AI504: Programming for AI (Fall 2024)

Second Project: Training a Language Model

Start: Nov 26th, 11:59:00 AM

Due : Nov 27th, 11:59:00 AM

Project Instructions

In this project, your task is to build a custom language model by training on ELI5 data. Evaluation will be performed based on perplexity. We are providing a minimum baseline perplexity. We used Claude (free version) to write a simple 6 layer Decoder-only Transformer model using sinusoidal positional encoding. Each decoder layer consists of (1) multi-head masked attention and (2) feed forward network, each followed by dropout (0.1), residual connection, and normalization. The model uses hyperparameters following the "Attention is all you Need" paper, where multi-head masked attention consists of 8 heads, with $d_k = d_v = 64$ (per head), $d_{model} = 512$. The feed-forward network consists of 2 layers with ReLU activation in between, and $d_{ff} = 2048$ (inner-layer), $d_{output} = 512$. When tested in a Colab environment with **Python version: 3.10.12, PyTorch version: 2.5.1+cu121, Datasets version: 3.1.0, Transformers version: 4.46.2** it achieved a perplexity of **432** on the test set (with minimal training). Your goal is to design a model with test set perplexity lower than this baseline.

Project Requirements

- **Dataset**
  - The ELI5 dataset is an English long-form question-answering dataset. In this project, we use only the "answer" portion of the data. Preprocessing code for this data is provided in **base.py**. Running this code generates a dataset with 17,655 training samples, 5,344 validation samples, and 75 test samples. Each sample has a fixed sequence length of 200. You must use the provided code for preprocessing without any modifications; only the training, validation, and test samples produced by this code may be used. For tokenizing, the GPT-2 tokenizer specified in the code must be used.
  - **Important Note: Do not use the test set during training.** If you are found to have used the test set for training, it will result in an **automatic fail**.
- **Model Specifications**
  - You are free to design any language model architecture, including pre-built models or custom configurations.

- **No restrictions** on model architecture, and training process. However, your submitted code must be **runnable without errors within the Colab Free version (e.g., no out of memory issues)**.
- You must use **datasets version: 3.1.0**, and **Transformers version: 4.46.2**.
- Your model must achieve perplexity **lower than 432** on the test set.

Submission Instructions

You must submit three files. These files must strictly follow the format below:

1. **Logits File (studentID.npy)**
   - **Description**: Logits refer to the raw, unnormalized output scores from your model for the **test set** before applying the softmax function. These scores represent the model's confidence in each token within the tokenizer's vocabulary, where larger values indicate higher confidence.
   - **Format:** A NumPy array with the shape **(75, 200, 50257)**, with 50257 corresponding to the tokenizer vocabulary size.
   - **File Name:** Must be named as **studentID.npy** (e.g., 20241234.npy).
   - **Important:**
     i. **Incorrect file names or formats will result in an automatic fail.**
     ii. The logits must be directly usable (without modification) as input logits for the provided test_lm.py evaluation function to calculate perplexity in the test_lm.py evaluation function. Make sure that your .npy file is compatible with this function. **Shuffling the test set is strictly prohibited, as it will invalidate your results**.
2. **Script File (studentID.py)**
   - **Functionality:** Your Python script should preprocess the data, train the model and save the logits in a **single execution.**
   - **File Name**: The script must be named studentID.py (e.g., 20241234.py).
   - When we run python studentID.py, it should:
     i. **Fix the seed to 0 for reproducibility** by calling the set_seed(seed=0) function (provided in **base.py)** once at the top of your code (before executing any other code).

```
def set_seed(seed=0):
    random.seed(seed)
    np.random.seed(seed)
    torch.manual_seed(seed)
    torch.cuda.manual_seed(seed)
    torch.backends.cudnn.deterministic = True
    torch.backends.cudnn.benchmark = False
```

ii. Use the preprocessing code in **base.py without modification** for preprocessing.
iii. Train your language model. As mentioned above, you must use **Datasets version: 3.1.0, Transformers version: 4.46.2.**
iv. Save the logits as studentID.npy **in the same directory.**

- **Important**
  i. The .py file must save the .npy file directly without manual intervention.
  ii. To ensure reproducibility, use the set_seed function to fix the seed 0 in the Colab environment **(Python version: 3.10.12, PyTorch version: 2.5.1+cu121).**

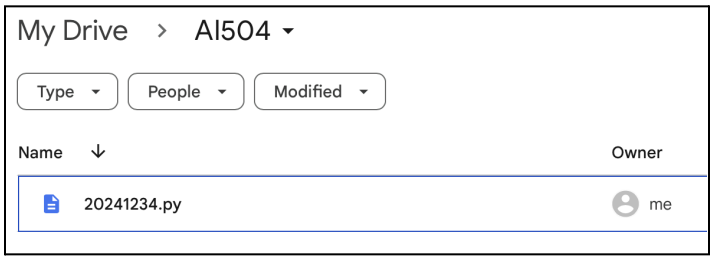
3. **Colab File (studentID.ipynb)**
   - We are providing a **studentID.ipynb** file as shown below. You must add code **within "Code Cell 3" (below the comments)** to install the necessary packages with the specified versions to ensure reproducibility of your submitted script file (studentID.py). Use the following format to install packages: "**!pip install [package_name]==[version]**". Your submitted Colab file may modify the path ('/content/drive/MyDrive/AI504') or name of the .py file ('20241234.py'), but do not modify any other code. Ensure that when running your .py file in the specified Colab environment **(Python version: 3.10.12, PyTorch version: 2.5.1+cu121, seed 0)** with **datasets version 3.1.0, transformers version 4.46.2,** and your specified list of packages (list of "!pip install xx"), the generated logits file (.npy) matches your submitted logits file.
   - You **must not modify** datasets version (**3.1.0**) and transformers version (**4.46.2**).
   - Modify the "studentID.ipynb" file name to **your own studentID**.

```
[ ]  # Code Cell 1
     from google.colab import drive
     drive.mount('/content/drive/')

[ ]  # Code Cell 2
     import os
     print(os.path.isfile('/content/drive/MyDrive/AI504/20241234.py'))

[ ]  # Code Cell 3
     !pip install datasets==3.1.0
     !pip install transformers==4.46.2
     ########## [TODO] Install packages used in your .py file with specified version ############
     # You must only add code below this line, within this cell.

[ ]  # Code Cell 4
     import datasets
     import transformers
     print(datasets.__version__)
     print(transformers.__version__)

[ ]  # Code Cell 5
     !python /content/drive/MyDrive/AI504/20241234.py
```

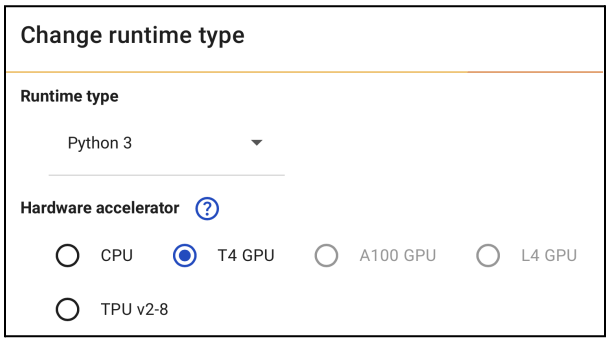
4. **Submission Guidelines**
    - Upload the studentID.py, studentID.npy, studentID.ipynb files to KLMS.
    - **Do not create folders or compress the files**.
    - Make sure that your script file (.py) is **runnable without errors within the Colab Free version (e.g., no out of memory issues)**.
    - Failure to adhere to **any** of the above guidelines, including incorrect file names or formats, will result in an **automatic fail**.

Additional Guides

1. **Running your Script File on Colab**
    - Upload your studentID.py script file on Google Drive.



    - Open the **studentID.ipynb** file, and set the runtime type to Python 3 & CPU or GPU.



- In the **studentID.ipynb** file, (1) mount Google Drive, (2) modify the path and name of your studentID.py file, (3) install packages, and (4) run your studentID.py file.

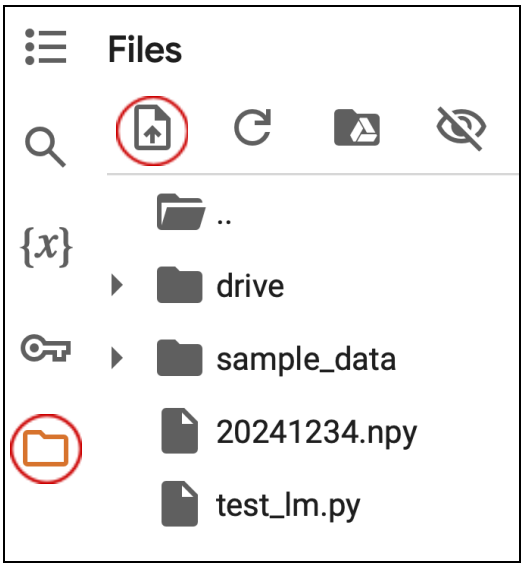
```
[ ]  # Code Cell 1
     from google.colab import drive
     drive.mount('/content/drive/')

[ ]  # Code Cell 2
     import os
     print(os.path.isfile('/content/drive/MyDrive/AI504/20241234.py'))

[ ]  # Code Cell 3
     !pip install datasets==3.1.0
     !pip install transformers==4.46.2
     ########## [TODO] Install packages used in your .py file with specified version ############
     # You must only add code below this line, within this cell.

[ ]  # Code Cell 4
     import datasets
     import transformers
     print(datasets.__version__)
     print(transformers.__version__)

[ ]  # Code Cell 5
     !python /content/drive/MyDrive/AI504/20241234.py
```

2. **Evaluation**

We have provided a **test_lm.py** file to assist with your project. This file includes the **"evaluate"** function, which loads your logits file and calculates perplexity using the test set.

Follow these steps to use the **test_lm.py** file:

1. **Upload the test_lm.py File**:
   - In Colab, click on the folder icon on the left-hand side of the interface.
   - Click the "Upload" button and select the test_lm.py file from your local machine.
   - This will upload the file to your Colab environment.



2. **Install Datasets version: 3.1.0, Transformers version: 4.46.2** using pip
3. **Import the test_lm.py File**:
    - After uploading, you can import the file by using the following command: // from test import evaluate
4. **Running Evaluation**:
    - Once you've generated your logits (studentID.npy), you can call the evaluate function to check your model's perplexity // evaluate("studentID.npy")

```
[ ]  !pip install datasets==3.1.0
     !pip install transformers==4.46.2
```

```
[ ]  import torch
     from test_lm import evaluate

     print(torch.__version__)
     evaluate("20241234.npy")
```

This process ensures you can seamlessly evaluate your model's perplexity in Colab using the provided test_lm.py file.

Evaluation Criteria

- Your model will be evaluated using the test_lm.py file. Perplexity will be calculated and **rounded from one decimal place** (e.g., 431.5 → fail, 431.1 → pass).
- Grading Criteria:
    - Pass: Perplexity lower than 432.
    - Fail: Perplexity of 432 or higher, or if:
        - No submission is made.
        - The submission is late.
        - The test set is used during training.
        - **The submission does not follow the submission guidelines.**

Important Notes

- **Perplexity Metric:** Only the perplexity calculated using the **test_lm.py** evaluation function will be considered for grading.
- **Project Support:** All project-related questions will be answered **only through KLMS** during the designated period: **November 26nd, 11:59:00 AM to 11:59:00 PM**.
    - **Emails will not be accepted.**
    - When posting questions on KLMS, please post them **publicly** so that others can also see them.
    - **Before posting your questions**, please make sure to (1) **carefully read the project guidelines**, (2) **read answers to KLMS questions posted from other students**, so that there are no duplicate questions.
- **Submission Guidelines:** Failure to follow any part of the submission guidelines will result in an automatic Fail (grade F). No exceptions will be made due to the class size (approximately 500 students). Ensure your submission strictly follows the required format.