Homework 7
600.482/682 Deep Learning
Fall 2025

November 7, 2025

Due Wednesday Dec 3 11:59 pm EST

Instructions. Please submit 1) a single zip file containing your Jupyter Notebook and PDF of your Jupyter Notebook to "Homework 7 - Notebook" and 2) your written report (LaTeX generated PDF) to "Homework 7 - Report" on Gradescope (Entry Code: **VWJRB3**)

Important: You must program this homework using the PyTorch framework. We highly recommend using Google Colaboratory. If you don't have local GPU access, you should port the provided Python scripts to Colaboratory and enable GPU in your environment (Edit/Notebook Settings).

1. *TinyGPT.* Consider the provided Jupyter notebook `Homework7.ipynb`. You will implement a minimal version of a *Generative Pretrained Transformer (GPT)* to understand how Transformers model sequential data through self-attention, normalization, and next-token prediction. Please implement the following `forward` functions in the notebook marked as `# TODO`.

   i. *Self-Attention Head.* Compute the normalized similarity score between Query and Key vectors, attention weights (e.g. `softmax`), and apply dropout. Apply the attention weights to the Value vectors to produce the output.

   ii. *Multi-Head Attention.* Concatenate Self-Attention heads implemented in (i).

   iii. *Transformer Block.* Normalize embeddings and assemble the residual paths from the Multi-Head Attention block (ii) and the FeedForward MLP.

   iv. *TinyGPT Model.* Compute token and positional embeddings, logits (using the transformer blocks (iii)), and loss.

   Considering the above implementation, please answer the following questions:

   (a) Draw a computational graph showing the flow of information from input to output, clearly annotating the blocks described above.

   (b) Consider the following equation for the attention weights A from the query (Q) and value (V) vectors.

   $$A = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) \tag{1}$$

   What is the purpose of dividing by $\sqrt{d_k}$? What happens to the softmax function if we remove this division?

   (c) *Causal Masking.*

      i. Train until convergence and attach your loss curve. Generate a sample output (approx. 500 tokens) and include a screenshot of the results.

      ii. Now, include a **causal mask** before computing the attention weights so that for each position, only current and previous tokens contribute to its attention scores. Retrain until convergence, attach your loss curve, and generate a new sample including a screenshot of the results.

iii. Compare your results before and after including the causal mask. What do you observe in the loss curve and generated text? How does the causal mask affect model training and prediction quality?

(d) Why is it important to normalize the embeddings (e.g. using Layer Normalization) in the Transformer block? What would happen if we do not use this normalization?

2. *Optional - for bonus credit.* Consider the `tiny_shakespeare.txt` corpus used to train your *GPT* which is approximately 1M characters/200k words. In this exercise, we will explore fine-tuning a GPT model on a custom dataset. Please select your own article or essay with **5,000 - 15,000** characters and complete the following questions.

   (a) *Dataset Preprocessing.* Convert your selected text into a single `UTF-8 .txt` file, removing non-text artifacts and unrelated content. Describe (if any) the steps you used to clean the data.
   Describe your selected text. How many characters are there, and what is the tone of the text (e.g. casual/personal, academic, news, etc.)? Summarize the information content in 1-2 sentences.

   (b) *GPT2 Inference.* Use a sample input phrase from your selected text to prompt the GPT2 model. Include a screenshot of both the input phrase and the result below.

   (c) *Fine-tuning.* Using the GPT2 model, fine-tune with your selected text. Attach the loss curve and generated text sample with the same input phrase and the fine-tuned model.

   (d) *Interpret.* Compare your results before and after fine-tuning. How did the fine-tuned TinyGPT text style, vocabulary, or tone change to reflect your dataset?