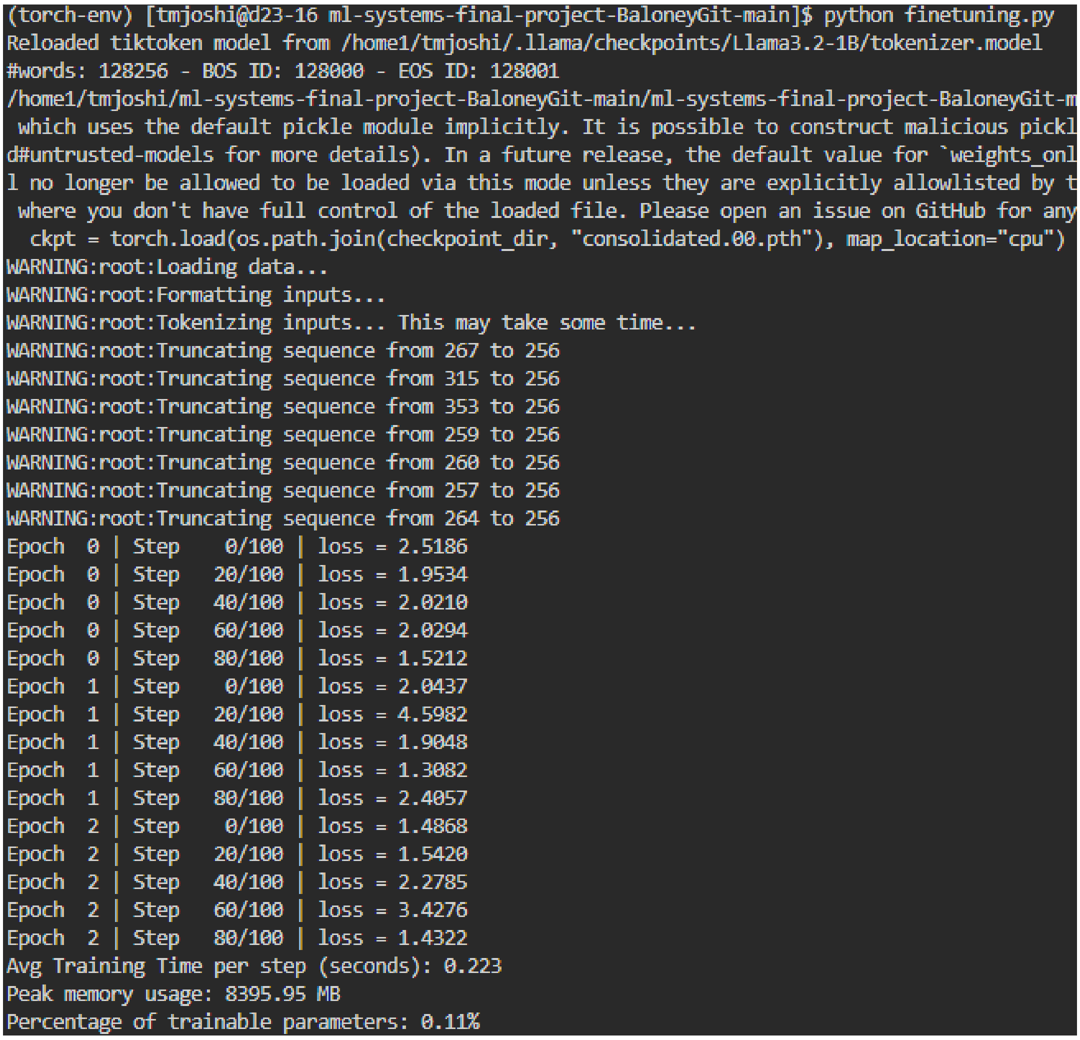
Phase 3

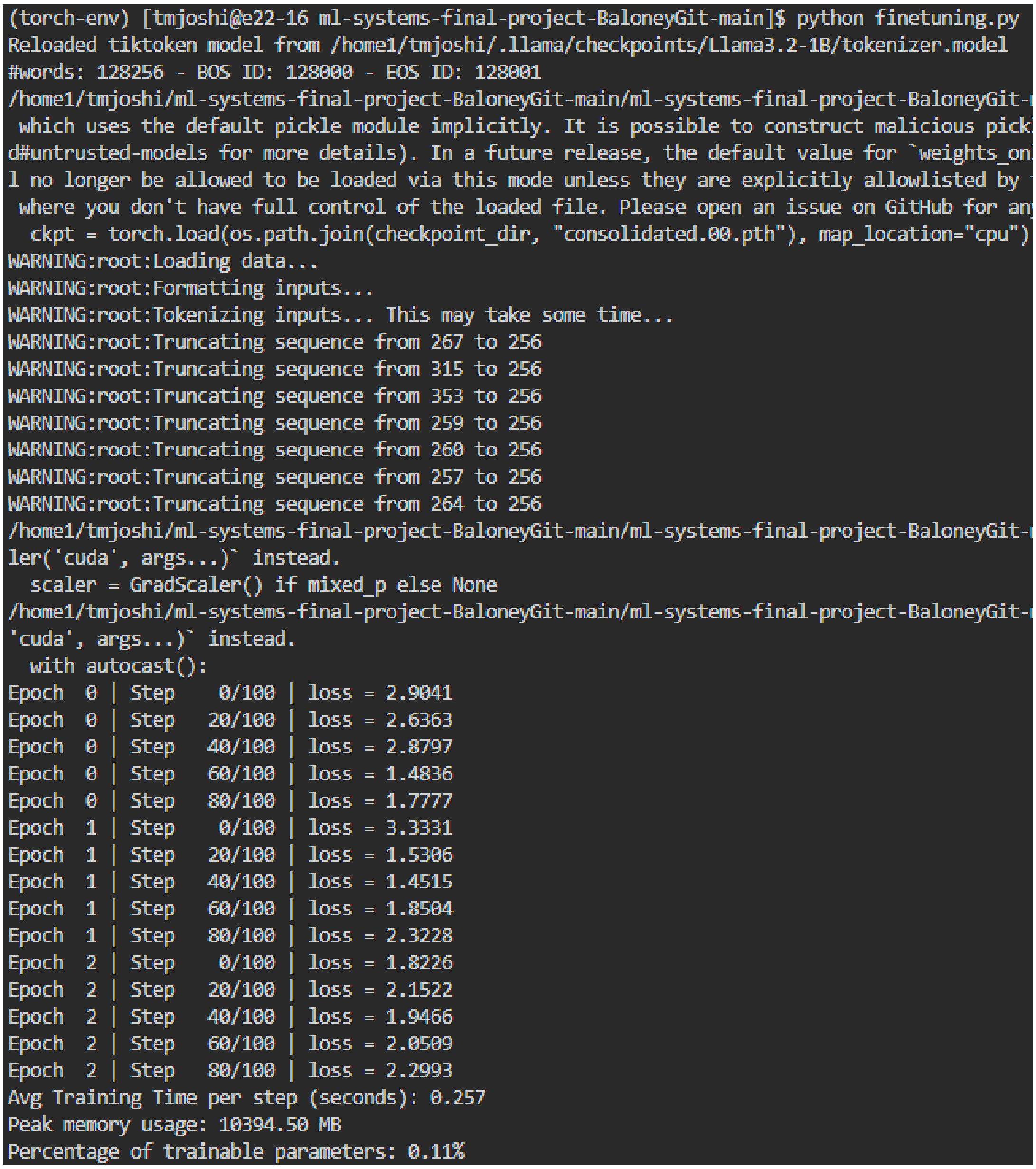
Training Logs:

1. GC = OFF, MP = OFF, LoRA = ON



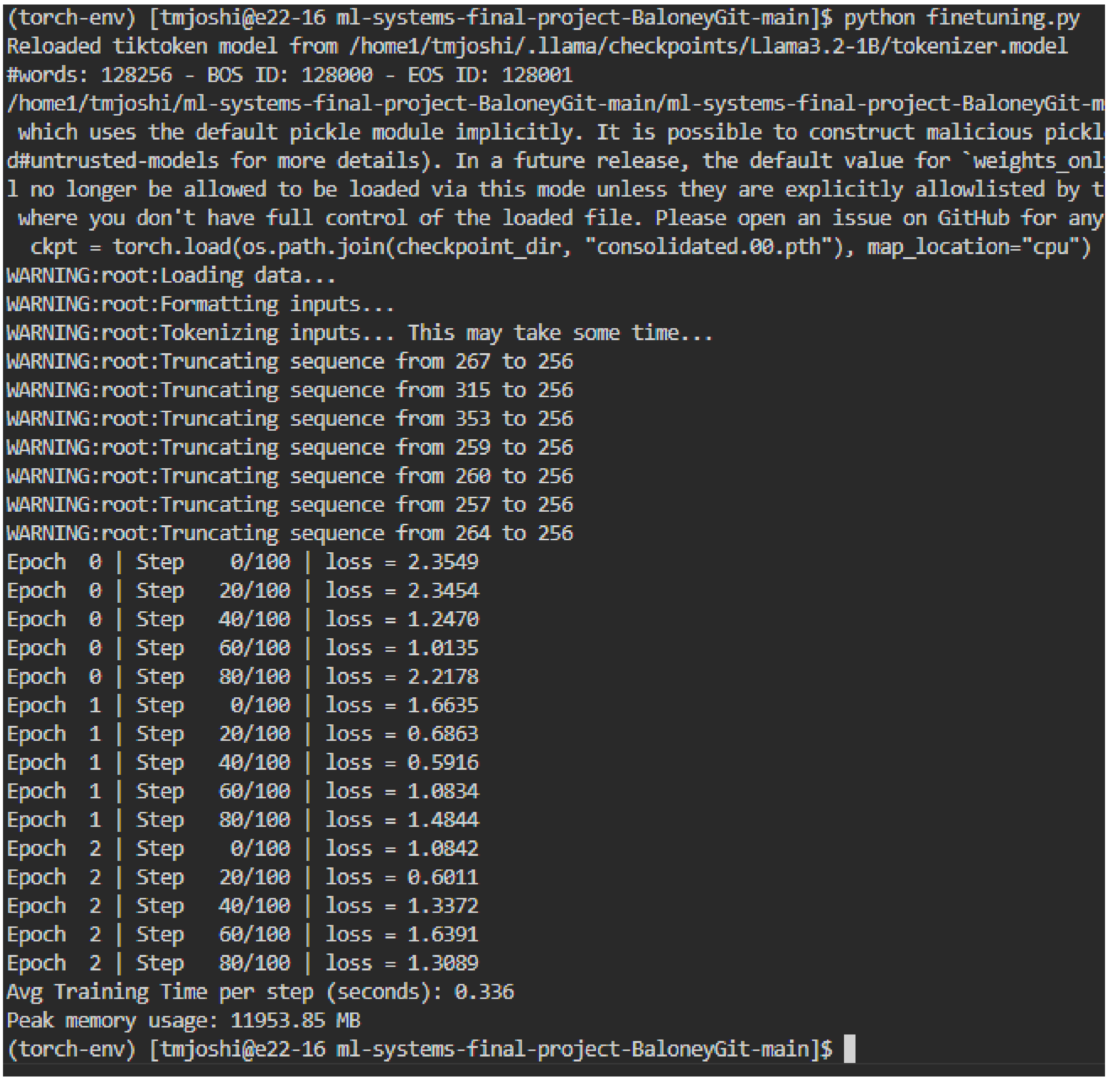
```
(torch-env) [tmjoshi@d23-16 ml-systems-final-project-BaloneyGit-main]$ python finetuning.py
Reloaded tiktoken model from /home1/tmjoshi/.llama/checkpoints/Llama3.2-1B/tokenizer.model
#words: 128256 - BOS ID: 128000 - EOS ID: 128001
/home1/tmjoshi/ml-systems-final-project-BaloneyGit-main/ml-systems-final-project-BaloneyGit-m
 which uses the default pickle module implicitly. It is possible to construct malicious pickl
d#untrusted-models for more details). In a future release, the default value for `weights_onl
l no longer be allowed to be loaded via this mode unless they are explicitly allowlisted by t
 where you don't have full control of the loaded file. Please open an issue on GitHub for any
  ckpt = torch.load(os.path.join(checkpoint_dir, "consolidated.00.pth"), map_location="cpu")
WARNING:root:Loading data...
WARNING:root:Formatting inputs...
WARNING:root:Tokenizing inputs... This may take some time...
WARNING:root:Truncating sequence from 267 to 256
WARNING:root:Truncating sequence from 315 to 256
WARNING:root:Truncating sequence from 353 to 256
WARNING:root:Truncating sequence from 259 to 256
WARNING:root:Truncating sequence from 260 to 256
WARNING:root:Truncating sequence from 257 to 256
WARNING:root:Truncating sequence from 264 to 256
Epoch  0 | Step   0/100 | loss = 2.5186
Epoch  0 | Step  20/100 | loss = 1.9534
Epoch  0 | Step  40/100 | loss = 2.0210
Epoch  0 | Step  60/100 | loss = 2.0294
Epoch  0 | Step  80/100 | loss = 1.5212
Epoch  1 | Step   0/100 | loss = 2.0437
Epoch  1 | Step  20/100 | loss = 4.5982
Epoch  1 | Step  40/100 | loss = 1.9048
Epoch  1 | Step  60/100 | loss = 1.3082
Epoch  1 | Step  80/100 | loss = 2.4057
Epoch  2 | Step   0/100 | loss = 1.4868
Epoch  2 | Step  20/100 | loss = 1.5420
Epoch  2 | Step  40/100 | loss = 2.2785
Epoch  2 | Step  60/100 | loss = 3.4276
Epoch  2 | Step  80/100 | loss = 1.4322
Avg Training Time per step (seconds): 0.223
Peak memory usage: 8395.95 MB
Percentage of trainable parameters: 0.11%
```

2. GC = OFF, MP = ON, LoRA = ON



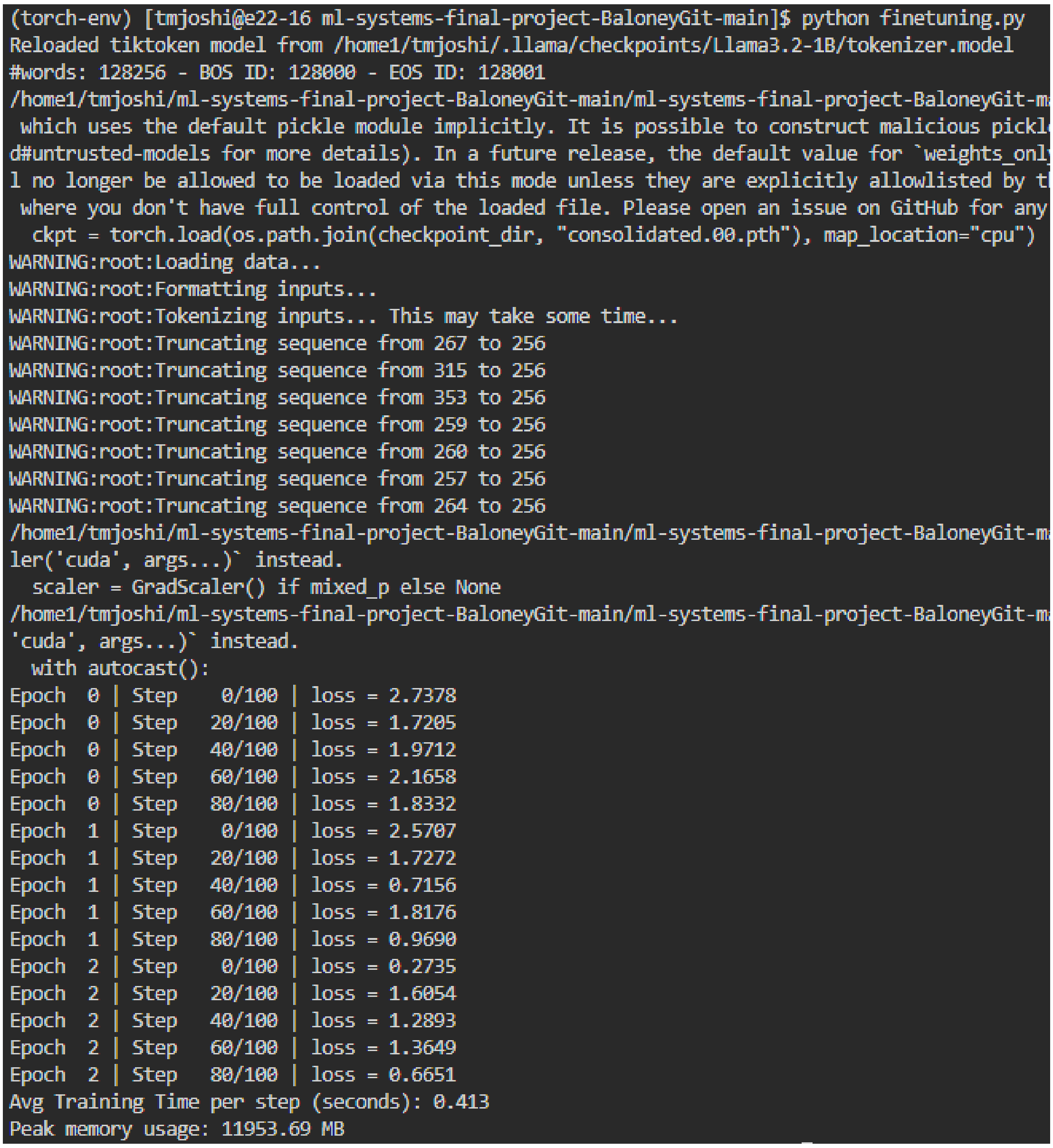
```
(torch-env) [tmjoshi@e22-16 ml-systems-final-project-BaloneyGit-main]$ python finetuning.py
Reloaded tiktoken model from /home1/tmjoshi/.llama/checkpoints/Llama3.2-1B/tokenizer.model
#words: 128256 - BOS ID: 128000 - EOS ID: 128001
/home1/tmjoshi/ml-systems-final-project-BaloneyGit-main/ml-systems-final-project-BaloneyGit-
 which uses the default pickle module implicitly. It is possible to construct malicious pick
d#untrusted-models for more details). In a future release, the default value for `weights_on
l no longer be allowed to be loaded via this mode unless they are explicitly allowlisted by
 where you don't have full control of the loaded file. Please open an issue on GitHub for an
  ckpt = torch.load(os.path.join(checkpoint_dir, "consolidated.00.pth"), map_location="cpu")
WARNING:root:Loading data...
WARNING:root:Formatting inputs...
WARNING:root:Tokenizing inputs... This may take some time...
WARNING:root:Truncating sequence from 267 to 256
WARNING:root:Truncating sequence from 315 to 256
WARNING:root:Truncating sequence from 353 to 256
WARNING:root:Truncating sequence from 259 to 256
WARNING:root:Truncating sequence from 260 to 256
WARNING:root:Truncating sequence from 257 to 256
WARNING:root:Truncating sequence from 264 to 256
/home1/tmjoshi/ml-systems-final-project-BaloneyGit-main/ml-systems-final-project-BaloneyGit-
ler('cuda', args...)` instead.
  scaler = GradScaler() if mixed_p else None
/home1/tmjoshi/ml-systems-final-project-BaloneyGit-main/ml-systems-final-project-BaloneyGit-
'cuda', args...)` instead.
  with autocast():
Epoch  0 | Step   0/100 | loss = 2.9041
Epoch  0 | Step  20/100 | loss = 2.6363
Epoch  0 | Step  40/100 | loss = 2.8797
Epoch  0 | Step  60/100 | loss = 1.4836
Epoch  0 | Step  80/100 | loss = 1.7777
Epoch  1 | Step   0/100 | loss = 3.3331
Epoch  1 | Step  20/100 | loss = 1.5306
Epoch  1 | Step  40/100 | loss = 1.4515
Epoch  1 | Step  60/100 | loss = 1.8504
Epoch  1 | Step  80/100 | loss = 2.3228
Epoch  2 | Step   0/100 | loss = 1.8226
Epoch  2 | Step  20/100 | loss = 2.1522
Epoch  2 | Step  40/100 | loss = 1.9466
Epoch  2 | Step  60/100 | loss = 2.0509
Epoch  2 | Step  80/100 | loss = 2.2993
Avg Training Time per step (seconds): 0.257
Peak memory usage: 10394.50 MB
Percentage of trainable parameters: 0.11%
```

3. GC = OFF, MP = OFF, LoRA = OFF



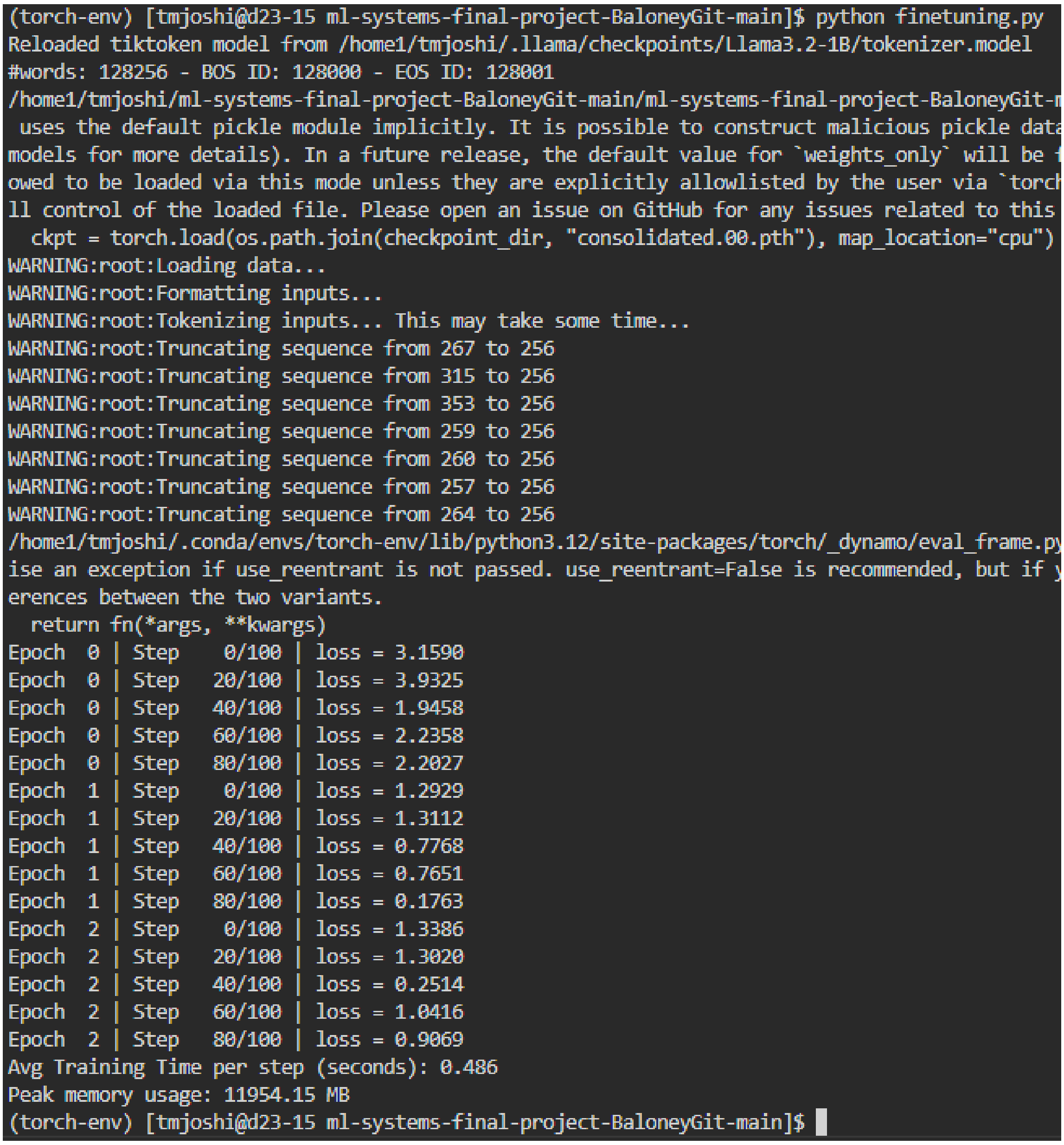
```
(torch-env) [tmjoshi@e22-16 ml-systems-final-project-BaloneyGit-main]$ python finetuning.py
Reloaded tiktoken model from /home1/tmjoshi/.llama/checkpoints/Llama3.2-1B/tokenizer.model
#words: 128256 - BOS ID: 128000 - EOS ID: 128001
/home1/tmjoshi/ml-systems-final-project-BaloneyGit-main/ml-systems-final-project-BaloneyGit-m
 which uses the default pickle module implicitly. It is possible to construct malicious pickl
d#untrusted-models for more details). In a future release, the default value for `weights_onl
l no longer be allowed to be loaded via this mode unless they are explicitly allowlisted by t
 where you don't have full control of the loaded file. Please open an issue on GitHub for any
  ckpt = torch.load(os.path.join(checkpoint_dir, "consolidated.00.pth"), map_location="cpu")
WARNING:root:Loading data...
WARNING:root:Formatting inputs...
WARNING:root:Tokenizing inputs... This may take some time...
WARNING:root:Truncating sequence from 267 to 256
WARNING:root:Truncating sequence from 315 to 256
WARNING:root:Truncating sequence from 353 to 256
WARNING:root:Truncating sequence from 259 to 256
WARNING:root:Truncating sequence from 260 to 256
WARNING:root:Truncating sequence from 257 to 256
WARNING:root:Truncating sequence from 264 to 256
Epoch  0 | Step    0/100 | loss = 2.3549
Epoch  0 | Step   20/100 | loss = 2.3454
Epoch  0 | Step   40/100 | loss = 1.2470
Epoch  0 | Step   60/100 | loss = 1.0135
Epoch  0 | Step   80/100 | loss = 2.2178
Epoch  1 | Step    0/100 | loss = 1.6635
Epoch  1 | Step   20/100 | loss = 0.6863
Epoch  1 | Step   40/100 | loss = 0.5916
Epoch  1 | Step   60/100 | loss = 1.0834
Epoch  1 | Step   80/100 | loss = 1.4844
Epoch  2 | Step    0/100 | loss = 1.0842
Epoch  2 | Step   20/100 | loss = 0.6011
Epoch  2 | Step   40/100 | loss = 1.3372
Epoch  2 | Step   60/100 | loss = 1.6391
Epoch  2 | Step   80/100 | loss = 1.3089
Avg Training Time per step (seconds): 0.336
Peak memory usage: 11953.85 MB
(torch-env) [tmjoshi@e22-16 ml-systems-final-project-BaloneyGit-main]$ 
```

4. GC = OFF, MP = ON, LoRA = OFF



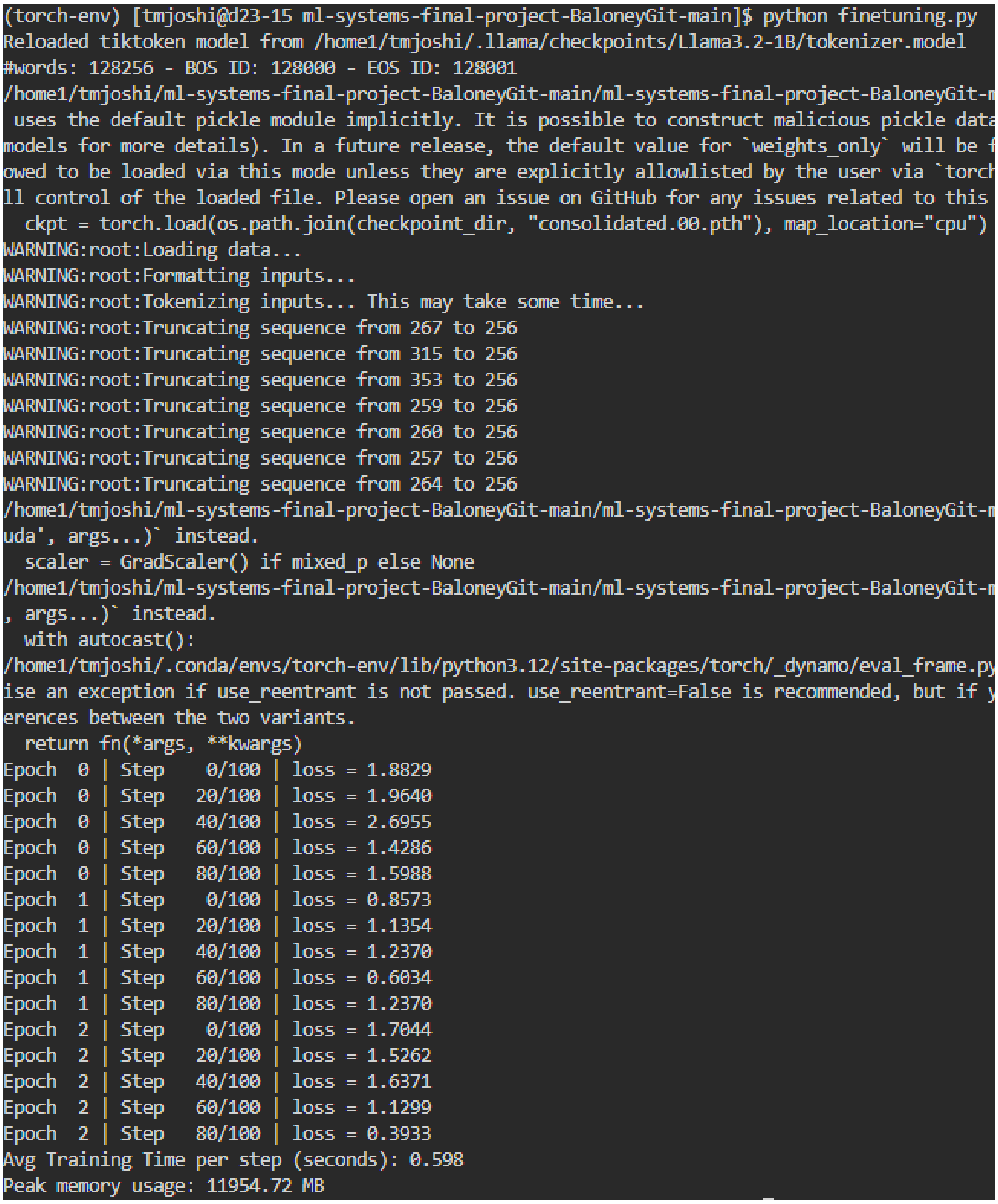
```
(torch-env) [tmjoshi@e22-16 ml-systems-final-project-BaloneyGit-main]$ python finetuning.py
Reloaded tiktoken model from /home1/tmjoshi/.llama/checkpoints/Llama3.2-1B/tokenizer.model
#words: 128256 - BOS ID: 128000 - EOS ID: 128001
/home1/tmjoshi/ml-systems-final-project-BaloneyGit-main/ml-systems-final-project-BaloneyGit-ma
 which uses the default pickle module implicitly. It is possible to construct malicious pickle
d#untrusted-models for more details). In a future release, the default value for `weights_only
l no longer be allowed to be loaded via this mode unless they are explicitly allowlisted by th
 where you don't have full control of the loaded file. Please open an issue on GitHub for any
  ckpt = torch.load(os.path.join(checkpoint_dir, "consolidated.00.pth"), map_location="cpu")
WARNING:root:Loading data...
WARNING:root:Formatting inputs...
WARNING:root:Tokenizing inputs... This may take some time...
WARNING:root:Truncating sequence from 267 to 256
WARNING:root:Truncating sequence from 315 to 256
WARNING:root:Truncating sequence from 353 to 256
WARNING:root:Truncating sequence from 259 to 256
WARNING:root:Truncating sequence from 260 to 256
WARNING:root:Truncating sequence from 257 to 256
WARNING:root:Truncating sequence from 264 to 256
/home1/tmjoshi/ml-systems-final-project-BaloneyGit-main/ml-systems-final-project-BaloneyGit-ma
ler('cuda', args...)` instead.
  scaler = GradScaler() if mixed_p else None
/home1/tmjoshi/ml-systems-final-project-BaloneyGit-main/ml-systems-final-project-BaloneyGit-ma
'cuda', args...)` instead.
  with autocast():
Epoch  0 | Step    0/100 | loss = 2.7378
Epoch  0 | Step   20/100 | loss = 1.7205
Epoch  0 | Step   40/100 | loss = 1.9712
Epoch  0 | Step   60/100 | loss = 2.1658
Epoch  0 | Step   80/100 | loss = 1.8332
Epoch  1 | Step    0/100 | loss = 2.5707
Epoch  1 | Step   20/100 | loss = 1.7272
Epoch  1 | Step   40/100 | loss = 0.7156
Epoch  1 | Step   60/100 | loss = 1.8176
Epoch  1 | Step   80/100 | loss = 0.9690
Epoch  2 | Step    0/100 | loss = 0.2735
Epoch  2 | Step   20/100 | loss = 1.6054
Epoch  2 | Step   40/100 | loss = 1.2893
Epoch  2 | Step   60/100 | loss = 1.3649
Epoch  2 | Step   80/100 | loss = 0.6651
Avg Training Time per step (seconds): 0.413
Peak memory usage: 11953.69 MB
```

5. GC = ON, MP = OFF, LoRA = OFF



```
(torch-env) [tmjoshi@d23-15 ml-systems-final-project-BaloneyGit-main]$ python finetuning.py
Reloaded tiktoken model from /home1/tmjoshi/.llama/checkpoints/Llama3.2-1B/tokenizer.model
#words: 128256 - BOS ID: 128000 - EOS ID: 128001
/home1/tmjoshi/ml-systems-final-project-BaloneyGit-main/ml-systems-final-project-BaloneyGit-m
 uses the default pickle module implicitly. It is possible to construct malicious pickle data
models for more details). In a future release, the default value for `weights_only` will be f
owed to be loaded via this mode unless they are explicitly allowlisted by the user via `torch
ll control of the loaded file. Please open an issue on GitHub for any issues related to this
  ckpt = torch.load(os.path.join(checkpoint_dir, "consolidated.00.pth"), map_location="cpu")
WARNING:root:Loading data...
WARNING:root:Formatting inputs...
WARNING:root:Tokenizing inputs... This may take some time...
WARNING:root:Truncating sequence from 267 to 256
WARNING:root:Truncating sequence from 315 to 256
WARNING:root:Truncating sequence from 353 to 256
WARNING:root:Truncating sequence from 259 to 256
WARNING:root:Truncating sequence from 260 to 256
WARNING:root:Truncating sequence from 257 to 256
WARNING:root:Truncating sequence from 264 to 256
/home1/tmjoshi/.conda/envs/torch-env/lib/python3.12/site-packages/torch/_dynamo/eval_frame.py
ise an exception if use_reentrant is not passed. use_reentrant=False is recommended, but if y
erences between the two variants.
  return fn(*args, **kwargs)
Epoch  0 | Step    0/100 | loss = 3.1590
Epoch  0 | Step   20/100 | loss = 3.9325
Epoch  0 | Step   40/100 | loss = 1.9458
Epoch  0 | Step   60/100 | loss = 2.2358
Epoch  0 | Step   80/100 | loss = 2.2027
Epoch  1 | Step    0/100 | loss = 1.2929
Epoch  1 | Step   20/100 | loss = 1.3112
Epoch  1 | Step   40/100 | loss = 0.7768
Epoch  1 | Step   60/100 | loss = 0.7651
Epoch  1 | Step   80/100 | loss = 0.1763
Epoch  2 | Step    0/100 | loss = 1.3386
Epoch  2 | Step   20/100 | loss = 1.3020
Epoch  2 | Step   40/100 | loss = 0.2514
Epoch  2 | Step   60/100 | loss = 1.0416
Epoch  2 | Step   80/100 | loss = 0.9069
Avg Training Time per step (seconds): 0.486
Peak memory usage: 11954.15 MB
(torch-env) [tmjoshi@d23-15 ml-systems-final-project-BaloneyGit-main]$
```

6. GC = ON, MP = ON, LoRA = OFF



```
(torch-env) [tmjoshi@d23-15 ml-systems-final-project-BaloneyGit-main]$ python finetuning.py
Reloaded tiktoken model from /home1/tmjoshi/.llama/checkpoints/Llama3.2-1B/tokenizer.model
#words: 128256 - BOS ID: 128000 - EOS ID: 128001
/home1/tmjoshi/ml-systems-final-project-BaloneyGit-main/ml-systems-final-project-BaloneyGit-m
 uses the default pickle module implicitly. It is possible to construct malicious pickle data
models for more details). In a future release, the default value for `weights_only` will be f
owed to be loaded via this mode unless they are explicitly allowlisted by the user via `torch
ll control of the loaded file. Please open an issue on GitHub for any issues related to this
  ckpt = torch.load(os.path.join(checkpoint_dir, "consolidated.00.pth"), map_location="cpu")
WARNING:root:Loading data...
WARNING:root:Formatting inputs...
WARNING:root:Tokenizing inputs... This may take some time...
WARNING:root:Truncating sequence from 267 to 256
WARNING:root:Truncating sequence from 315 to 256
WARNING:root:Truncating sequence from 353 to 256
WARNING:root:Truncating sequence from 259 to 256
WARNING:root:Truncating sequence from 260 to 256
WARNING:root:Truncating sequence from 257 to 256
WARNING:root:Truncating sequence from 264 to 256
/home1/tmjoshi/ml-systems-final-project-BaloneyGit-main/ml-systems-final-project-BaloneyGit-m
uda', args...)` instead.
  scaler = GradScaler() if mixed_p else None
/home1/tmjoshi/ml-systems-final-project-BaloneyGit-main/ml-systems-final-project-BaloneyGit-m
, args...)` instead.
  with autocast():
/home1/tmjoshi/.conda/envs/torch-env/lib/python3.12/site-packages/torch/_dynamo/eval_frame.py
ise an exception if use_reentrant is not passed. use_reentrant=False is recommended, but if y
erences between the two variants.
  return fn(*args, **kwargs)
Epoch  0 | Step    0/100 | loss = 1.8829
Epoch  0 | Step   20/100 | loss = 1.9640
Epoch  0 | Step   40/100 | loss = 2.6955
Epoch  0 | Step   60/100 | loss = 1.4286
Epoch  0 | Step   80/100 | loss = 1.5988
Epoch  1 | Step    0/100 | loss = 0.8573
Epoch  1 | Step   20/100 | loss = 1.1354
Epoch  1 | Step   40/100 | loss = 1.2370
Epoch  1 | Step   60/100 | loss = 0.6034
Epoch  1 | Step   80/100 | loss = 1.2370
Epoch  2 | Step    0/100 | loss = 1.7044
Epoch  2 | Step   20/100 | loss = 1.5262
Epoch  2 | Step   40/100 | loss = 1.6371
Epoch  2 | Step   60/100 | loss = 1.1299
Epoch  2 | Step   80/100 | loss = 0.3933
Avg Training Time per step (seconds): 0.598
Peak memory usage: 11954.72 MB
```

Out of Memory for:

1. GC = ON, MP = OFF, LoRA = ON
2. GC = ON, MP = ON, LoRA = ON

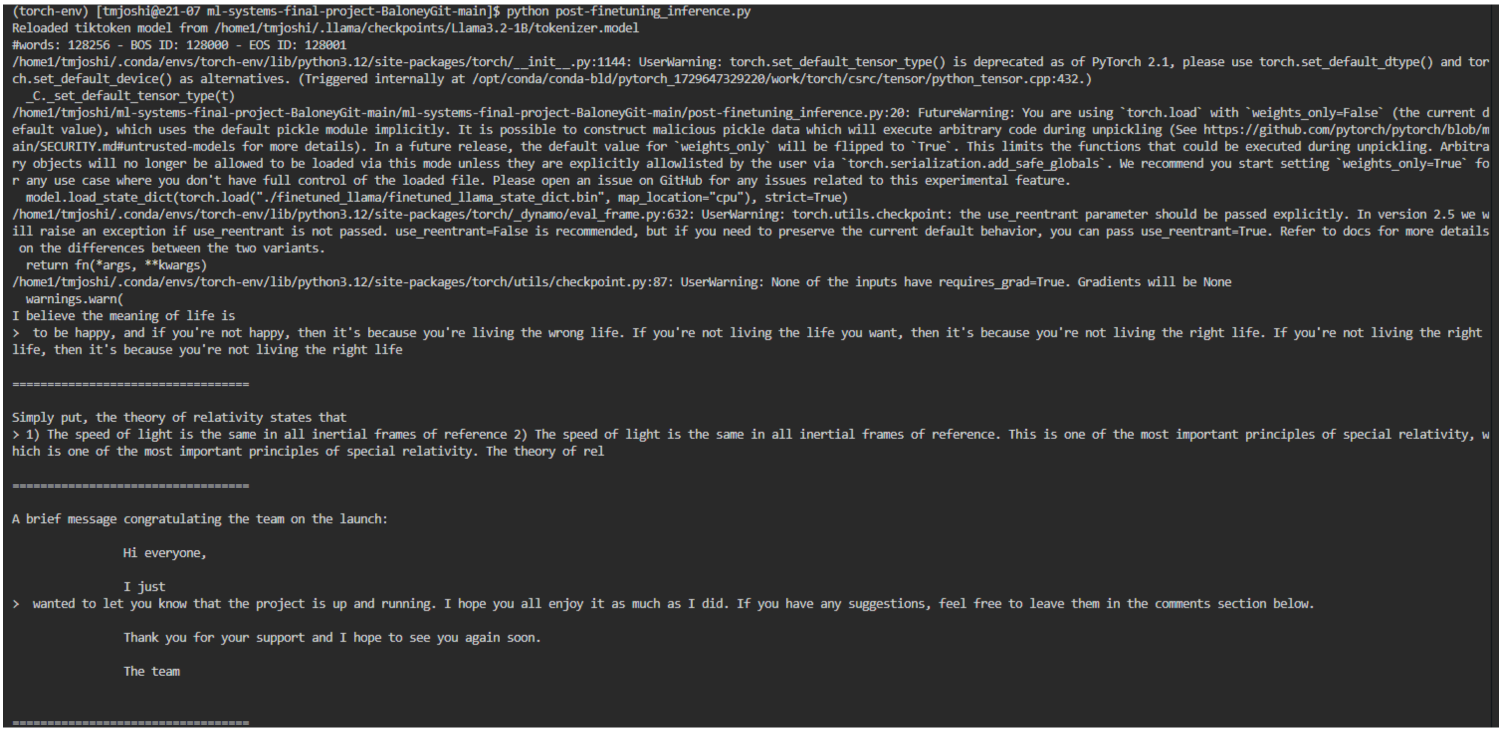
Changes made to code:

1. Entire finetuning.py python file containing:
    a. Preprocessing for pytorch DataLoader
        i. classes DataCollatorForSupervisedDataset, SupervisedDataset
        ii. functions _tokenize_fn_llama, preprocess_llama
    b. helper functions
        i. get_peak_memory_mb: for peak memory calculation
        ii. compute_shift_logits_labels: shifting logits and labels for Llama decoder
    c. finetune function:
        i. logic for:
            1. gradient accumulation
            2. mixed precision
            3. LoRA
        ii. forward pass and backprop
        iii. printing Training Time, Peak Mem usage, Percentage of trainable parameters
        iv. saving finetuned model to an output directory
2. model.py:
    a. For LoRA:
        i. Linear projection layer for Q and K (nn.Linear) changed to LoRALinear
    b. For gradient checkpointing:
        i. Checkpoint entire forward pass for Feedforward
        ii. Checkpoint TransformerBlock attention and feed forward layer
3. Entire lora.py file for Linear LoRA layer

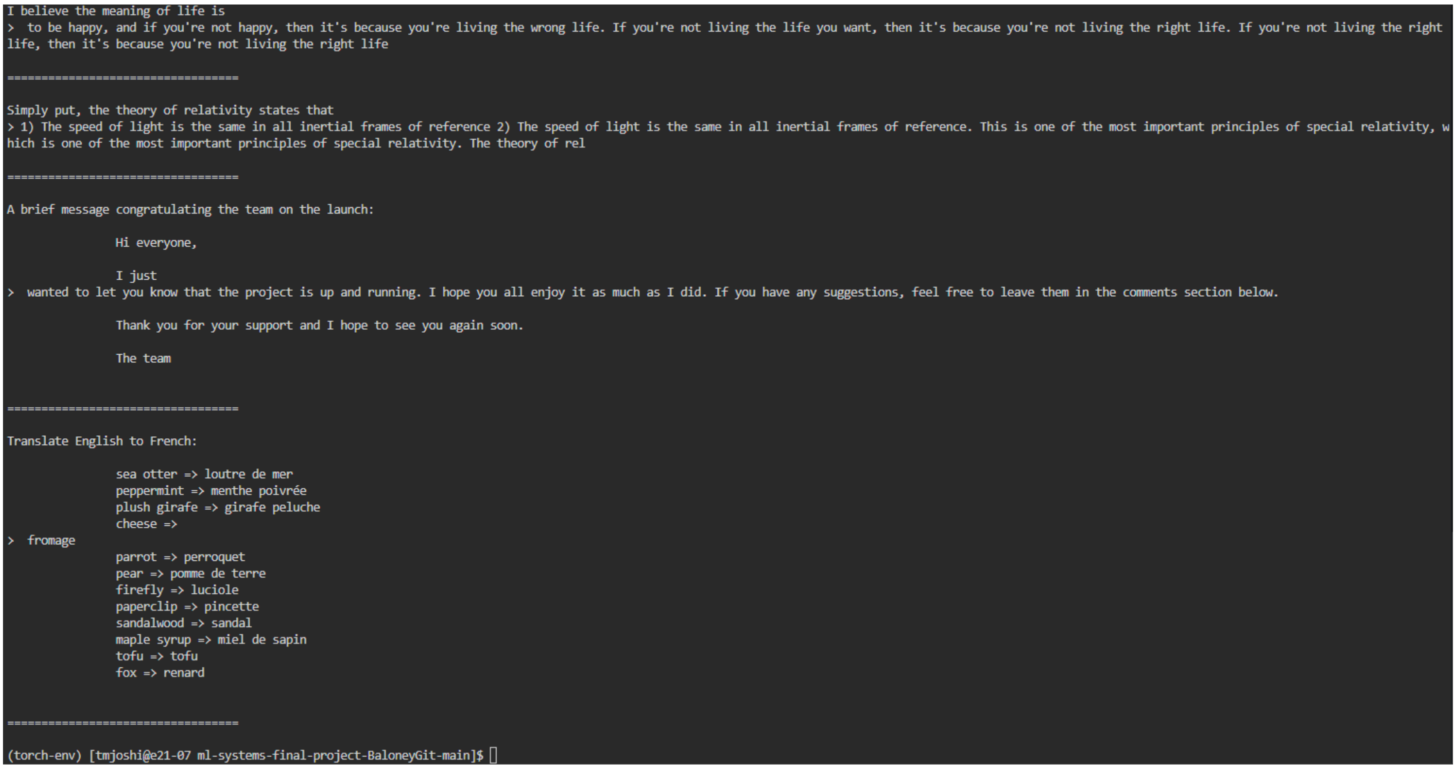
Model output comparison before and after finetuning:

Model after finetuning (Check attached screenshots below):

- More succinct text generation (eg: output for ‘A brief message congratulating the team on the launch:’)
- Better variety in language translation. Pre-finetuning language translation had examples closer to prompts. Post-finetuning have overall better variety



```
(torch-env) [tmjoshi@e21-07 ml-systems-final-project-BaloneyGit-main]$ python post-finetuning_inference.py
Reloaded tiktoken model from /home1/tmjoshi/.llama/checkpoints/Llama3.2-1B/tokenizer.model
#words: 128256 - BOS ID: 128000 - EOS ID: 128001
/home1/tmjoshi/.conda/envs/torch-env/lib/python3.12/site-packages/torch/__init__.py:1144: UserWarning: torch.set_default_tensor_type() is deprecated as of PyTorch 2.1, please use torch.set_default_dtype() and tor
ch.set_default_device() as alternatives. (Triggered internally at /opt/conda/conda-bld/pytorch_1729647329220/work/torch/csrc/tensor/python_tensor.cpp:432.)
  _C._set_default_tensor_type(t)
/home1/tmjoshi/ml-systems-final-project-BaloneyGit-main/ml-systems-final-project-BaloneyGit-main/post-finetuning_inference.py:20: FutureWarning: You are using `torch.load` with `weights_only=False` (the current d
efault value), which uses the default pickle module implicitly. It is possible to construct malicious pickle data which will execute arbitrary code during unpickling (See https://github.com/pytorch/pytorch/blob/m
ain/SECURITY.md#untrusted-models for more details). In a future release, the default value for `weights_only` will be flipped to `True`. This limits the functions that could be executed during unpickling. Arbitra
ry objects will no longer be allowed to be loaded via this mode unless they are explicitly allowlisted by the user via `torch.serialization.add_safe_globals`. We recommend you start setting `weights_only=True` fo
r any use case where you don't have full control of the loaded file. Please open an issue on GitHub for any issues related to this experimental feature.
  model.load_state_dict(torch.load("./finetuned_llama/finetuned_llama_state_dict.bin", map_location="cpu"), strict=True)
/home1/tmjoshi/.conda/envs/torch-env/lib/python3.12/site-packages/torch/_dynamo/eval_frame.py:632: UserWarning: torch.utils.checkpoint: the use_reentrant parameter should be passed explicitly. In version 2.5 we w
ill raise an exception if use_reentrant is not passed. use_reentrant=False is recommended, but if you need to preserve the current default behavior, you can pass use_reentrant=True. Refer to docs for more details
 on the differences between the two variants.
  return fn(*args, **kwargs)
/home1/tmjoshi/.conda/envs/torch-env/lib/python3.12/site-packages/torch/utils/checkpoint.py:87: UserWarning: None of the inputs have requires_grad=True. Gradients will be None
  warnings.warn(
I believe the meaning of life is
>  to be happy, and if you're not happy, then it's because you're living the wrong life. If you're not living the life you want, then it's because you're not living the right life. If you're not living the right
life, then it's because you're not living the right life

==================================

Simply put, the theory of relativity states that
> 1) The speed of light is the same in all inertial frames of reference 2) The speed of light is the same in all inertial frames of reference. This is one of the most important principles of special relativity, w
hich is one of the most important principles of special relativity. The theory of rel

==================================

A brief message congratulating the team on the launch:

                Hi everyone,

                I just
>  wanted to let you know that the project is up and running. I hope you all enjoy it as much as I did. If you have any suggestions, feel free to leave them in the comments section below.

                Thank you for your support and I hope to see you again soon.

                The team


==================================
```



```
I believe the meaning of life is
>  to be happy, and if you're not happy, then it's because you're living the wrong life. If you're not living the life you want, then it's because you're not living the right life. If you're not living the right
life, then it's because you're not living the right life

==================================

Simply put, the theory of relativity states that
> 1) The speed of light is the same in all inertial frames of reference 2) The speed of light is the same in all inertial frames of reference. This is one of the most important principles of special relativity, w
hich is one of the most important principles of special relativity. The theory of rel

==================================

A brief message congratulating the team on the launch:

                Hi everyone,

                I just
>  wanted to let you know that the project is up and running. I hope you all enjoy it as much as I did. If you have any suggestions, feel free to leave them in the comments section below.

                Thank you for your support and I hope to see you again soon.

                The team


==================================
```

```
Translate English to French:

                sea otter => loutre de mer
                peppermint => menthe poivrée
                plush girafe => girafe peluche
                cheese =>
>  fromage
                parrot => perroquet
                pear => pomme de terre
                firefly => luciole
                paperclip => pincette
                sandalwood => sandal
                maple syrup => miel de sapin
                tofu => tofu
                fox => renard


==================================

(torch-env) [tmjoshi@e21-07 ml-systems-final-project-BaloneyGit-main]$ 
```