 0.87484.

Trial 3 finished with value: **0.8642** and parameters: {'bert.encoder.layer.0.attention.self.query\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.0.attention.self.query\_data\_in\_width': 8, 'bert.encoder.layer.0.attention.self.query\_data\_in\_frac\_width': 2, 'bert.encoder.layer.0.attention.self.query\_weight\_width': 8, 'bert.encoder.layer.0.attention.self.query\_weight\_frac\_width': 4, 'bert.encoder.layer.0.attention.self.query\_bias\_width': 16, 'bert.encoder.layer.0.attention.self.query\_bias\_frac\_width': 2, 'bert.encoder.layer.0.attention.self.key\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.0.attention.self.key\_data\_in\_width': 16, 'bert.encoder.layer.0.attention.self.key\_data\_in\_frac\_width': 2, 'bert.encoder.layer.0.attention.self.key\_weight\_width': 16, 'bert.encoder.layer.0.attention.self.key\_weight\_frac\_width': 4, 'bert.encoder.layer.0.attention.self.key\_bias\_width': 32, 'bert.encoder.layer.0.attention.self.key\_bias\_frac\_width': 8, 'bert.encoder.layer.0.attention.self.value\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.0.attention.self.value\_data\_in\_width': 16, 'bert.encoder.layer.0.attention.self.value\_data\_in\_frac\_width': 2, 'bert.encoder.layer.0.attention.self.value\_weight\_width': 8, 'bert.encoder.layer.0.attention.self.value\_weight\_frac\_width': 8, 'bert.encoder.layer.0.attention.self.value\_bias\_width': 32, 'bert.encoder.layer.0.attention.self.value\_bias\_frac\_width': 2, 'bert.encoder.layer.0.attention.output.dense\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.encoder.layer.0.intermediate.dense\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.0.intermediate.dense\_data\_in\_width': 8, 'bert.encoder.layer.0.intermediate.dense\_data\_in\_frac\_width': 4, 'bert.encoder.layer.0.intermediate.dense\_weight\_width': 8, 'bert.encoder.layer.0.intermediate.dense\_weight\_frac\_width': 8, 'bert.encoder.layer.0.intermediate.dense\_bias\_width': 8, 'bert.encoder.layer.0.intermediate.dense\_bias\_frac\_width': 8, 'bert.encoder.layer.0.output.dense\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.0.output.dense\_data\_in\_width': 16, 'bert.encoder.layer.0.output.dense\_data\_in\_frac\_width': 8, 'bert.encoder.layer.0.output.dense\_weight\_width': 32, 'bert.encoder.layer.0.output.dense\_weight\_frac\_width': 8, 'bert.encoder.layer.0.output.dense\_bias\_width': 8, 'bert.encoder.layer.0.output.dense\_bias\_frac\_width': 2, 'bert.encoder.layer.1.attention.self.query\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.encoder.layer.1.attention.self.key\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.1.attention.self.key\_data\_in\_width': 16, 'bert.encoder.layer.1.attention.self.key\_data\_in\_frac\_width': 8, 'bert.encoder.layer.1.attention.self.key\_weight\_width': 16, 'bert.encoder.layer.1.attention.self.key\_weight\_frac\_width': 8, 'bert.encoder.layer.1.attention.self.key\_bias\_width': 8, 'bert.encoder.layer.1.attention.self.key\_bias\_frac\_width': 4, 'bert.encoder.layer.1.attention.self.value\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.1.attention.self.value\_data\_in\_width': 8, 'bert.encoder.layer.1.attention.self.value\_data\_in\_frac\_width': 2, 'bert.encoder.layer.1.attention.self.value\_weight\_width': 8, 'bert.encoder.layer.1.attention.self.value\_weight\_frac\_width': 8, 'bert.encoder.layer.1.attention.self.value\_bias\_width': 16, 'bert.encoder.layer.1.attention.self.value\_bias\_frac\_width': 4, 'bert.encoder.layer.1.attention.output.dense\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.1.attention.output.dense\_data\_in\_width': 8, 'bert.encoder.layer.1.attention.output.dense\_data\_in\_frac\_width': 4, 'bert.encoder.layer.1.attention.output.dense\_weight\_width': 16, 'bert.encoder.layer.1.attention.output.dense\_weight\_frac\_width': 8, 'bert.encoder.layer.1.attention.output.dense\_bias\_width': 16, 'bert.encoder.layer.1.attention.output.dense\_bias\_frac\_width': 8, 'bert.encoder.layer.1.intermediate.dense\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.1.intermediate.dense\_data\_in\_width': 8, 'bert.encoder.layer.1.intermediate.dense\_data\_in\_frac\_width': 2, 'bert.encoder.layer.1.intermediate.dense\_weight\_width': 16, 'bert.encoder.layer.1.intermediate.dense\_weight\_frac\_width': 8, 'bert.encoder.layer.1.intermediate.dense\_bias\_width': 16, 'bert.encoder.layer.1.intermediate.dense\_bias\_frac\_width': 4, 'bert.encoder.layer.1.output.dense\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.pooler.dense\_type': <class 'torch.nn.modules.linear.Linear'>, 'classifier\_type': <class 'torch.nn.modules.linear.Linear'>}. Best is trial 2 with value: 0.87484.

Trial 4 finished with value: **0.8636** and parameters: {'bert.encoder.layer.0.attention.self.query\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.0.attention.self.query\_data\_in\_width': 32, 'bert.encoder.layer.0.attention.self.query\_data\_in\_frac\_width': 2, 'bert.encoder.layer.0.attention.self.query\_weight\_width': 32, 'bert.encoder.layer.0.attention.self.query\_weight\_frac\_width': 2, 'bert.encoder.layer.0.attention.self.query\_bias\_width': 32, 'bert.encoder.layer.0.attention.self.query\_bias\_frac\_width': 8, 'bert.encoder.layer.0.attention.self.key\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.encoder.layer.0.attention.self.value\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.encoder.layer.0.attention.output.dense\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.encoder.layer.0.intermediate.dense\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.encoder.layer.0.output.dense\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.encoder.layer.1.attention.self.query\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.1.attention.self.query\_data\_in\_width': 16, 'bert.encoder.layer.1.attention.self.query\_data\_in\_frac\_width': 8, 'bert.encoder.layer.1.attention.self.query\_weight\_width': 8, 'bert.encoder.layer.1.attention.self.query\_weight\_frac\_width': 8, 'bert.encoder.layer.1.attention.self.query\_bias\_width': 32, 'bert.encoder.layer.1.attention.self.query\_bias\_frac\_width': 8, 'bert.encoder.layer.1.attention.self.key\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.encoder.layer.1.attention.self.value\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.1.attention.self.value\_data\_in\_width': 8, 'bert.encoder.layer.1.attention.self.value\_data\_in\_frac\_width': 2, 'bert.encoder.layer.1.attention.self.value\_weight\_width': 8, 'bert.encoder.layer.1.attention.self.value\_weight\_frac\_width': 8, 'bert.encoder.layer.1.attention.self.value\_bias\_width': 16, 'bert.encoder.layer.1.attention.self.value\_bias\_frac\_width': 8, 'bert.encoder.layer.1.attention.output.dense\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.encoder.layer.1.intermediate.dense\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.1.intermediate.dense\_data\_in\_width': 32, 'bert.encoder.layer.1.intermediate.dense\_data\_in\_frac\_width': 4, 'bert.encoder.layer.1.intermediate.dense\_weight\_width': 32, 'bert.encoder.layer.1.intermediate.dense\_weight\_frac\_width': 4, 'bert.encoder.layer.1.intermediate.dense\_bias\_width': 16, 'bert.encoder.layer.1.intermediate.dense\_bias\_frac\_width': 8, 'bert.encoder.layer.1.output.dense\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.1.output.dense\_data\_in\_width': 8, 'bert.encoder.layer.1.output.dense\_data\_in\_frac\_width': 4, 'bert.encoder.layer.1.output.dense\_weight\_width': 8, 'bert.encoder.layer.1.output.dense\_weight\_frac\_width': 8, 'bert.encoder.layer.1.output.dense\_bias\_width': 32, 'bert.encoder.layer.1.output.dense\_bias\_frac\_width': 2, 'bert.pooler.dense\_type': <class 'torch.nn.modules.linear.Linear'>, 'classifier\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'classifier\_data\_in\_width': 8, 'classifier\_data\_in\_frac\_width': 8, 'classifier\_weight\_width': 16, 'classifier\_weight\_frac\_width': 8, 'classifier\_bias\_width': 16, 'classifier\_bias\_frac\_width': 4}. Best is trial 2 with value: 0.87484.

Trial 5 finished with value: **0.85048** and parameters: {'bert.encoder.layer.0.attention.self.query\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.encoder.layer.0.attention.self.key\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.0.attention.self.key\_data\_in\_width': 32, 'bert.encoder.layer.0.attention.self.key\_data\_in\_frac\_width': 2, 'bert.encoder.layer.0.attention.self.key\_weight\_width': 16, 'bert.encoder.layer.0.attention.self.key\_weight\_frac\_width': 8, 'bert.encoder.layer.0.attention.self.key\_bias\_width': 32, 'bert.encoder.layer.0.attention.self.key\_bias\_frac\_width': 8, 'bert.encoder.layer.0.attention.self.value\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.0.attention.self.value\_data\_in\_width': 8, 'bert.encoder.layer.0.attention.self.value\_data\_in\_frac\_width': 2, 'bert.encoder.layer.0.attention.self.value\_weight\_width': 16, 'bert.encoder.layer.0.attention.self.value\_weight\_frac\_width': 2, 'bert.encoder.layer.0.attention.self.value\_bias\_width': 16, 'bert.encoder.layer.0.attention.self.value\_bias\_frac\_width': 2, 'bert.encoder.layer.0.attention.output.dense\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.0.attention.output.dense\_data\_in\_width': 16, 'bert.encoder.layer.0.attention.output.dense\_data\_in\_frac\_width': 8, 'bert.encoder.layer.0.attention.output.dense\_weight\_width': 16, 'bert.encoder.layer.0.attention.output.dense\_weight\_frac\_width': 2, 'bert.encoder.layer.0.attention.output.dense\_bias\_width': 16, 'bert.encoder.layer.0.attention.output.dense\_bias\_frac\_width': 8, 'bert.encoder.layer.0.intermediate.dense\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.encoder.layer.0.output.dense\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.encoder.layer.1.attention.self.query\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.1.attention.self.query\_data\_in\_width': 8, 'bert.encoder.layer.1.attention.self.query\_data\_in\_frac\_width': 8, 'bert.encoder.layer.1.attention.self.query\_weight\_width': 16, 'bert.encoder.layer.1.attention.self.query\_weight\_frac\_width': 2, 'bert.encoder.layer.1.attention.self.query\_bias\_width': 16, 'bert.encoder.layer.1.attention.self.query\_bias\_frac\_width': 8, 'bert.encoder.layer.1.attention.self.key\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.1.attention.self.key\_data\_in\_width': 32, 'bert.encoder.layer.1.attention.self.key\_data\_in\_frac\_width': 4, 'bert.encoder.layer.1.attention.self.key\_weight\_width': 16, 'bert.encoder.layer.1.attention.self.key\_weight\_frac\_width': 8, 'bert.encoder.layer.1.attention.self.key\_bias\_width': 32, 'bert.encoder.layer.1.attention.self.key\_bias\_frac\_width': 4, 'bert.encoder.layer.1.attention.self.value\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.1.attention.self.value\_data\_in\_width': 8, 'bert.encoder.layer.1.attention.self.value\_data\_in\_frac\_width': 4, 'bert.encoder.layer.1.attention.self.value\_weight\_width': 16, 'bert.encoder.layer.1.attention.self.value\_weight\_frac\_width': 2, 'bert.encoder.layer.1.attention.self.value\_bias\_width': 16, 'bert.encoder.layer.1.attention.self.value\_bias\_frac\_width': 4, 'bert.encoder.layer.1.attention.output.dense\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.1.attention.output.dense\_data\_in\_width': 32, 'bert.encoder.layer.1.attention.output.dense\_data\_in\_frac\_width': 2, 'bert.encoder.layer.1.attention.output.dense\_weight\_width': 8, 'bert.encoder.layer.1.attention.output.dense\_weight\_frac\_width': 4, 'bert.encoder.layer.1.attention.output.dense\_bias\_width': 16, 'bert.encoder.layer.1.attention.output.dense\_bias\_frac\_width': 8, 'bert.encoder.layer.1.intermediate.dense\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.1.intermediate.dense\_data\_in\_width': 16, 'bert.encoder.layer.1.intermediate.dense\_data\_in\_frac\_width': 8, 'bert.encoder.layer.1.intermediate.dense\_weight\_width': 16, 'bert.encoder.layer.1.intermediate.dense\_weight\_frac\_width': 4, 'bert.encoder.layer.1.intermediate.dense\_bias\_width': 16, 'bert.encoder.layer.1.intermediate.dense\_bias\_frac\_width': 8, 'bert.encoder.layer.1.output.dense\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.1.output.dense\_data\_in\_width': 32, 'bert.encoder.layer.1.output.dense\_data\_in\_frac\_width': 8, 'bert.encoder.layer.1.output.dense\_weight\_width': 8, 'bert.encoder.layer.1.output.dense\_weight\_frac\_width': 4, 'bert.encoder.layer.1.output.dense\_bias\_width': 8, 'bert.encoder.layer.1.output.dense\_bias\_frac\_width': 4, 'bert.pooler.dense\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.pooler.dense\_data\_in\_width': 32, 'bert.pooler.dense\_data\_in\_frac\_width': 8, 'bert.pooler.dense\_weight\_width': 32, 'bert.pooler.dense\_weight\_frac\_width': 4, 'bert.pooler.dense\_bias\_width': 32, 'bert.pooler.dense\_bias\_frac\_width': 2, 'classifier\_type': <class 'torch.nn.modules.linear.Linear'>}. Best is trial 2 with value: 0.87484.

Trial 6 finished with value: **0.8592** and parameters: {'bert.encoder.layer.0.attention.self.query\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.encoder.layer.0.attention.self.key\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.0.attention.self.key\_data\_in\_width': 32, 'bert.encoder.layer.0.attention.self.key\_data\_in\_frac\_width': 2, 'bert.encoder.layer.0.attention.self.key\_weight\_width': 32, 'bert.encoder.layer.0.attention.self.key\_weight\_frac\_width': 8, 'bert.encoder.layer.0.attention.self.key\_bias\_width': 16, 'bert.encoder.layer.0.attention.self.key\_bias\_frac\_width': 4, 'bert.encoder.layer.0.attention.self.value\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.0.attention.self.value\_data\_in\_width': 8, 'bert.encoder.layer.0.attention.self.value\_data\_in\_frac\_width': 4, 'bert.encoder.layer.0.attention.self.value\_weight\_width': 32, 'bert.encoder.layer.0.attention.self.value\_weight\_frac\_width': 2, 'bert.encoder.layer.0.attention.self.value\_bias\_width': 16, 'bert.encoder.layer.0.attention.self.value\_bias\_frac\_width': 4, 'bert.encoder.layer.0.attention.output.dense\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.0.attention.output.dense\_data\_in\_width': 32, 'bert.encoder.layer.0.attention.output.dense\_data\_in\_frac\_width': 2, 'bert.encoder.layer.0.attention.output.dense\_weight\_width': 16, 'bert.encoder.layer.0.attention.output.dense\_weight\_frac\_width': 8, 'bert.encoder.layer.0.attention.output.dense\_bias\_width': 16, 'bert.encoder.layer.0.attention.output.dense\_bias\_frac\_width': 4, 'bert.encoder.layer.0.intermediate.dense\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.0.intermediate.dense\_data\_in\_width': 16, 'bert.encoder.layer.0.intermediate.dense\_data\_in\_frac\_width': 4, 'bert.encoder.layer.0.intermediate.dense\_weight\_width': 16, 'bert.encoder.layer.0.intermediate.dense\_weight\_frac\_width': 8, 'bert.encoder.layer.0.intermediate.dense\_bias\_width': 16, 'bert.encoder.layer.0.intermediate.dense\_bias\_frac\_width': 8, 'bert.encoder.layer.0.output.dense\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.encoder.layer.1.attention.self.query\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.encoder.layer.1.attention.self.key\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.encoder.layer.1.attention.self.value\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.encoder.layer.1.attention.output.dense\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.encoder.layer.1.intermediate.dense\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.encoder.layer.1.output.dense\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.1.output.dense\_data\_in\_width': 8, 'bert.encoder.layer.1.output.dense\_data\_in\_frac\_width': 4, 'bert.encoder.layer.1.output.dense\_weight\_width': 16, 'bert.encoder.layer.1.output.dense\_weight\_frac\_width': 2, 'bert.encoder.layer.1.output.dense\_bias\_width': 32, 'bert.encoder.layer.1.output.dense\_bias\_frac\_width': 2, 'bert.pooler.dense\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.pooler.dense\_data\_in\_width': 8, 'bert.pooler.dense\_data\_in\_frac\_width': 4, 'bert.pooler.dense\_weight\_width': 8, 'bert.pooler.dense\_weight\_frac\_width': 2, 'bert.pooler.dense\_bias\_width': 16, 'bert.pooler.dense\_bias\_frac\_width': 4, 'classifier\_type': <class 'torch.nn.modules.linear.Linear'>}. Best is trial 2 with value: 0.87484.

Trial 7 finished with value: **0.86852** and parameters: {'bert.encoder.layer.0.attention.self.query\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.encoder.layer.0.attention.self.key\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.encoder.layer.0.attention.self.value\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.encoder.layer.0.attention.output.dense\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.encoder.layer.0.intermediate.dense\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.0.intermediate.dense\_data\_in\_width': 16, 'bert.encoder.layer.0.intermediate.dense\_data\_in\_frac\_width': 8, 'bert.encoder.layer.0.intermediate.dense\_weight\_width': 8, 'bert.encoder.layer.0.intermediate.dense\_weight\_frac\_width': 2, 'bert.encoder.layer.0.intermediate.dense\_bias\_width': 16, 'bert.encoder.layer.0.intermediate.dense\_bias\_frac\_width': 2, 'bert.encoder.layer.0.output.dense\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.0.output.dense\_data\_in\_width': 8, 'bert.encoder.layer.0.output.dense\_data\_in\_frac\_width': 2, 'bert.encoder.layer.0.output.dense\_weight\_width': 16, 'bert.encoder.layer.0.output.dense\_weight\_frac\_width': 8, 'bert.encoder.layer.0.output.dense\_bias\_width': 16, 'bert.encoder.layer.0.output.dense\_bias\_frac\_width': 8, 'bert.encoder.layer.1.attention.self.query\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.encoder.layer.1.attention.self.key\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.encoder.layer.1.attention.self.value\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.encoder.layer.1.attention.output.dense\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.encoder.layer.1.intermediate.dense\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.encoder.layer.1.output.dense\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.pooler.dense\_type': <class 'torch.nn.modules.linear.Linear'>, 'classifier\_type': <class 'torch.nn.modules.linear.Linear'>}. Best is trial 2 with value: 0.87484.

Trial 8 finished with value: **0.8678** and parameters: {'bert.encoder.layer.0.attention.self.query\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.encoder.layer.0.attention.self.key\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.0.attention.self.key\_data\_in\_width': 16, 'bert.encoder.layer.0.attention.self.key\_data\_in\_frac\_width': 8, 'bert.encoder.layer.0.attention.self.key\_weight\_width': 32, 'bert.encoder.layer.0.attention.self.key\_weight\_frac\_width': 8, 'bert.encoder.layer.0.attention.self.key\_bias\_width': 16, 'bert.encoder.layer.0.attention.self.key\_bias\_frac\_width': 2, 'bert.encoder.layer.0.attention.self.value\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.encoder.layer.0.attention.output.dense\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.encoder.layer.0.intermediate.dense\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.0.intermediate.dense\_data\_in\_width': 8, 'bert.encoder.layer.0.intermediate.dense\_data\_in\_frac\_width': 4, 'bert.encoder.layer.0.intermediate.dense\_weight\_width': 32, 'bert.encoder.layer.0.intermediate.dense\_weight\_frac\_width': 2, 'bert.encoder.layer.0.intermediate.dense\_bias\_width': 8, 'bert.encoder.layer.0.intermediate.dense\_bias\_frac\_width': 8, 'bert.encoder.layer.0.output.dense\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.0.output.dense\_data\_in\_width': 8, 'bert.encoder.layer.0.output.dense\_data\_in\_frac\_width': 2, 'bert.encoder.layer.0.output.dense\_weight\_width': 16, 'bert.encoder.layer.0.output.dense\_weight\_frac\_width': 2, 'bert.encoder.layer.0.output.dense\_bias\_width': 8, 'bert.encoder.layer.0.output.dense\_bias\_frac\_width': 4, 'bert.encoder.layer.1.attention.self.query\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.1.attention.self.query\_data\_in\_width': 8, 'bert.encoder.layer.1.attention.self.query\_data\_in\_frac\_width': 8, 'bert.encoder.layer.1.attention.self.query\_weight\_width': 16, 'bert.encoder.layer.1.attention.self.query\_weight\_frac\_width': 8, 'bert.encoder.layer.1.attention.self.query\_bias\_width': 16, 'bert.encoder.layer.1.attention.self.query\_bias\_frac\_width': 4, 'bert.encoder.layer.1.attention.self.key\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.1.attention.self.key\_data\_in\_width': 8, 'bert.encoder.layer.1.attention.self.key\_data\_in\_frac\_width': 4, 'bert.encoder.layer.1.attention.self.key\_weight\_width': 32, 'bert.encoder.layer.1.attention.self.key\_weight\_frac\_width': 8, 'bert.encoder.layer.1.attention.self.key\_bias\_width': 16, 'bert.encoder.layer.1.attention.self.key\_bias\_frac\_width': 2, 'bert.encoder.layer.1.attention.self.value\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.1.attention.self.value\_data\_in\_width': 16, 'bert.encoder.layer.1.attention.self.value\_data\_in\_frac\_width': 8, 'bert.encoder.layer.1.attention.self.value\_weight\_width': 16, 'bert.encoder.layer.1.attention.self.value\_weight\_frac\_width': 8, 'bert.encoder.layer.1.attention.self.value\_bias\_width': 16, 'bert.encoder.layer.1.attention.self.value\_bias\_frac\_width': 2, 'bert.encoder.layer.1.attention.output.dense\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.1.attention.output.dense\_data\_in\_width': 8, 'bert.encoder.layer.1.attention.output.dense\_data\_in\_frac\_width': 4, 'bert.encoder.layer.1.attention.output.dense\_weight\_width': 8, 'bert.encoder.layer.1.attention.output.dense\_weight\_frac\_width': 8, 'bert.encoder.layer.1.attention.output.dense\_bias\_width': 32, 'bert.encoder.layer.1.attention.output.dense\_bias\_frac\_width': 8, 'bert.encoder.layer.1.intermediate.dense\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.encoder.layer.1.output.dense\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.pooler.dense\_type': <class 'torch.nn.modules.linear.Linear'>, 'classifier\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'classifier\_data\_in\_width': 32, 'classifier\_data\_in\_frac\_width': 8, 'classifier\_weight\_width': 32, 'classifier\_weight\_frac\_width': 4, 'classifier\_bias\_width': 32, 'classifier\_bias\_frac\_width': 2}. Best is trial 2 with value: 0.87484.

Trial 9 finished with value: **0.87176** and parameters: {'bert.encoder.layer.0.attention.self.query\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.0.attention.self.query\_data\_in\_width': 8, 'bert.encoder.layer.0.attention.self.query\_data\_in\_frac\_width': 4, 'bert.encoder.layer.0.attention.self.query\_weight\_width': 16, 'bert.encoder.layer.0.attention.self.query\_weight\_frac\_width': 8, 'bert.encoder.layer.0.attention.self.query\_bias\_width': 32, 'bert.encoder.layer.0.attention.self.query\_bias\_frac\_width': 8, 'bert.encoder.layer.0.attention.self.key\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.encoder.layer.0.attention.self.value\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.encoder.layer.0.attention.output.dense\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.encoder.layer.0.intermediate.dense\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.encoder.layer.0.output.dense\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.encoder.layer.1.attention.self.query\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.encoder.layer.1.attention.self.key\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.encoder.layer.1.attention.self.value\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.1.attention.self.value\_data\_in\_width': 8, 'bert.encoder.layer.1.attention.self.value\_data\_in\_frac\_width': 2, 'bert.encoder.layer.1.attention.self.value\_weight\_width': 8, 'bert.encoder.layer.1.attention.self.value\_weight\_frac\_width': 4, 'bert.encoder.layer.1.attention.self.value\_bias\_width': 32, 'bert.encoder.layer.1.attention.self.value\_bias\_frac\_width': 4, 'bert.encoder.layer.1.attention.output.dense\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.1.attention.output.dense\_data\_in\_width': 32, 'bert.encoder.layer.1.attention.output.dense\_data\_in\_frac\_width': 4, 'bert.encoder.layer.1.attention.output.dense\_weight\_width': 8, 'bert.encoder.layer.1.attention.output.dense\_weight\_frac\_width': 2, 'bert.encoder.layer.1.attention.output.dense\_bias\_width': 8, 'bert.encoder.layer.1.attention.output.dense\_bias\_frac\_width': 4, 'bert.encoder.layer.1.intermediate.dense\_type': <class 'torch.nn.modules.linear.Linear'>, 'bert.encoder.layer.1.output.dense\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'bert.encoder.layer.1.output.dense\_data\_in\_width': 8, 'bert.encoder.layer.1.output.dense\_data\_in\_frac\_width': 4, 'bert.encoder.layer.1.output.dense\_weight\_width': 16, 'bert.encoder.layer.1.output.dense\_weight\_frac\_width': 8, 'bert.encoder.layer.1.output.dense\_bias\_width': 16, 'bert.encoder.layer.1.output.dense\_bias\_frac\_width': 2, 'bert.pooler.dense\_type': <class 'torch.nn.modules.linear.Linear'>, 'classifier\_type': <class 'chop.nn.quantized.modules.linear.LinearInteger'>, 'classifier\_data\_in\_width': 32, 'classifier\_data\_in\_frac\_width': 4, 'classifier\_weight\_width': 8, 'classifier\_weight\_frac\_width': 8, 'classifier\_bias\_width': 16, 'classifier\_bias\_frac\_width': 2}. Best is trial 2 with value: 0.87484.

Shape:

Input: (∗,in\_features)(∗,in\_features) where *\** means any number of additional dimensions, including none

Weight: (out\_features,in\_features)(out\_features,in\_features) or (in\_features)(in\_features)

Bias: (out\_features)(out\_features) or ()()

Output: (∗,out\_features)(∗,out\_features) or (∗)(∗), based on the shape of the weight

Best Combination (Trail 2):

# Best Hyperparameter Choices from the Search (Trial 2)

The following hyperparameter choices were made for the layers that were dynamically tuned:

- \*\*bert.encoder.layer.0.attention.self.query\*\*

- \*\*Type:\*\* `LinearInteger`

- \*\*Data Input Width:\*\* 32

- \*\*Data Input Fractional Width:\*\* 2

- \*\*Weight Width:\*\* 16

- \*\*Weight Fractional Width:\*\* 8

- \*\*Bias Width:\*\* 8

- \*\*Bias Fractional Width:\*\* 8

- \*\*bert.encoder.layer.0.intermediate.dense\*\*

- \*\*Type:\*\* `LinearInteger`

- \*\*Data Input Width:\*\* 16

- \*\*Data Input Fractional Width:\*\* 2

- \*\*Weight Width:\*\* 8

- \*\*Weight Fractional Width:\*\* 8

- \*\*Bias Width:\*\* 16

- \*\*Bias Fractional Width:\*\* 4

- \*\*bert.encoder.layer.1.attention.self.query\*\*

- \*\*Type:\*\* `LinearInteger`

- \*\*Data Input Width:\*\* 32

- \*\*Data Input Fractional Width:\*\* 8

- \*\*Weight Width:\*\* 16

- \*\*Weight Fractional Width:\*\* 2

- \*\*Bias Width:\*\* 16

- \*\*Bias Fractional Width:\*\* 2

- \*\*bert.encoder.layer.1.attention.self.key\*\*

- \*\*Type:\*\* `LinearInteger`

- \*\*Data Input Width:\*\* 16

- \*\*Data Input Fractional Width:\*\* 4

- \*\*Weight Width:\*\* 32

- \*\*Weight Fractional Width:\*\* 2

- \*\*Bias Width:\*\* 32

- \*\*Bias Fractional Width:\*\* 4

- \*\*bert.encoder.layer.1.attention.self.value\*\*

- \*\*Type:\*\* `LinearInteger`

- \*\*Data Input Width:\*\* 16

- \*\*Data Input Fractional Width:\*\* 4

- \*\*Weight Width:\*\* 16

- \*\*Weight Fractional Width:\*\* 4

- \*\*Bias Width:\*\* 32

- \*\*Bias Fractional Width:\*\* 8

- \*\*bert.encoder.layer.1.intermediate.dense\*\*

- \*\*Type:\*\* `LinearInteger`

- \*\*Data Input Width:\*\* 8

- \*\*Data Input Fractional Width:\*\* 2

- \*\*Weight Width:\*\* 16

- \*\*Weight Fractional Width:\*\* 8

- \*\*Bias Width:\*\* 16

- \*\*Bias Fractional Width:\*\* 8

- \*\*bert.encoder.layer.1.output.dense\*\*

- \*\*Type:\*\* `LinearInteger`

- \*\*Data Input Width:\*\* 16

- \*\*Data Input Fractional Width:\*\* 2

- \*\*Weight Width:\*\* 16

- \*\*Weight Fractional Width:\*\* 8

- \*\*Bias Width:\*\* 16

- \*\*Bias Fractional Width:\*\* 4

- \*\*bert.pooler.dense\*\*

- \*\*Type:\*\* `LinearInteger`

- \*\*Data Input Width:\*\* 16

- \*\*Data Input Fractional Width:\*\* 8

- \*\*Weight Width:\*\* 16

- \*\*Weight Fractional Width:\*\* 2

- \*\*Bias Width:\*\* 16

- \*\*Bias Fractional Width:\*\* 2

**Combined Optuna Results:**

FullPrecision

0, 0.87852

1, 0.8764

2, 0.87616

3, 0.87632

4, 0.87668

LinearInteger

0, 0.87264

1, 0.85892

2, 0.86988

3, 0.87984

4, 0.86192

LinearMinifloatDenorm

0, 0.87764

1, 0.8764

2, 0.878

3, 0.8756

4, 0.87752

LinearMinifloatIEEE

0, 0.87636

1, 0.87652

2, 0.87708

3, 0.87684

4, 0.87672

LinearLog

0, 0.5

1, 0.5

2, 0.5

3, 0.5

4, 0.5

LinearBlockFP

0, 0.8768

1, 0.87592

2, 0.87508

3, 0.87736

4, 0.8768

LinearBlockLog

0, 0.5

1, 0.5

2, 0.5

3, 0.5

4, 0.5

LinearBinary

0, 0.50424

1, 0.50424

2, 0.50424

3, 0.50424

4, 0.50424

LinearBinaryScaling

0, 0.59436

1, 0.59436

2, 0.59436

3, 0.59436

4, 0.59436