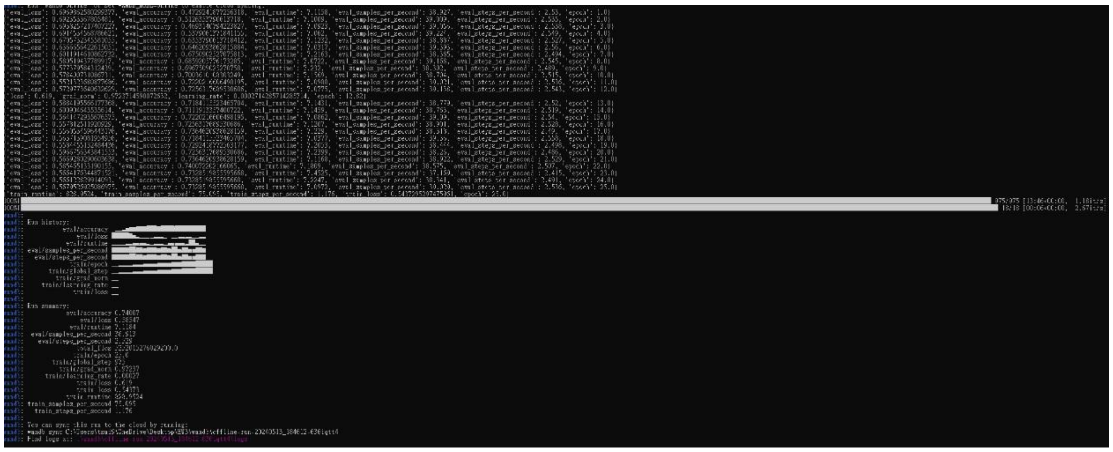
Model Analysis

In this assignment, I selected RoBERTa, a natural language processing model based on BERT. It enhances BERT by removing the Next Sentence Prediction task, as research has shown that it contributes little to performance improvements. Instead, RoBERTa only uses the Masked Language Model task for training. Additionally, RoBERTa is trained for a longer time and with more steps, enabling the model to capture language features more effectively.

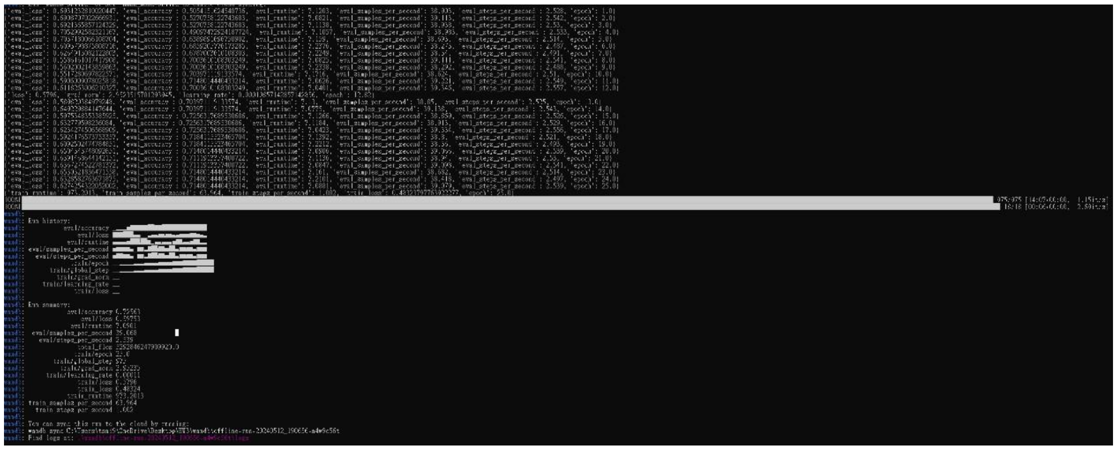
In the code, I used the `transformers.AutoModelForSequenceClassification.from_pretrained()` function, setting `num_labels` to 2 for classification, and then loading the model. Afterward, I used the `trainer` function to train the model with the dataset. This function optimizes the model parameters through backpropagation and gradient descent and computes performance metrics on the validation set at the end of each training epoch, such as accuracy and loss. The trained model is saved, and I used the default AdamW optimizer.

- Rte dataset_bitfit: Acc: 0.74007



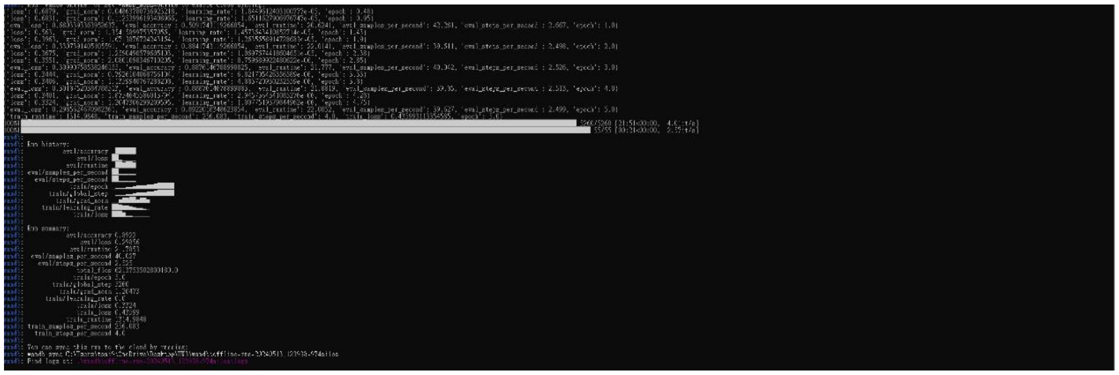
```
[illegible]
{'eval_loss': 0.6959352580299377, 'eval_accuracy': 0.4729241877256318, 'eval_runtime': 7.1158, 'eval_samples_per_second': 38.927, 'eval_steps_per_second': 2.53, 'epoch': 1.0}
{'eval_loss': 0.6925535578053481, 'eval_accuracy': 0.5126353790613718, 'eval_runtime': 7.1009, 'eval_samples_per_second': 39.009, 'eval_steps_per_second': 2.535, 'epoch': 2.0}
{'eval_loss': 0.6953257217407227, 'eval_accuracy': 0.4693140794223827, 'eval_runtime': 7.0923, 'eval_samples_per_second': 39.055, 'eval_steps_per_second': 2.538, 'epoch': 3.0}
{'eval_loss': 0.6917557568786621, 'eval_accuracy': 0.5379061371841155, 'eval_runtime': 7.062, 'eval_samples_per_second': 39.224, 'eval_steps_per_second': 2.549, 'epoch': 4.0}
{'eval_loss': 0.6705732345581055, 'eval_accuracy': 0.6353790613718412, 'eval_runtime': 7.1232, 'eval_samples_per_second': 38.887, 'eval_steps_per_second': 2.527, 'epoch': 5.0}
{'eval_loss': 0.6566566422615051, 'eval_accuracy': 0.6462093862815884, 'eval_runtime': 7.0317, 'eval_samples_per_second': 39.393, 'eval_steps_per_second': 2.56, 'epoch': 6.0}
{'eval_loss': 0.6011914610862732, 'eval_accuracy': 0.6750902527075813, 'eval_runtime': 7.2163, 'eval_samples_per_second': 38.385, 'eval_steps_per_second': 2.494, 'epoch': 7.0}
{'eval_loss': 0.5805104377899170, 'eval_accuracy': 0.6859205776173285, 'eval_runtime': 7.0722, 'eval_samples_per_second': 39.168, 'eval_steps_per_second': 2.545, 'epoch': 8.0}
{'eval_loss': 0.5773795843124390, 'eval_accuracy': 0.6967509025270758, 'eval_runtime': 7.232, 'eval_samples_per_second': 38.302, 'eval_steps_per_second': 2.489, 'epoch': 9.0}
{'eval_loss': 0.5764307308197021, 'eval_accuracy': 0.7003610108303249, 'eval_runtime': 7.1569, 'eval_samples_per_second': 38.704, 'eval_steps_per_second': 2.515, 'epoch': 10.0}
{'eval_loss': 0.5521323680877686, 'eval_accuracy': 0.7220216606498195, 'eval_runtime': 7.0980, 'eval_samples_per_second': 39.021, 'eval_steps_per_second': 2.536, 'epoch': 11.0}
{'eval_loss': 0.5729773640632629, 'eval_accuracy': 0.7256317689530686, 'eval_runtime': 7.0775, 'eval_samples_per_second': 39.138, 'eval_steps_per_second': 2.543, 'epoch': 12.0}
{'loss': 0.619, 'grad_norm': 0.5723714590072632, 'learning_rate': 0.00027142857142857144, 'epoch': 12.82}
{'eval_loss': 0.5884195566177368, 'eval_accuracy': 0.7184115523465704, 'eval_runtime': 7.1431, 'eval_samples_per_second': 38.779, 'eval_steps_per_second': 2.52, 'epoch': 13.0}
{'eval_loss': 0.6009064535535614, 'eval_accuracy': 0.7111913357400722, 'eval_runtime': 7.1459, 'eval_samples_per_second': 38.763, 'eval_steps_per_second': 2.519, 'epoch': 14.0}
{'eval_loss': 0.5641472935676575, 'eval_accuracy': 0.7220216606498195, 'eval_runtime': 7.0862, 'eval_samples_per_second': 39.09, 'eval_steps_per_second': 2.54, 'epoch': 15.0}
{'eval_loss': 0.5578125119209290, 'eval_accuracy': 0.7256317689530686, 'eval_runtime': 7.1207, 'eval_samples_per_second': 38.901, 'eval_steps_per_second': 2.528, 'epoch': 16.0}
{'eval_loss': 0.5595055[illegible], 'eval_accuracy': 0.7364620938628159, 'eval_runtime': 7.229, 'eval_samples_per_second': 38.318, 'eval_steps_per_second': 2.49, 'epoch': 17.0}
{'eval_loss': 0.5657159081954956, 'eval_accuracy': 0.7184115523465704, 'eval_runtime': 7.0377, 'eval_samples_per_second': 39.35, 'eval_steps_per_second': 2.558, 'epoch': 18.0}
{'eval_loss': 0.5584455132484436, 'eval_accuracy': 0.7292418772563177, 'eval_runtime': 7.2053, 'eval_samples_per_second': 38.[illegible], 'eval_steps_per_second': 2.498, 'epoch': 19.0}
{'eval_loss': 0.5966756343841553, 'eval_accuracy': 0.7256317689530686, 'eval_runtime': 7.2399, 'eval_samples_per_second': 38.26, 'eval_steps_per_second': 2.486, 'epoch': 20.0}
{'eval_loss': 0.5669280290603638, 'eval_accuracy': 0.7364620938628159, 'eval_runtime': 7.1168, 'eval_samples_per_second': 38.922, 'eval_steps_per_second': 2.529, 'epoch': 21.0}
{'eval_loss': 0.5854551331901550, 'eval_accuracy': 0.740072202166065, 'eval_runtime': 7.1809, 'eval_samples_per_second': 38.575, 'eval_steps_per_second': 2.507, 'epoch': 22.0}
{'eval_loss': 0.5664178344487152, 'eval_accuracy': 0.7328519855595668, 'eval_runtime': 7.4525, 'eval_samples_per_second': 37.169, 'eval_steps_per_second': 2.415, 'epoch': 23.0}
{'eval_loss': 0.5581228299140930, 'eval_accuracy': 0.7328519855595668, 'eval_runtime': 7.2247, 'eval_samples_per_second': 38.341, 'eval_steps_per_second': 2.491, 'epoch': 24.0}
{'eval_loss': 0.5879525925086975, 'eval_accuracy': 0.7328519855595668, 'eval_runtime': 7.0972, 'eval_samples_per_second': 39.029, 'eval_steps_per_second': 2.536, 'epoch': 25.0}
{'train_runtime': 828.9524, 'train_samples_per_second': 75.095, 'train_steps_per_second': 1.176, 'train_loss': 0.5437205297475051, 'epoch': 25.0}
100%|████████████████████████████████████████| 975/975 [13:46<00:00,  1.18it/s]
100%|████████████████████████████████████████| 18/18 [00:06<00:00,  2.67it/s]
wandb:
wandb: Run history:
wandb:            eval/accuracy
wandb:                eval/loss
wandb:             eval/runtime
wandb:  eval/samples_per_second
wandb:    eval/steps_per_second
wandb:              train/epoch
wandb:        train/global_step
wandb:          train/grad_norm
wandb:      train/learning_rate
wandb:               train/loss
wandb:
wandb: Run summary:
wandb:            eval/accuracy 0.74007
wandb:                eval/loss 0.58547
wandb:             eval/runtime 7.1184
wandb:  eval/samples_per_second 38.513
wandb:    eval/steps_per_second 2.528
wandb:               total_flos 2252015276029200.0
wandb:              train/epoch 25.0
wandb:        train/global_step 975
wandb:          train/grad_norm 0.57237
wandb:      train/learning_rate 0.00027
wandb:               train/loss 0.619
wandb:               train_loss 0.54373
wandb:            train_runtime 828.9524
wandb: train_samples_per_second 75.095
wandb:   train_steps_per_second 1.176
wandb:
wandb: You can sync this run to the cloud by running:
wandb: wandb sync C:\Users\tami5\OneDrive\Desktop\HW3\wandb\offline-run-20240513_184612-636iqtt4
wandb: Find logs at: .\wandb\offline-run-20240513_184612-636iqtt4\logs
```

- Rte dataset_lora: Acc: 0.72563



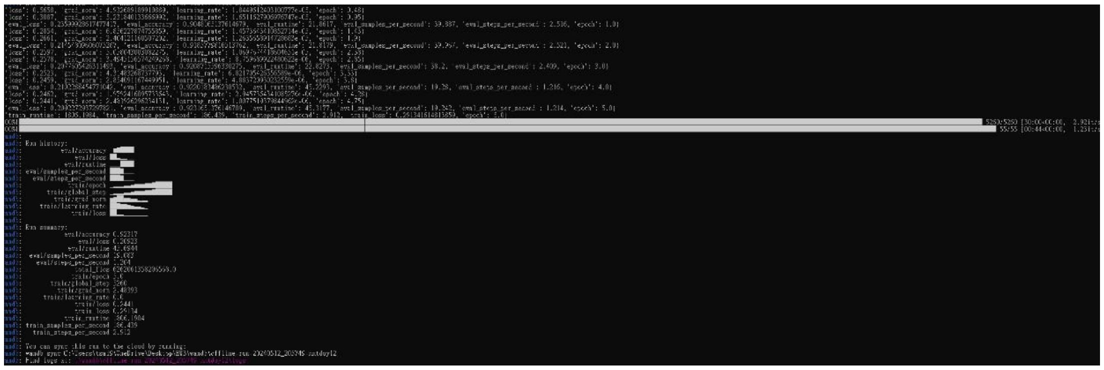
```
{'eval_loss': 0.6931252810020447, 'eval_accuracy': 0.5054151624548736, 'eval_runtime': 7.1203, 'eval_samples_per_second': 38.903, 'eval_steps_per_second': 2.528, 'epoch': 1.0}
{'eval_loss': 0.6908707022666931, 'eval_accuracy': 0.5270758122743683, 'eval_runtime': 7.0821, 'eval_samples_per_second': 39.113, 'eval_steps_per_second': 2.542, 'epoch': 2.0}
{'eval_loss': 0.6921355857124325, 'eval_accuracy': 0.5270758122743683, 'eval_runtime': 7.1138, 'eval_samples_per_second': 38.938, 'eval_steps_per_second': 2.53, 'epoch': 3.0}
{'eval_loss': 0.7052992582321167, 'eval_accuracy': 0.49097472924187724, 'eval_runtime': 7.1057, 'eval_samples_per_second': 38.983, 'eval_steps_per_second': 2.533, 'epoch': 4.0}
{'eval_loss': 0.7057130066108704, 'eval_accuracy': 0.628158844765343, 'eval_runtime': 7.159, 'eval_samples_per_second': 38.693, 'eval_steps_per_second': 2.514, 'epoch': 5.0}
{'eval_loss': 0.6095793875808716, 'eval_accuracy': 0.6859205776173285, 'eval_runtime': 7.2276, 'eval_samples_per_second': 38.325, 'eval_steps_per_second': 2.487, 'epoch': 6.0}
{'eval_loss': 0.6297915082122802, 'eval_accuracy': 0.6787003610108303, 'eval_runtime': 7.2249, 'eval_samples_per_second': 38.34, 'eval_steps_per_second': 2.491, 'epoch': 7.0}
{'eval_loss': 0.5586161017417908, 'eval_accuracy': 0.7002610108303249, 'eval_runtime': 7.0825, 'eval_samples_per_second': 39.111, 'eval_steps_per_second': 2.541, 'epoch': 8.0}
{'eval_loss': 0.5692032143859863, 'eval_accuracy': 0.7002610108303249, 'eval_runtime': 7.2338, 'eval_samples_per_second': 38.292, 'eval_steps_per_second': 2.488, 'epoch': 9.0}
{'eval_loss': 0.5517289697822257, 'eval_accuracy': 0.703971119133574, 'eval_runtime': 7.1716, 'eval_samples_per_second': 38.624, 'eval_steps_per_second': 2.51, 'epoch': 10.0}
{'eval_loss': 0.5906309078025818, 'eval_accuracy': 0.7148014440433214, 'eval_runtime': 7.0626, 'eval_samples_per_second': 39.221, 'eval_steps_per_second': 2.549, 'epoch': 11.0}
{'eval_loss': 0.6116353006210327, 'eval_accuracy': 0.7003610108303249, 'eval_runtime': 7.0401, 'eval_samples_per_second': 39.346, 'eval_steps_per_second': 2.557, 'epoch': 12.0}
{'loss': 0.5796, 'grad_norm': 2.552351570129304, 'learning_rate': 0.00013857142857142856, 'epoch': 12.82}
{'eval_loss': 0.5898233849793248, 'eval_accuracy': 0.703971119133574, 'eval_runtime': 7.1, 'eval_samples_per_second': 39.05, 'eval_steps_per_second': 2.525, 'epoch': 13.0}
{'eval_loss': 0.6492298844147644, 'eval_accuracy': 0.703971119133574, 'eval_runtime': 7.0775, 'eval_samples_per_second': 39.138, 'eval_steps_per_second': 2.543, 'epoch': 14.0}
{'eval_loss': 0.5975348353385925, 'eval_accuracy': 0.7256317689530686, 'eval_runtime': 7.1266, 'eval_samples_per_second': 38.869, 'eval_steps_per_second': 2.526, 'epoch': 15.0}
{'eval_loss': 0.6327795982360840, 'eval_accuracy': 0.7256317689530686, 'eval_runtime': 7.1184, 'eval_samples_per_second': 38.913, 'eval_steps_per_second': 2.529, 'epoch': 16.0}
{'eval_loss': 0.6234274506568909, 'eval_accuracy': 0.7256317689530686, 'eval_runtime': 7.0423, 'eval_samples_per_second': 39.334, 'eval_steps_per_second': 2.556, 'epoch': 17.0}
{'eval_loss': 0.5924176557373337, 'eval_accuracy': 0.7184115523465704, 'eval_runtime': 7.1392, 'eval_samples_per_second': 38.8, 'eval_steps_per_second': 2.521, 'epoch': 18.0}
{'eval_loss': 0.6092532474784851, 'eval_accuracy': 0.7184115523465704, 'eval_runtime': 7.2212, 'eval_samples_per_second': 38.36, 'eval_steps_per_second': 2.493, 'epoch': 19.0}
{'eval_loss': 0.6575757480092651, 'eval_accuracy': 0.7148014440433214, 'eval_runtime': 7.0906, 'eval_samples_per_second': 39.065, 'eval_steps_per_second': 2.539, 'epoch': 20.0}
{'eval_loss': 0.6591763644142151, 'eval_accuracy': 0.7111913357400722, 'eval_runtime': 7.1136, 'eval_samples_per_second': 38.94, 'eval_steps_per_second': 2.53, 'epoch': 21.0}
{'eval_loss': 0.6597245522781372, 'eval_accuracy': 0.7111913357400722, 'eval_runtime': 7.0847, 'eval_samples_per_second': 39.098, 'eval_steps_per_second': 2.541, 'epoch': 22.0}
{'eval_loss': 0.6530521836471558, 'eval_accuracy': 0.7148014440433214, 'eval_runtime': 7.161, 'eval_samples_per_second': 38.682, 'eval_steps_per_second': 2.514, 'epoch': 23.0}
{'eval_loss': 0.6328582763671875, 'eval_accuracy': 0.7148014440433214, 'eval_runtime': 7.2101, 'eval_samples_per_second': 38.418, 'eval_steps_per_second': 2.497, 'epoch': 24.0}
{'eval_loss': 0.6274254322052002, 'eval_accuracy': 0.7148014440433214, 'eval_runtime': 7.0801, 'eval_samples_per_second': 39.123, 'eval_steps_per_second': 2.539, 'epoch': 25.0}
{'train_runtime': 973.2013, 'train_samples_per_second': 63.964, 'train_steps_per_second': 1.002, 'train_loss': 0.48323797763922277, 'epoch': 25.0}
100%|                                                                   | 975/975 [14:07<00:00,  1.15it/s]
100%|                                                                   | 18/18 [00:06<00:00,  2.80it/s]
wandb:
wandb: Run history:
wandb:                eval/accuracy
wandb:                    eval/loss
wandb:                 eval/runtime
wandb:      eval/samples_per_second
wandb:        eval/steps_per_second
wandb:                  train/epoch
wandb:            train/global_step
wandb:              train/grad_norm
wandb:          train/learning_rate
wandb:                   train/loss
wandb:
wandb: Run summary:
wandb:                eval/accuracy 0.72563
wandb:                    eval/loss 0.59753
wandb:                 eval/runtime 7.0901
wandb:      eval/samples_per_second 39.068
wandb:        eval/steps_per_second 2.539
wandb:                   total_flos 2252846247909920.0
wandb:                  train/epoch 25.0
wandb:            train/global_step 975
wandb:              train/grad_norm 2.55235
wandb:          train/learning_rate 0.00011
wandb:                   train/loss 0.5796
wandb:                   train_loss 0.48324
wandb:                train_runtime 973.2013
wandb:     train_samples_per_second 63.964
wandb:       train_steps_per_second 1.002
wandb:
wandb: You can sync this run to the cloud by running:
wandb: wandb sync C:\Users\tsai9\OneDrive\Desktop\HW3\wandb\offline-run-20240512_190656-a4w9c56t
wandb: Find logs at: .\wandb\offline-run-20240512_190656-a4w9c56t\logs
```

- Sst2 dataset_bitfit: Acc: 0.8922



```
{'loss': 0.6879, 'grad_norm': 0.6486378073692521, 'learning_rate': 1.8449612403100777e-05, 'epoch': 0.48}
{'loss': 0.6831, 'grad_norm': 0.253996193408966, 'learning_rate': 1.6511627906976745e-05, 'epoch': 0.95}
{'eval_loss': 0.6833593363952635, 'eval_accuracy': 0.5091743119266054, 'eval_runtime': 20.6241, 'eval_samples_per_second': 42.281, 'eval_steps_per_second': 2.667, 'epoch': 1.0}
{'loss': 0.563, 'grad_norm': 1.354389975357055, 'learning_rate': 1.4573643410852714e-05, 'epoch': 1.43}
{'loss': 0.3963, 'grad_norm': 1.657076724243154, 'learning_rate': 1.2635658914728683e-05, 'epoch': 1.9}
{'eval_loss': 0.3307391405105591, 'eval_accuracy': 0.8841743119266054, 'eval_runtime': 22.0141, 'eval_samples_per_second': 39.611, 'eval_steps_per_second': 2.498, 'epoch': 2.0}
{'loss': 0.3675, 'grad_norm': 1.2596490579605135, 'learning_rate': 1.0697674418604651e-05, 'epoch': 2.38}
{'loss': 0.3551, 'grad_norm': 2.080109834671020, 'learning_rate': 8.759689922480622e-06, 'epoch': 2.85}
{'eval_loss': 0.3099075853824615, 'eval_accuracy': 0.8876146788990825, 'eval_runtime': 21.777, 'eval_samples_per_second': 40.042, 'eval_steps_per_second': 2.526, 'epoch': 3.0}
{'loss': 0.3444, 'grad_norm': 0.7526104068755136, 'learning_rate': 6.821705426356589e-06, 'epoch': 3.33}
{'loss': 0.3406, 'grad_norm': 1.1255840767283208, 'learning_rate': 4.883720930232558e-06, 'epoch': 3.8}
{'eval_loss': 0.3013752038478852, 'eval_accuracy': 0.8887614678899083, 'eval_runtime': 21.8819, 'eval_samples_per_second': 39.85, 'eval_steps_per_second': 2.513, 'epoch': 4.0}
{'loss': 0.3401, 'grad_norm': 1.8754045586013579, 'learning_rate': 2.945736434108527e-06, 'epoch': 4.28}
{'loss': 0.3324, 'grad_norm': 1.2047306299209595, 'learning_rate': 1.0077519379844962e-06, 'epoch': 4.75}
{'eval_loss': 0.2985624670982961, 'eval_accuracy': 0.8922018348623854, 'eval_runtime': 22.0052, 'eval_samples_per_second': 39.627, 'eval_steps_per_second': 2.499, 'epoch': 5.0}
{'train_runtime': 1314.9848, 'train_samples_per_second': 256.083, 'train_steps_per_second': 4.0, 'train_loss': 0.4359931135594585, 'epoch': 5.0}
100%|                                                                   | 5260/5260 [21:51<00:00,  4.01it/s]
100%|                                                                   | 55/55 [00:21<00:00,  2.55it/s]
wandb:
wandb: Run history:
wandb:                eval/accuracy
wandb:                    eval/loss
wandb:                 eval/runtime
wandb:      eval/samples_per_second
wandb:        eval/steps_per_second
wandb:                  train/epoch
wandb:            train/global_step
wandb:              train/grad_norm
wandb:          train/learning_rate
wandb:                   train/loss
wandb:
wandb: Run summary:
wandb:                eval/accuracy 0.8922
wandb:                    eval/loss 0.29856
wandb:                 eval/runtime 22.7053
wandb:      eval/samples_per_second 40.027
wandb:        eval/steps_per_second 2.525
wandb:                   total_flos 6213753502800180.0
wandb:                  train/epoch 5.0
wandb:            train/global_step 5260
wandb:              train/grad_norm 1.20473
wandb:          train/learning_rate 0.0
wandb:                   train/loss 0.3324
wandb:                   train_loss 0.43599
wandb:                train_runtime 1314.9848
wandb:     train_samples_per_second 256.083
wandb:       train_steps_per_second 4.0
wandb:
wandb: You can sync this run to the cloud by running:
wandb: wandb sync C:\Users\tsai9\OneDrive\Desktop\HW3\wandb\offline-run-20240513_123938-974ailoo
wandb: Find logs at: .\wandb\offline-run-20240513_123938-974ailoo\logs
```

- Sst2 dataset_lora: Acc: 0.92317



```
{'loss': 0.5658, 'grad_norm': 4.932689189910889, 'learning_rate': 1.8449612403100777e-05, 'epoch': 0.48}
{'loss': 0.3087, 'grad_norm': 5.221840133666992, 'learning_rate': 1.6511627906976747e-05, 'epoch': 0.95}
{'eval_loss': 0.2359992861747743, 'eval_accuracy': 0.9048165137614679, 'eval_runtime': 21.8617, 'eval_samples_per_second': 39.887, 'eval_steps_per_second': 2.516, 'epoch': 1.0}
{'loss': 0.2854, 'grad_norm': 6.836227874755859, 'learning_rate': 1.4573643410852714e-05, 'epoch': 1.43}
{'loss': 0.2661, 'grad_norm': 2.404121160507202, 'learning_rate': 1.2635658914728682e-05, 'epoch': 1.9}
{'eval_loss': 0.2175730960607529, 'eval_accuracy': 0.9185779816513762, 'eval_runtime': 21.8179, 'eval_samples_per_second': 39.967, 'eval_steps_per_second': 2.521, 'epoch': 2.0}
{'loss': 0.2597, 'grad_norm': 5.028043003082275, 'learning_rate': 1.0697674418604651e-05, 'epoch': 2.38}
{'loss': 0.2578, 'grad_norm': 2.484513657424926, 'learning_rate': 8.759689922480622e-06, 'epoch': 2.85}
{'eval_loss': 0.2077605426311493, 'eval_accuracy': 0.9208715596330275, 'eval_runtime': 22.8273, 'eval_samples_per_second': 38.2, 'eval_steps_per_second': 2.409, 'epoch': 3.0}
{'loss': 0.2523, 'grad_norm': 4.3.483268737793, 'learning_rate': 6.821705426356589e-06, 'epoch': 3.33}
{'loss': 0.2459, 'grad_norm': 2.854091167449951, 'learning_rate': 4.883720930232559e-06, 'epoch': 3.8}
{'eval_loss': 0.2102288454771042, 'eval_accuracy': 0.9220183486238532, 'eval_runtime': 45.2293, 'eval_samples_per_second': 19.28, 'eval_steps_per_second': 1.216, 'epoch': 4.0}
{'loss': 0.2462, 'grad_norm': 1.9592416095733643, 'learning_rate': 2.945736434108527e-06, 'epoch': 4.28}
{'loss': 0.2441, 'grad_norm': 2.4839262962341, 'learning_rate': 1.0077519379844962e-06, 'epoch': 4.75}
{'eval_loss': 0.2092272937297821, 'eval_accuracy': 0.9231651376146789, 'eval_runtime': 45.3177, 'eval_samples_per_second': 19.242, 'eval_steps_per_second': 1.214, 'epoch': 5.0}
{'train_runtime': 1806.1984, 'train_samples_per_second': 186.439, 'train_steps_per_second': 2.912, 'train_loss': 0.2913416148135859, 'epoch': 5.0}
100%|                                                                   | 5260/5260 [30:00<00:00,  2.92it/s
100%|                                                                   | 55/55 [00:44<00:00,  1.23it/s
wandb:
wandb: Run history:
wandb:                eval/accuracy
wandb:                    eval/loss
wandb:                 eval/runtime
wandb:      eval/samples_per_second
wandb:        eval/steps_per_second
wandb:                  train/epoch
wandb:            train/global_step
wandb:              train/grad_norm
wandb:          train/learning_rate
wandb:                   train/loss
wandb:
wandb: Run summary:
wandb:                eval/accuracy 0.92317
wandb:                    eval/loss 0.20923
wandb:                 eval/runtime 45.6944
wandb:      eval/samples_per_second 19.083
wandb:        eval/steps_per_second 1.204
wandb:                   total_flos 6262061358206568.0
wandb:                  train/epoch 5.0
wandb:            train/global_step 5260
wandb:              train/grad_norm 2.48393
wandb:          train/learning_rate 0.0
wandb:                   train/loss 0.2441
wandb:                   train_loss 0.29134
wandb:                train_runtime 1806.1984
wandb:     train_samples_per_second 186.439
wandb:       train_steps_per_second 2.912
wandb:
wandb: You can sync this run to the cloud by running:
wandb: wandb sync C:\Users\tsai9\OneDrive\Desktop\HW3\wandb\offline-run-20240512_203745-zntdoy12
wandb: Find logs at: .\wandb\offline-run-20240512_203745-zntdoy12\logs
```

PEFT Discussion

For transformer models, Bitfit freezes most model parameters and only updates the bias parameters and the classification layer specific to the task. Examples of bias parameters include:

- Bias in the MLP layer

- Bias involved in calculating query, key, and value in the attention module and merging multiple attention results
- Bias in the Layer Normalization layer

These bias parameters account for around 0.08% of the total parameters in a model like BERT.

By updating only a portion of the bias parameters and task-specific classification layer parameters, the model can better recognize specific downstream tasks, thereby improving accuracy without training a large number of parameters.

The best hyperparameters I found are:

Rte dataset:
```
args = transformers.TrainingArguments(
  output_dir='/save',
  num_train_epochs=25,
  learning_rate=5e-4,
  per_device_train_batch_size=16,
  per_device_eval_batch_size=16,
  gradient_accumulation_steps=4,
  warmup_steps=100,
  weight_decay=0.01,
  evaluation_strategy='epoch',
  save_strategy='epoch',
  save_steps=100,
  eval_steps=100,
  load_best_model_at_end=True,
  metric_for_best_model='accuracy'
)

lora_config = LoraConfig(
  task_type=TaskType.SEQ_CLS,
  r=2,
  lora_alpha=8,
  lora_dropout=0.01
)
```
Since the Rte dataset is small, I used more epochs to increase its training.

Sst2 dataset:
```
```

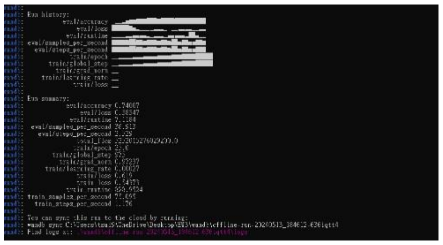
```
args = transformers.TrainingArguments(
  output_dir='/save',
  num_train_epochs=5,
  learning_rate=2e-5,
  per_device_train_batch_size=16,
  per_device_eval_batch_size=16,
  gradient_accumulation_steps=4,
  warmup_steps=100,
  weight_decay=0.01,
  evaluation_strategy='epoch',
  save_strategy='epoch',
  save_steps=100,
  eval_steps=100,
  load_best_model_at_end=True,
  metric_for_best_model='accuracy'
)

lora_config = LoraConfig(
  task_type=TaskType.SEQ_CLS,
  r=2,
  lora_alpha=8,
  lora_dropout=0.01
)
```

In full-finetuning, we update all the model parameters, which usually requires a smaller learning rate to avoid disrupting previously learned knowledge. However, in PEFT, we only update part of the model's parameters, often those in the last few layers or certain other layers. These layers may have different feature representations and learning speeds, so the learning rate may need to be adjusted to better suit their updates. If certain layers require larger parameter changes, a higher learning rate might be needed, while for other layers, a smaller learning rate could prevent overfitting. Therefore, the learning rate needs to be adjusted according to the specific layers chosen for partial fine-tuning and the requirements of the task to achieve optimal performance.

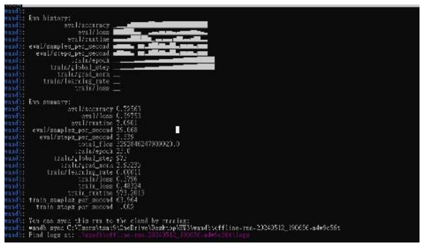
PEFT Comparison

Rte dataset_bitfit:



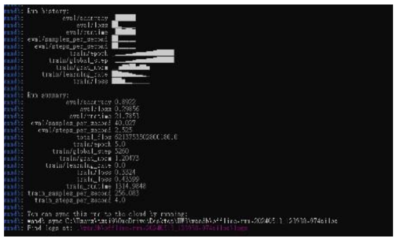
```
wandb:
wandb: Run history:
wandb:            eval/accuracy
wandb:                eval/loss
wandb:             eval/runtime
wandb: eval/samples_per_second
wandb:   eval/steps_per_second
wandb:              train/epoch
wandb:        train/global_step
wandb:          train/grad_norm
wandb:      train/learning_rate
wandb:               train/loss
wandb:
wandb: Run summary:
wandb:            eval/accuracy 0.74007
wandb:                eval/loss 0.58547
wandb:             eval/runtime 7.1184
wandb:  eval/samples_per_second 38.913
wandb:    eval/steps_per_second 2.529
wandb:               total_flos 232015276029200.0
wandb:              train/epoch 22.0
wandb:        train/global_step 572
wandb:          train/grad_norm 0.97237
wandb:      train/learning_rate 0.00027
wandb:               train/loss 0.619
wandb:               train_loss 0.54371
wandb:            train_runtime 824.9524
wandb: train_samples_per_second 75.495
wandb:   train_steps_per_second 1.176
wandb:
wandb: You can sync this run to the cloud by running:
wandb: wandb sync C:\Users\tsai9\OneDrive\Desktop\HW3\wandb\offline-run-20240513_184612-6361qtt4
wandb: Find logs at: .\wandb\offline-run-20240513_184612-6361qtt4\logs
```

Rte dataset_lora:



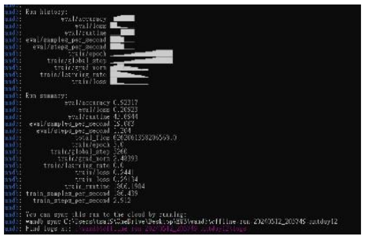
```
wandb:
wandb: Run history:
wandb:            eval/accuracy
wandb:                eval/loss
wandb:             eval/runtime
wandb: eval/samples_per_second
wandb:   eval/steps_per_second
wandb:              train/epoch
wandb:        train/global_step
wandb:          train/grad_norm
wandb:      train/learning_rate
wandb:               train/loss
wandb:
wandb: Run summary:
wandb:            eval/accuracy 0.72563
wandb:                eval/loss 0.59753
wandb:             eval/runtime 7.0901
wandb:  eval/samples_per_second 39.068
wandb:    eval/steps_per_second 2.539
wandb:               total_flos 225284624790992.0
wandb:              train/epoch 22.0
wandb:        train/global_step 572
wandb:          train/grad_norm 2.53235
wandb:      train/learning_rate 0.00011
wandb:               train/loss 0.3796
wandb:               train_loss 0.48324
wandb:            train_runtime 933.2013
wandb: train_samples_per_second 63.964
wandb:   train_steps_per_second 1.002
wandb:
wandb: You can sync this run to the cloud by running:
wandb: wandb sync C:\Users\tsai9\OneDrive\Desktop\HW3\wandb\offline-run-20240512_190656-a4w9c56t
wandb: Find logs at: .\wandb\offline-run-20240512_190656-a4w9c56t\logs
```

Sst2 dataset_bitfit:



```
wandb: Run history:
wandb:            eval/accuracy
wandb:                eval/loss
wandb:             eval/runtime
wandb:  eval/samples_per_second
wandb:    eval/steps_per_second
wandb:              train/epoch
wandb:        train/global_step
wandb:          train/grad_norm
wandb:      train/learning_rate
wandb:               train/loss
wandb:
wandb: Run summary:
wandb:            eval/accuracy 0.8922
wandb:                eval/loss 0.29856
wandb:             eval/runtime 21.7851
wandb:  eval/samples_per_second 40.027
wandb:    eval/steps_per_second 2.525
wandb:               total_flos 6213753502800180.0
wandb:              train/epoch 5.0
wandb:        train/global_step 5260
wandb:          train/grad_norm 1.20473
wandb:      train/learning_rate 0.0
wandb:               train/loss 0.3324
wandb:               train_loss 0.43399
wandb:            train_runtime 1314.9848
wandb: train_samples_per_second 256.083
wandb:   train_steps_per_second 4.0
wandb:
wandb: You can sync this run to the cloud by running:
wandb: wandb sync C:\Users\tsai9\OneDrive\Desktop\HW3\wandb\offline-run-20240513_123930-974ailoc
wandb: Find logs at: .\wandb\offline-run-20240513_123930-974ailoc\logs
```

sst2 dataset_lora:



```
wandb: Run history:
wandb:            eval/accuracy
wandb:                eval/loss
wandb:             eval/runtime
wandb:  eval/samples_per_second
wandb:    eval/steps_per_second
wandb:              train/epoch
wandb:        train/global_step
wandb:          train/grad_norm
wandb:      train/learning_rate
wandb:               train/loss
wandb:
wandb: Run summary:
wandb:            eval/accuracy 0.92317
wandb:                eval/loss 0.20923
wandb:             eval/runtime 41.6944
wandb:  eval/samples_per_second 25.083
wandb:    eval/steps_per_second 1.204
wandb:               total_flos 6262061358206568.0
wandb:              train/epoch 5.0
wandb:        train/global_step 5260
wandb:          train/grad_norm 2.48393
wandb:      train/learning_rate 0.0
wandb:               train/loss 0.2441
wandb:               train_loss 0.29134
wandb:            train_runtime 1806.1904
wandb: train_samples_per_second 186.439
wandb:   train_steps_per_second 2.912
wandb:
wandb: You can sync this run to the cloud by running:
wandb: wandb sync C:\Users\tsai9\OneDrive\Desktop\HW3\wandb\offline-run-20240512_203749-nntduy12
wandb: Find logs at: .\wandb\offline-run-20240512_203749-nntduy12\logs
```

We can see that regardless of the dataset, the accuracy measured is similar. I believe this could be because the SST-2 and Rte datasets are relatively simple, and the capabilities of these different models may be sufficient for the task, so there is no noticeable performance difference.
The size of the Lora rank affects model training in the following ways:

A lower rank may result in underfitting, which, while improving generalization, could cause the model to miss important features in the data, reducing its performance. The parameter count will decrease, whereas a higher rank could lead to overfitting and increased computational complexity due to more parameters and larger matrices. This may increase training time, especially for large datasets and complex models, where the model may overfit the training data, leading to poor performance on unseen data, and the parameter count increases.