Assignment # 3

Generative AI **Deadline: November 16, 2024**

Instructor: Dr. Akhtar Jamil

Instructions

Note: There is no extension in the deadline. Please submit on or before the deadline. There is a **grace time of 2 hours** after the submission deadline expires. You must verify that your submissions are correct. Any submission received after this slack time will be considered late and NO marks will be awarded. There should be no comments provided on each line of code. You must provide detailed comments about functions that you create; that is enough to understand the function.

Deliverables:

1. Original Source file, including all code.

2. **Technical Report.** It must be written in Overleaf. Use IEEE Conference Paper Format for composing the report.

IEEE Conference Paper Format

Your report should include:

1. Introduction: A brief overview of the assignment.

2. Methodology: Description of the dataset, preprocessing steps, and model architecture.

3. Results: Training and validation loss and accuracy, examples of completed sentences.

4. Discussion: Analysis of sentence coherence, how the model's predictions improve over time, and any challenges encountered.

5. Conclusion: A summary of your findings.

6. Prompts

7. References

ZIP File Submission

Put all your files inside one folder and name it according to the following convention:

RollNo_Name_Ass1.ZIP

You must upload this ZIP file

Question # 1 Transformer Implementation for Machine Translation

Objective

Implement a Transformer model for machine translation using the UMC005: English-Urdu Parallel Corpus. The task involves training a model that can translate English sentences into Urdu. The assignment will focus on understanding the core components of the Transformer architecture and evaluating its performance on the given dataset.

The following tasks are required the least:

- Preprocess the dataset by cleaning and tokenizing the text. Ensure that English and Urdu texts are properly aligned.
- Use suitable tokenizers for English and Urdu. You can experiment with Byte Pair Encoding (BPE) or other suitable tokenization techniques.
- Implement the Transformer model from scratch using a deep learning framework like TensorFlow or PyTorch. The architecture should include an encoder and a decoder with multiple layers.
- Tune hyperparameters such as the number of layers, hidden units, heads in multi-head attention, dropout rates, learning rates, and batch size. Use techniques like early stopping and learning rate scheduling to improve training efficiency.
- Evaluate the trained model on the test set using translation quality metrics BLEU Score and ROUGE (mandatory). You can add additional scores if you want (optional)
- Comparative Analysis: Implement an LSTM model for translation using the same dataset.
- Compare the performance of two models in terms of their translation accuracy, training time, memory usage, inference speed, perplexity.
- Deploy your model on a local machine or cloud and create a simple GUI like ChatGPT UI. The user will be able to input English language text which will be aligned to the left, then press enter, and the model will translate the text. The translated Urdu text should appear below the source sentence and be right-aligned. The source and target language texts' history should remain visible to the user, similar to ChatGPT.
- Plot training and validation loss curves to illustrate the training process to see if the model converges effectively.
- Present detailed results in a report, summarizing the methodology, findings, and challenges.
- Provide the code in a well-documented format, clearly explaining each step.
- Implement attention visualization to show which parts of the English sentence the model focused on when generating the Urdu translation.
- **Bonus:** Experiment with fine-tuning a pretrained model and compare the results with your custom-trained model.

UMC005: English-Urdu Parallel Corpus

UMC005 English-Urdu is a parallel corpus of texts in English and Urdu language with sentence

alignments. The corpus can be used for experiments with statistical machine translation. For details, please visit the following website:

English Urdu Dataset

Question # 2 Vision Transformer for CIFAR-10 Image Classification

Objective

Implement a Vision Transformer (ViT) for CIFAR-10 Image Classification. The goal is to classify images from the CIFAR-10 dataset using ViT and compare its performance with other models. This assignment will involve evaluating the effectiveness of the Transformer-based approach for image classification and comparing it with traditional and hybrid architectures.

The following tasks are required the least:

- Preprocess the CIFAR-10 dataset, ensuring that the images are properly normalized and prepared for training. Use standard data augmentation techniques like random cropping, flipping, and normalization to enhance dataset which may be useful for model performance.
- Implement the Vision Transformer (ViT) from scratch using a deep learning framework like TensorFlow or PyTorch. Ensure to divide the images into patches and apply positional encoding for input to the Transformer.
- Tune hyperparameters for ViT, such as the number of Transformer layers, number of heads in multi-head attention, learning rate, batch size, and patch size. Utilize techniques like early stopping and learning rate scheduling to optimize training.
- Evaluate the model using accuracy, precision, recall and f1-score. Visualize the results of test images using confusion matrix.
- For comparision, implement a hybrid architecture where features from image patches are first learned using a Convolutional Neural Network (CNN) and then fed into a Multi-Layer Perceptron (MLP) instead of using a Transformer.
- Implement a ResNet model for image classification on the same dataset. Instead of training a ResNet model from scratch on the CIFAR-10 dataset, utilize a pretrained ResNet and apply transfer learning technique. Use a ResNet model that was trained on a large-scale image dataset. Freeze the pretrained layers (feature extraction), and only train and fine-tune the additional classifier layers added to fine tune on the CIFAR-10 dataset.
- Compare the performance of the three models (ResNet, ViT, and the hybrid architecture) based on classification accuracy, training time, memory usage, and inference speed. You can also include additional metrics if you want (optional).
- Deploy the models on a local machine or cloud and visualize classification results for a set of test images. Include visualizations of correct and incorrect classifications.

- Plot training and validation accuracy and loss curves to illustrate the training process and observe if the models converge effectively.
- Present detailed results in a report, summarizing the methodology, findings, and challenges. Clearly explain the strengths and weaknesses of each architecture.
- Provide the code in a well-documented format, clearly explaining each step.

Data Set

CIFAR-10 Dataset