# CSE665: Large Language Models
# Assignment 3
# Fine Tuning Large Language Models

## Approach

My code implements a comprehensive approach to fine-tune Phi2 using QLoRA (Quantized Low-Rank Adaptation) on the SNLI (Stanford Natural Language Inference) dataset to improve performance on the classification task. Here's a step-by-step summary of the approach:

1. **Model and Dataset Preparation**:
    - The code sets up a PyTorch device, selecting GPU (if available) for efficient training.
    - You load the pre-trained model `microsoft/phi-2` with 4-bit quantization to reduce memory usage, using the `BitsAndBytesConfig` to configure the quantization settings.
    - The `AutoTokenizer` is initialized with padding and special tokens, followed by loading and subsampling the SNLI dataset for efficient testing and training.
2. **Initial Evaluation (Pre-trained Model)**:
    - A prompt format function (`create_prompt`) is defined to generate text prompts for each SNLI sample.
    - For each sample in the test dataset, the model generates a prediction and decodes it, checking if it matches the correct label. If incorrect, the sample details are logged in `failure_cases` to track pre-trained model errors.
    - The initial accuracy of the model on the test set is computed, providing a baseline before fine-tuning.
3. **Fine-Tuning Preparation**:
    - The model is prepared for k-bit training, enabling gradient checkpointing to save memory during training.
    - Training arguments, including batch size, learning rate, and evaluation strategy, are configured.
    - LoRA (Low-Rank Adaptation) configuration is defined, with specific attention to fine-tuning selective parts of the model (e.g., `q_proj`, `k_proj`).

- `preprocess_dataset` function is used to format and tokenize the training and validation datasets according to the prompt style, preparing them for the fine-tuning stage.

4. **Resource Logging Callback**:
   - A custom callback (`ResourceLoggingCallback`) is defined to log resource usage (CPU, GPU memory, etc.) after each epoch, providing insights into computational efficiency during training.
5. **Fine-Tuning Execution**:
   - The model is fine-tuned using the `Trainer` with LoRA and QLoRA applied, saving the fine-tuned model parameters for later evaluation.
   - Resource logging tracks hardware usage per epoch, and the total time for fine-tuning is recorded.
6. **Post-Fine-Tuning Evaluation**:
   - The code re-loads the pre-trained base model, applies the fine-tuned LoRA weights, and evaluates on the test set again.
   - Each test sample is processed through the fine-tuned model with a similar prompt structure.
   - The results are categorized into `corrected_by_finetuned` (cases corrected after fine-tuning) and `not_corrected_by_finetuned` (still incorrect after fine-tuning) by comparing predictions against the initial pre-trained model failures.
7. **Output**:
   - The approach concludes with a summary of corrected vs. uncorrected failures by the fine-tuned model, offering insights into the efficacy of fine-tuning with QLoRA for improving entailment classification.

## Accuracy before fine-tuning

```
    accuracy = correct/total
    print(f"Accuracy before fine-tuning: {accuracy * 100:.2f}%")
```

```
Accuracy before fine-tuning: 59.00%
```

## Accuracy after fine-tuning

```
    accuracy = correct/total
    print(f"Accuracy after fine-tuning: {accuracy * 100:.2f}%")

Accuracy after fine-tuning: 87.00%
```

The accuracy improvement from 59% to 87% after fine-tuning the Phi2 model on the SNLI dataset indicates significant gains in the model's ability to classify natural language inferences accurately.

## Impact of Fine-Tuning on Model Performance

- **Pre-Fine-Tuning**: With a baseline accuracy of 59%, the model likely struggled with the specific nuances of entailment, contradiction, and neutrality in the SNLI dataset. This lower accuracy suggests that the model had limited exposure to the patterns and subtleties present in natural language inference (NLI) tasks, possibly due to a different or more generalized pre-training dataset.
- **Post-Fine-Tuning**: Achieving 87% accuracy after five epochs indicates that fine-tuning on SNLI allowed the model to capture the semantic relationships more effectively. Fine-tuning on SNLI provided task-specific examples that helped the model adapt, focusing on entailment-based language structures and the intricacies of contradiction and neutrality.

## Time taken to fine-tune the model using QLoRA.

```
start_time = time.time()
peft_trainer.train()
end_time = time.time()
```

```
    print(f"Time taken to fine-tune the model using QLoRA: {(end_time - start_time) / 60:.2f} minutes")

Time taken to fine-tune the model using QLoRA: 22.89 minutes
```

## Total parameters in the model and the number of parameters fine-tuned

```
    total_params = peft_model.num_parameters()
    trainable_params = sum(p.numel() for p in peft_model.parameters() if p.requires_grad)
    print(f"Total parameters in the model: {total_params:,}")
    print(f"Number of parameters fine-tuned: {trainable_params:,}")
```

```
Total parameters in the model: 2,800,655,360
Number of parameters fine-tuned: 20,971,520
```

## Resources used during fine-tuning

```
End of Epoch 1.0/5
Trainer.tokenizer is now deprecated. You should use Trainer.processing_class instead.
GPU Usage - Memory Allocated: 2.55 GB, Memory Reserved: 3.41 GB, Utilization: 0%
CPU Usage: 3.5%, Memory Usage: 2.79 GB

End of Epoch 2.0/5
Trainer.tokenizer is now deprecated. You should use Trainer.processing_class instead.
GPU Usage - Memory Allocated: 2.55 GB, Memory Reserved: 4.15 GB, Utilization: 0%
CPU Usage: 3.5%, Memory Usage: 2.80 GB

End of Epoch 3.0/5
Trainer.tokenizer is now deprecated. You should use Trainer.processing_class instead.
GPU Usage - Memory Allocated: 2.55 GB, Memory Reserved: 4.15 GB, Utilization: 0%
CPU Usage: 2.0%, Memory Usage: 2.81 GB

End of Epoch 4.0/5
Trainer.tokenizer is now deprecated. You should use Trainer.processing_class instead.
GPU Usage - Memory Allocated: 2.55 GB, Memory Reserved: 4.15 GB, Utilization: 0%
CPU Usage: 3.0%, Memory Usage: 2.82 GB

End of Epoch 5.0/5
Trainer.tokenizer is now deprecated. You should use Trainer.processing_class instead.
GPU Usage - Memory Allocated: 2.55 GB, Memory Reserved: 4.15 GB, Utilization: 0%
CPU Usage: 3.5%, Memory Usage: 2.99 GB
```

During the fine-tuning process, the GPU memory usage was consistent, with 2.55 GB allocated throughout all five epochs, and the reserved memory increased slightly after the first epoch from 3.41 GB to 4.15 GB. This level of GPU memory usage is relatively low for a fine-tuning task, suggesting that the model is either lightweight or has been optimized to use minimal resources. However, despite GPU memory allocation, the reported GPU utilization remains at 0%, which is unusual. This indicates that the GPU may not have been actively performing the fine-tuning computations, possibly due to configuration issues, such as the model defaulting to the CPU instead of the GPU for processing.

On the CPU side, the utilization was low, around 2-3.5%, with memory usage remaining stable at approximately 2.8-2.9 GB across epochs. This usage pattern suggests that the CPU handled minor tasks, such as data loading and preprocessing, rather than intensive computations. Additionally, the log shows a repeated deprecation warning: "Trainer.tokenizer is now deprecated," suggesting outdated syntax. Updating the tokenizer to the recommended Trainer.processing_class syntax could prevent compatibility issues in future software versions. Overall, the system resources appear underutilized, particularly the GPU, and ensuring the model properly leverages the GPU could improve training efficiency.

## Failure cases before fine-tuning


```
[{'premise': 'This church choir sings to the masses as they sing joyous songs from the book at a church.',
  'hypothesis': 'The church has cracks in the ceiling.',
  'predicted_label': 2,
  'actual_label': 1},
 {'premise': 'A woman within an orchestra is playing a violin.',
  'hypothesis': 'A woman is playing the violin.',
  'predicted_label': 1,
  'actual_label': 0},
 {'premise': 'Two men climbing on a wooden scaffold.',
  'hypothesis': 'Two sad men climbing on a wooden scaffold.',
  'predicted_label': 0,
  'actual_label': 1},
 {'premise': 'A group of people stand near and on a large black square on the ground with some yellow writing on it.',
  'hypothesis': 'a group of people wait',
  'predicted_label': 0,
  'actual_label': 1},
 {'premise': 'A Skier ski-jumping while two other skiers watch his act.',
  'hypothesis': 'A skier preparing a trick',
  'predicted_label': 1,
  'actual_label': 0},
 {'premise': 'Children bathe in water from large drums.',
  'hypothesis': 'The kids are wet.',
  'predicted_label': 1,
  'actual_label': 0},
```


## Pre-trained failure cases corrected by fine-tuning

```
[{'premise': 'A woman within an orchestra is playing a violin.',
  'hypothesis': 'A woman is playing the violin.',
  'pretrained_predicted_label': 1,
  'fine_tuned_predicted_label': 0,
  'actual_label': 0},
 {'premise': 'A group of people stand near and on a large black square on the ground with some yellow writing on it.',
  'hypothesis': 'a group of people wait',
  'pretrained_predicted_label': 0,
  'fine_tuned_predicted_label': 1,
  'actual_label': 1},
 {'premise': 'Children bathe in water from large drums.',
  'hypothesis': 'The kids are wet.',
  'pretrained_predicted_label': 1,
  'fine_tuned_predicted_label': 0,
  'actual_label': 0},
 {'premise': 'A man is renovating a room.',
  'hypothesis': 'A man is using a hammer in a room.',
  'pretrained_predicted_label': 0,
  'fine_tuned_predicted_label': 1,
  'actual_label': 1},
 {'premise': 'An Ambulance is passing a man wearing a bandanna and girl.',
  'hypothesis': 'The man in the bandana is running after the ambulance',
  'pretrained_predicted_label': 1,
  'fine_tuned_predicted_label': 2,
  'actual_label': 2},
 {'premise': 'The Sooner football player carrying the ball is trying to avoid being tackled.',
  'hypothesis': 'A football player is holding a ball.',
  'pretrained_predicted_label': 1,
  'fine_tuned_predicted_label': 0,
  'actual_label': 0},
```


## Pre-trained failure cases not corrected by fine-tuning


```
[{'premise': 'This church choir sings to the masses as they sing joyous songs from the book at a church.',
  'hypothesis': 'The church has cracks in the ceiling.',
  'pretrained_predicted_label': 2,
  'fine_tuned_predicted_label': 2,
  'actual_label': 1},
 {'premise': 'Two men climbing on a wooden scaffold.',
  'hypothesis': 'Two sad men climbing on a wooden scaffold.',
  'pretrained_predicted_label': 0,
  'fine_tuned_predicted_label': 2,
  'actual_label': 1},
 {'premise': 'A Skier ski-jumping while two other skiers watch his act.',
  'hypothesis': 'A skier preparing a trick',
  'pretrained_predicted_label': 1,
  'fine_tuned_predicted_label': 1,
  'actual_label': 0},
 {'premise': 'A woman is standing near three stores, two have beautiful artwork and the other store has Largo written on it.',
  'hypothesis': 'A woman standing on a street corner outside beside three different stores, two of which contain beautiful artwork
  'pretrained_predicted_label': 1,
  'fine_tuned_predicted_label': 1,
  'actual_label': 0},
 {'premise': 'An older gentleman wearing a hat is walking on crutches next to a busy street.',
  'hypothesis': 'A man with a walking stick is next to the street.',
  'pretrained_predicted_label': 1,
  'fine_tuned_predicted_label': 1,
  'actual_label': 2},
 {'premise': 'Two middle-aged police officers watch over a parking lot, at night.',
  'hypothesis': 'The officers are actually security guards.',
  'pretrained_predicted_label': 1,
  'fine_tuned_predicted_label': 1,
  'actual_label': 2},
```

## Corrected by Fine-Tuning

**Premise:** "A woman within an orchestra is playing a violin."
**Hypothesis:** "A woman is playing the violin."
**Pre-trained Prediction:** Contradiction (1)
**Fine-tuned Prediction:** Entailment (0)
**Explanation:** The pre-trained model likely struggled to recognize the core semantic similarity here, as "within an orchestra" is not essential to the hypothesis. Fine-tuning helped the model focus on the main action, correctly identifying this as entailment.

**Premise:** "A group of people stand near and on a large black square on the ground with some yellow writing on it."
**Hypothesis:** "A group of people wait."
**Pre-trained Prediction:** Entailment (0)
**Fine-tuned Prediction:** Contradiction (1)
**Explanation:** The fine-tuned model recognized that simply "standing" doesn't necessarily imply "waiting," which was missed by the pre-trained model. Fine-tuning likely helped clarify nuanced distinctions between similar states or actions.

**Premise:** "Children bathe in water from large drums."
**Hypothesis**: "The kids are wet."
**Pre-trained Prediction:** Contradiction (1)
**Fine-tuned Prediction:** Entailment (0)
**Explanation:** The fine-tuned model correctly inferred that children bathing would indeed make them wet. The pre-trained model may have missed this due to a lack of real-world inference ability, which fine-tuning improved.

## Not Corrected by Fine-Tuning

**Premise:** "This church choir sings to the masses as they sing joyous songs from the book at a church."
**Hypothesis**: "The church has cracks in the ceiling."
**Pre-trained and Fine-tuned Prediction:** Neutral (2)
**Actual:** Contradiction (1)
**Explanation:** The model likely fails to connect that there is no indication of cracks in the ceiling in the premise. Both models might struggle with distant, implicit logical relationships that don't directly contradict or entail.

**Premise:** "Two men climbing on a wooden scaffold."
**Hypothesis:** "Two sad men climbing on a wooden scaffold."
**Pre-trained Prediction:** Entailment (0)
**Fine-tuned Prediction:** Neutral (2)
**Actual:** Contradiction (1)
Explanation: Both models struggled to detect that the addition of "sad" introduces a potential contradiction without any indication of emotion in the premise. Fine-tuning didn't fully equip the model to pick up subtle sentiment mismatches.

**Premise:** "A Skier ski-jumping while two other skiers watch his act."
**Hypothesis:** "A skier preparing a trick."
**Pre-trained and Fine-tuned Prediction:** Contradiction (1)
**Actual:** Entailment (0)
**Explanation:** The models fail to understand that "preparing a trick" could naturally relate to "ski-jumping." This suggests an area where the model struggles with action preparation versus action execution, which fine-tuning alone did not fully address.