Assignment 3: Transformer

Due date: 8th November, 2023 (11.59 pm)

Submission Guidelines

1. The assignment should be submitted in a written format (e.g., a report or essay). Please do not include screenshot of code in the report.
2. Code should be submitted along with the report in pdf format.
3. Include a list of references and resources used for the assignment.

Grading Criteria

- Understanding of the Transformer architecture.
- Quality of explanations and discussions in Tasks 1 and 2.
- Implementation and results in Task 2

Task 1: Summary of the Model (25 marks)

In this question, you are asked to review and summarize the contents of a tutorial on the Transformer architecture. The tutorial you provided is a Colab notebook link that explains Transformers, and you need to understand its contents and provide a summary.

Guidance:

- Go through the tutorial carefully, step by step.
- Understand the key concepts and components of the Transformer architecture, including self-attention mechanisms and positional encoding.
- Make concise notes on the main takeaways from the tutorial.
- Ensure your summary is well-organized, easy to follow, and free from plagiarism.

Task 2: Practical Applications - Next Word Prediction with Pretrained Transformers (25 marks)

This question is more involved and requires practical implementation and explanation. You have two main components to address:

2.1 Explore Creativity (Results: 5 marks)

- Experiment with 10 different prompts and approaches to explore the creativity of the model.
- Use various prompts like "Write a poem about..." or "Explain the concept of..." to generate text and assess the model's creative abilities.

2.2 Coding (15 marks)

- To generate text using a pretrained language model from Hugging Face (e.g., GPT-3), follow the provided steps in your Python script or Jupyter Notebook.
- Install the required libraries, import necessary modules, choose a pretrained model, create a text generation pipeline, generate text, and display/analyze the generated text.
- Ensure your code is well-documented, and the model's outputs are captured effectively.

To generate text using a pretrained language model from Hugging Face (e.g., GPT-3), you can use the Transformers library, which provides a simple and convenient way to interact with a variety of pre-trained models. Here's a step-by-step guide to generate text and explore the creativity of the model:

1. Install Required Libraries: Make sure you have the Hugging Face Transformers library installed. You can install it using pip:
2. Import the Necessary Modules: In your Python script or Jupyter Notebook, import the modules you need. For text generation, you will use the pipeline module from the Transformers library:
3. Choose a Pretrained Model: Hugging Face provides a variety of pretrained models for text generation, such as GPT-2 and GPT-3. You can explore their model hub to choose a model that fits your needs: Hugging Face Model Hub
4. Create a Text Generation Pipeline: Initialize a text generation pipeline for the chosen model:
5. Generate Text: Use the pipeline to generate text. You can provide a prompt or let the model generate text without a prompt:
6.Display and Analyze Generated Text: Print or analyze the generated text to see the model's creative output. You can use the generated_text variable to access the generated content.

2.3 Explain the detailed methodology, including model architecture, hyperparameters, and results briefly in the report.

In your report, you need to explain the methodology you followed in section 2.2 in detail. This should include:

- Details about the model architecture (e.g., BERT or GPT-1).
- Any hyperparameters you used (e.g., temperature or max length for text generation).
- Describe the results of your text generation experiments. Discuss the creativity of the model, whether it generated coherent and contextually relevant text, and any interesting observations.
