CSCE 421: Machine Learning (Fall 2025)
Assignment #5

Due 11/25/2025, 11:59PM

1. You need to submit (1) a report in PDF and (2) your code files, both to Canvas.
2. Your PDF report should include (1) answers to the non-programming part, and (2) results and analysis of the programming part. For the programming part, your PDF report should at least include the results you obtained, for example the accuracy, training curves, parameters, etc. You should also analyze your results as needed.
3. Please name your PDF report "HW#_FirstName_LastName.pdf". Please put all code files into a compressed file named "HW#_FirstName_LastName.zip". Please submit two files (.pdf and .zip) to Canvas (i.e., do not include the PDF file into the ZIP file).
4. Only write your code between the following lines. Do not modify other parts.

   ### YOUR CODE HERE

   ### END YOUR CODE
5. LFD refers to the textbook "Learning from Data".
6. All students are highly encouraged to typeset their reports using Word or LaTeX. In case you decide to hand-write, please make sure your answers are clearly readable in scanned PDF.
7. Unlimited number of submissions are allowed and the latest one will be timed and graded. If you make a resubmission after the due date, it will be considered late.
8. Please read and follow submission instructions. No exception will be made to accommodate incorrectly submitted files/reports.
9. Please start your submission to Canvas at least 15-30 minutes before the deadline, as there might be latency. We do NOT accept E-mail submissions.

---

1. (15 points) This question refers to the textbook "Deep Learning : Foundations and Concepts". Consider the autoregressive language model given by (12.31) and repeated below as:

$$p(\mathbf{x}_1, \ldots, \mathbf{x}_N) = \prod_{n=1}^{N} p(\mathbf{x}_n \mid \mathbf{x}_1, \ldots, \mathbf{x}_{n-1}) \tag{1}$$

   and suppose that the terms $p(\mathbf{x}_n \mid \mathbf{x}_1, \ldots, \mathbf{x}_{n-1})$ on the right-hand side are represented by general probability tables. Show that the number of entries in these tables grows exponentially with the value of n.

2. (15 points) This question is about transformer language models. You are given a decoder transformer language model in which the conditional distributions $p(\mathbf{x}_n \mid \mathbf{x}_1, \ldots, \mathbf{x}_{n-1})$ are modeled using a transformer. Now you are asked to modify the model to become a tri-gram model in which $p(\mathbf{x}_n \mid \mathbf{x}_{n-1}, \mathbf{x}_{n-2})$ is modeled. Explain how will you modify the model. You may use an example to illustrate your answer.

3. (15 points) Explain why encoder language models are NOT able to generate sequences.

4. (55 points) (Coding Task) **GPT for text generation tasks:** In this assignment, you will implement a decoder-only Transformer model on the SCAN dataset using *PyTorch*. The goal of SCAN is to translate commands presented in simplified natural language into a sequence of actions. In this translation task, the generation model will take a command sentence as input and output the corresponding action sequence. For more details, please refer to the dataset available at `https://github.com/brendenlake/SCAN`. You can also refer to its original paper at `https://arxiv.org/pdf/1711.00350.pdf` for the introduction of the task. The starting code is provided in the "code" folder. All the architecture and training code is provided, but you need to implement the CSABlock (causal self-attention blcok) class in the "model.py" file and the generate_sample function in the "generate.py" file. Similarly, in this assignment, you must use a GPU. If you do not have access to GPU, please refer to the course Syllabus about applying for GPU access on HPRC. Note that it may take two or three days for the application process.

   Requirements are the same as our project including Python, NumPy, PyTorch, plus tqdm and datasets (from Hugging Face). Other packages for transformer implementations are not allowed.

   Please submit running report (briefly explain how you complete those functions, capture screen shots of your training/validation loss, test acc etc., anything required in the original assignment) for all coding tasks. And paste training and testing console record as an appendix in your report. You mustn't submit the pre-trained model and the dataset which could be potentially large. Code should be written in a clean and organized way. Comments in the code are required. Questions that only include code without a report and explanations will not be graded. Please make sure all required files are included in your submission by downloading your submission on Canvas and double check it.

   (a) (10 points) Run the starting code directly to download the SCAN dataset automatically. Read the code, understand the data processing, and answer the following questions: What is a tokenizer? How does a tokenizer process the raw data? What is the size of the vocabulary? **hint**: use `pip install datasets` to install the `datasets` package. The downloaded dataset is saved to `~/.cache/huggingface/datasets/`.

   (b) (5 points) What is the maximum length of the input sequence? How should we determine the maximum length of the input sequence? **hint**: check the input arguments in "main.py".

   (c) (10 points) Implement the CSABlock class in the "model.py" file. Which steps are involved in the self-attention mechanism? Which step is critical to make it "causal" in your code? Why do we need a mask in the forward function of the class "GPT"? Report your training process and results.

   (d) (10 points) Implement the generate_sample function in the "generate.py" file. What is the generation process? Please explain the process using a concrete example in the dataset.

   (e) (10 points) Re-train your GPT model using different number of layers, number of heads, and number of embeddings. Report your validation loss, time per epoch, and test results in a table. What is the impact of these hyperparameters on the model performance?

   (f) (10 points) There are other splits (instead of the "simple" one) of the SCAN dataset `https://github.com/brendenlake/SCAN`. You can use other splits by simply setting

the CLI argument "data_split" to the names of the splits. Please try to use another split (your choice). What is the split you choose? What is the type of evaluation that the split is designed for? What insights do you get from the comparison? You can refer to the results reported in the original paper `https://arxiv.org/pdf/1711.00350.pdf`, where each **Experiment** subsection corresponds to a different split in the dataset.