Objective:

In this assignment, we are tasked with building a miniature version of GPT-2 from scratch, experimenting with different architectures and hyperparameters. As part of the project, we used Andrej Karpathy's llm.c as the baseline. The trained model is then evaluated on HellaSwag, with the goal of surpassing the baseline score of 0.31 on the HellaSwag benchmark.

Methodology:

As part of this assignment, I experimented with the following architectures. Initially, I ran the models on Colab, where I spent $30 to perform initial tests, running up to 2,000 iterations per architecture to gauge early performance for specific hyperparameter settings. After identifying the model that performed comparatively better, I transferred it to the HPRC for further training. However, I encountered several issues and bottlenecks, which I will discuss under the challenges section. The architectures I tested are as follows:

1. SwiGLU + RMS Prop + GroupedQuery + RoPE
2. RoPE (No Positional Encoding) + Grouped Attention + GeLU
3. RoPE (No Positional Encoding) + GeLU
4. Baseline (Andrej Karpathy's llm.c with modified hyperparameters and context embedding size)

Standard Hyperparameters:

1. batch_size - 16
2. dtype – float16
3. weight_decay – 0.1
4. zero_stage – 1
5. learning_rate – 0.004
6. warmup_iters – 1500
7. learning_rate_decay_frac – 0.85
8. overfit_single_batch – 0

Performance Comparison for minimum runs: (All tried on Collab with 1 A100 40GB GPU)

Click on the architecture and it redirects to .py of my GITHUB page where the GPT variant of these architectures can be found

Architecture	#Iterations Cutoff Point	Performance (Val)	Performance
SwiGLU+RMS+GroupedQuery+RoPE	5000	7.79 and hella accuracy: 0.2531	Worst Performance
RoPE+GELU	5000	3.39 and hella accuracy 0.261	Comparable to Baseline

Rope+Grouped Query	5000	3.57 and hella accuracy 0.2741	Slightly better than baseline
Modified Baseline – Increased Context Window and Embedding Size	5000	3.47 and hella accuracy 0.2801	Even better than baseline
Rope+Grouped Query—Increased Context Window and Embedding Size	5000	3.32 and hella accuracy 0.2912	Best so far…unfortunately could not fully complete it.

The sample output display logs for the above modles are shown here.

We increased the context window to **2048** and the embedding size to **1024** for both the baseline and the RoPE + Grouped Query architectures. As a result, we observed a significant increase in HellaSwag accuracy. However, due to resource and time constraints, and since this idea was explored at the last minute, we were only able to run **9000 iterations**. At **10000 iterations**, we achieved an accuracy of **0.3142** for the Grouped Query + RoPE architecture and **0.3002** for the baseline model. I am excited to see how the results unfold after reaching **19,650 iterations**. Unfortunately, we lost the track of log on Collab as the system crashed due to timeout and able to run only upto 10000 iterations.

What is submitted for final and why?

Due to the above cases, I submitted modified baseline. Honestly, would like to submit Grouped Query and RoPE with increased context window architecture but unfortunately I ran into issues like HF format conversion and could not complete the iteration on time, so both the log and checkpoint did not get generated as expected.Also in modified baseline, I ran on collab as I have very few SE on grace and faster and I plan to reserve for my course project. I paid $53 to google collab to avail premium service of one **A100 40GB GPU** scheduled for 24 hours and set the iteration to 12000. Again, unfortunately the job got terminated at **11163** steps. But the model check point was saved at 10,000step and I am submitting this as .bin and its associated log. Interestingly, we received an accuracy of **0.3068** at **10000** iterations vs **0.3002** at **19650** of normal baseline.

Challenges Faced:

1. We could not complete the entire 19650 iterations for any model. Unfortunately for Grouped Attention I fell in short of 1 second and job got terminated.
2. Paying to google colab is expensive. I used this technique to save my SE for my project. But Collab GPU with same configuration as the Grace and Faster was several times slower and resulted in constant run time crash resulting in the loss of data.
3. Even though the grouped query with extended window size performed better than all, due to complexities in HF format conversion and also this idea was created two days

back, I ran out of time and enough resources and could not fetch the logs as I ran on Collab and again due to run time crash lost all log information.

Total Budget Spent to Collab:

$80.64

Final Results:

1. Modified Baseline with increased context window and embedding size showed accuracy around **0.3062 at 10000** runs vs baseline's **0.3002 at 19560**

2. Grouped Query+RoPE with increased context window and embedding size showed accuracy around **0.3142 at 10000** runs vs baseline's **0.3002 at 19560**

3. Grouped Query+RoPE normal showed accuracy of around **0.3089 at 19560** runs vs **0.3002 at 19560**

Final Notes:

For plots and further information about the architecture and its details, all information are properly maintained in my github (Please look into readme for plots and refer to plot the results.ipynb)

Overall it was a great learning experience and I really loved the way of operating with different architectures!