Assignment 1: Implementation of Mini Chat-GPT

Lecturer: Dr. Wan-Lei Zhao
Xiamen University

1 Problem

In nowadays, Chat-GPT like tools are increasingly popular. Essentially, it is built upon a transformer decoder. It is trained to predict the next word of a sentence, given the leading block of a sentence is fed to the model. When the model is trained with billions of documents from various sources, it shows human-like intelligence when it responds to human questions or input prompts.

Figure 1 shows the general framework that how the autoregressive transformer decoder works. In each round of the loop, the decoder will generate the next word based on the produced logits. This newly generated word (expressed as a token) will be appended to the generated words so far to support the generation of new word in the next round. In our case, the decoder is fed with the words of user "prompt" at the beginning. The newly generated words are appended to the user's prompt to iteratively generate next new words until the generated new word is "<EOS>".

In this project, you are provided with the framework of a mini Chat-GPT, namely the codes for the model, for the training as well as for the chat interface. Some blocks of the codes (in the model and training) are deliberately omitted. You are required to fill these missing code lines and train a model out of the provided mini text corpus. The sections that you are required to fill the code lines are given as follows.

```
1 #fill your codes here
```

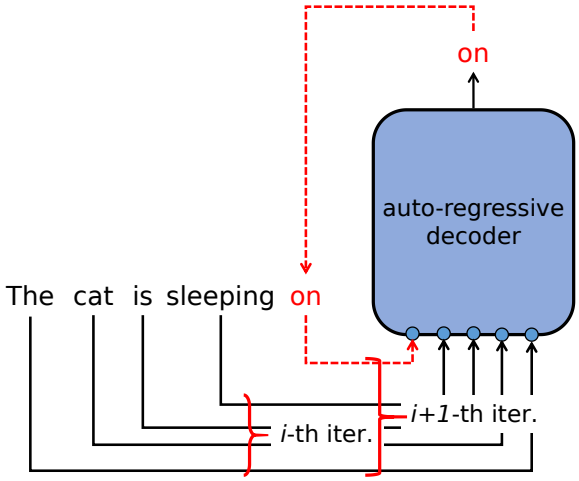


Fig. 1: The next-word generation loop in Chat-GPT.

You are NOT allowed to modify the structure or variable name of the project.

2 Hints

In order to fulfill the assignment, you are recommended to install Python 3.1.2 or later along with "pytorch", "numpy" packages. It is not necessary to have an NVIDIA card, while it may speed up the training a lot. It only takes around 50 minutes to train a model on Nvidia RTX 2080 when the epoch is set to 70. To run the training, you run the command "python3 decoder.py". The trained model is saved under directory "models/". You may need to modify the directories if it is necessary. Once your training is done, you can run the mini Chat-GPT by command "python3 prompt.py".

You are recommended to read the original paper [1] that proposed transformer model carefully. Before you fill the missing codes, you are recommended to get familiar with **pytorch tensor** first.

3 Requirements

- You are required to implement it in Python;

This material is distributed as a part of course "Multimedia technology". All rights are reserved by the author.

- No AI assistance is ALLOWED;
- Please send the source file ONLY to me. Indicate your student number and student name in both the source code comments and the email;
- The submission due is 2026-Nov.-05.
- Email: stonescx@gmail.com

Along with the source code, you are required to submit a report in English (PDF format). Basically, there should be three sections. In the first section, the Chat-GPT, namely the main idea about the autoregressive transformer decoder should be reviewed. In the second section, please elaborate how you implement the Chat-GPT, explain the parameter settings you used and how you train the network. In the section followed, please present your observations when you run the mini Chat-GPT. What are the behaviors. Which part is normal, which part is abnormal and explain why.

References

[1] Ashish Vaswani and Noam Shazeer, Niki Parmar and Jakob Uszkoreit, et. al: Attentions is all you need, NIPS 2017, pp.5998–6008.

This material is distributed as a part of course "Multimedia technology". All rights are reserved by the author.