MSAI-337, Spring 2025
Homework #3: Parameter Efficient Fine-tuning
Due Date: Sunday, June 8th @ 11:59PM
Total Points: 10.0

Description: In this assignment, you will explore three parameter efficient fine-tuning methods on the OpenBookQA dataset (see https://arxiv.org/abs/1809.02789) using a pre-trained BERT model. Please see Homework #2 for details on the OpenBookQA multiple-choice question-answering task.

Tasks:

1. **Full Fine-tuning Baseline (1.0 pts):** To establish a baseline for comparison with your parameter efficient fine-tuned models, you must use the BERT-base-uncased checkpoint for the BERT Model (see https://huggingface.co/docs/transformers/v4.26.1/en/model_doc/bert) and fine-tune the model on the training set using your own code. Implementations using BertForMultipleChoice or other specialized variants will not be accepted. The format of the text input to BERT and classification methodology for this task is up to you. One approach is to leverages the `[CLS]` token, that can be implemented as:

    a. Encode each instance as
    ```
    [CLS] <fact> <stem> <text of choice #1> [END]
    [CLS] <fact> <stem> <text of choice #2> [END]
    [CLS] <fact> <stem> <text of choice #3> [END]
    [CLS] <fact> <stem> <text of choice #4> [END]
    ```
    b. Retrieve the context embedding at the top of the transformer stack for the `[CLS]` token and take the dot product with a linear layer.
    c. Treat the resulting vector as logits, apply the softmax and compute a multiclass cross-entropy loss over the four choices.
    d. Fine-tune BERT on the complete dataset with this encoding for multiple epochs.
    e. Use the trained model to perform inference on the validation and test sets.
    f. Report accuracy on the OpenBookQA the validation and test set, number of model parameters,fine-tuning time and inference time.
    g. Please describe your approach sufficiently so that someone familiar with transformers can replicate your results.

    Please discuss any limitations of your approach and possible solutions. Using this approach described, I was able to achieve an accuracy of 55.9% on the test set.

2. **Adapter Fine-tuning (3.0 pts):** Instead of fine-tuning the full BERT model, freeze all pre-trained weights and embeddings, insert adapters and fine-tune adapter weights only using the methodology described above. The adapter architecture should be consistent with (arxiv.org/abs/1902.00751). Where you insert the adapters and their hyper-parameters is your choice - you may want to tune these decisions on the validation set.

    a. report accuracy on the OpenBookQA test set, the number of model parameters fine-tuned, fine-tuning time and inference time.
    b. Please describe your approach sufficiently so that someone familiar with transformers can replicate your results.
    c. provide code snippets that show where and how you implemented adapters.
    d. discuss your adapter fine-tuning results relative to your full fine-tuning results.

3. **LoRA Fine-tuning (3.0 pts):** Instead of fine-tuning the full BERT model, freeze all pre-trained weights and embeddings, implement LoRA and fine-tune these matrix weights only using the methodology described above. The LoRA implementation should be consistent with (arxiv.org/abs/2106.09685). Where you implement LoRA and their hyper-parameters is your choice - you may want to tune these decisions on the validation set.
    a. report accuracy on the OpenBookQA test set, the number of model parameters fine-tuned, fine-tuning time and inference time.
    b. Please describe your approach sufficiently so that someone familiar with transformers can replicate your results.
    c. provide code snippets that show where and how you implemented LoRA.
    d. discuss your LoRA fine-tuning results relative to your full fine-tuning results.

4. **Prefix-tuning (3.0 pts):** Instead of fine-tuning the full BERT model, freeze all pre-trained weights and embeddings, and implement prefix-tuning and fine-tune only the prefix embeddings using the methodology described above. The prefix implementation should be consistent with (arxiv.org/abs/2101.00190). The number of prefixes that you use and where you insert them your choice - you may want to tune these decisions on the validation set.
    a. report accuracy on the OpenBookQA test set, the number of model parameters fine-tuned, fine-tuning time and inference time.
    b. Please describe your approach sufficiently so that someone familiar with transformers can replicate your results.
    c. provide code snippets that show where and how you implemented prefix-tuning.
    d. discuss your prefix-tuning results relative to your full fine-tuning results.

Note: It would be helpful to present all numerical results in a single table to facilitate comparisons.

What to Submit: Please submit a single .zip file for the group. The zip file should contain the following:

- A single .pdf for all written responses and results.
- Individual .py files (three files maximum) of your code snippets for adapter, LoRA and prefix approaches.

Only one group member needs to submit.