Deep Learning (Homework 2)

Due date : 2025/5/9 23:55:00

1 Text Classification (30%)

To start this problem, some preliminary steps need to be conducted first:

1. **Join the in-class competition on Kaggle. (Link)**
2. Download the data and check for the description.
3. **Change your team name to your student ID.**

You are provided with News_train.json, News_test.json, and submission.csv, containing around 130,000 training examples, 1,000 test examples, and the submission format. Your task is to build a Transformer to classify these text into their corresponding categories and submit the classification result to the Kaggle competition.

1.1 Data Preprocessing (5%)

In this homework, you **cannot** directly use high-level API to help you process the text data. You are required to preprocess the texts and convert them into sequences of integers. Here are some packages you may find useful: FastText, TorchText, NLTK, spaCy or Gensim. If you are not familiar with tokenizer, check `https://www.analyticsvidhya.com/blog/2019/07/how-get-started-nlp-6-unique-ways-perform-tokenization/`.

explain how your processes the text data. (5%) (e.g. Which tokenizer do you choose? Why? What is your setting?)

1.2 Transformer (25%)

Build the Transformer (using high-level API is forbidden) to solve this task and answer the following questions. (Hint: You might want to read this first (Pytorch Transformer Doc))

1. Discuss the model structure or hyperparameter setting in your design. (4%) (e.g. hyperparameters of transformer: d_model, nhead, d_hid , nlayers, dropout, etc. Why do you choose these settings?)
2. The grading of this part will be based on the Kaggle leaderboard ranking. Submissions that surpass the baseline will receive a minimum score, while both the public and private leaderboard scores will be graded linearly according to their rankings.
   - Consolation prize (5%): Join Kaggle competition and pass the Baseline Line
   - Public ranking score (8%)
   - Private ranking score (8%)

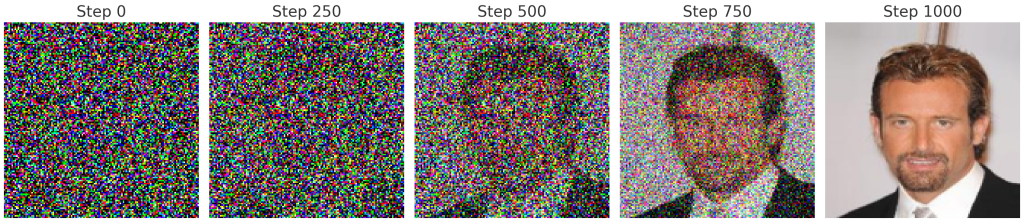
Note:

- To ensure fairness, please implement your model on the Kaggle platform so that everyone uses the same hardware resources.
- Since we will be running your code on Kaggle, please make sure it is fully reproducible for consistent results.

2 Image Generation (35%)

In this exercise, you will train a **diffusion model (DDPM)** for uncontrolled image generation by the provided **CelebFaces Attributes (CelebA) Dataset**. The dataset contains 10,000 images, each with a resolution of 64×64 pixels, selected from the CelebA dataset. Here are the link 1 and link 2 to the detailed introduction to diffusion model.

1. Please train a diffusion model for image generation using **CelebA dataset**. Your model should include a **forward process** which gradually adds noise to the images, and a **reverse process** that denoises the images to generate samples from the target distribution. Please **plot 5 images** to illustrate the denoising steps. For guidance on the diffusion model, you may refer to the provided diffusion.py file. (10%)



NOTE: Directly load the pre-traind model from pytorch is not allowed.

2. Using the model trained in Question 1, generate 1000 images through the diffusion process. Each image should be saved in JPG format with the filename corresponding to its index number, following this naming convention:

   - 1.jpg, 2.jpg, ..., 1000.jpg

   Once all 1000 images have been generated and saved, compress them into a submission.zip file. Please ensure the zip file contains only the images directly (i.e., no folders or subdirectories). All JPG files should be located at the root level of the zip archive.

   The competition will evaluate your submission using the **Fréchet Inception Distance (FID) metric**. A lower FID score indicates better image quality and higher similarity to the reference distribution. You need to minimize this score with your generated results.You can use the evaluate.py to calculate the FID score.

   The score will be divided into the **baseline score** and the **ranking score**.

   - Baseline : FID score lower than 100 (10%)
   - Ranking (10%)

3. Explain your model design, including any modifications or improvements . Discuss your findings and discussions based on the experimental results. For example, you may comment on how different design choices (e.g., number of timesteps, loss function, architecture, data augmentation) impacted generation quality or training stability. (5%)

3 Image captioning (35%)

In this homework assignment, using the image-caption dataset, you will fine-tune a pre-trained multimodal large language model, Llama 3.2-Vision 11B version. The goal is to enable the model to generate descriptive and coherent captions for a given set of images.

Files Included:

- train_data.parquet
  1. image
  2. caption
- valid_data.parquet
  1. image
  2. caption
- test_data.parquet
  1. idx: The unique identifier for the testing record
  2. image
- sample_submission.json
  1. idx: The unique identifier from test data
  2. output: Where you will write the caption obtained after executing the instructions based on the corresponding test image
- BLEU.py
  - The code that the TA will use to calculate your BLEU score

The generated output will be scored in terms of BLEU score. The BLEU score is reported as a percentage, with higher values indicating higher similarity between the generated text and the reference text.

- Baseline: BLEU score higher than 1% (20%)
- Discussion: Explain your fine-tuning strategy, including any modifications or improvements. Discuss your findings and analyze the experimental results. (15%)

Here are some resources that you can know more about

- unsloth
  - It is recommended to use **Colab** or **Linux devices**
- Meta Llama3

4 Rule

- Homework has three parts:
  - **HW2-1 is for Text classification:**
    Please submit one zip file named **hw2_1_<StudentID>.zip** contains
    - hw2_1_<StudentID>.ipynb
    - submission.csv
  - **HW2-2 is for Image Generation:**
    Please submit one zip file named **hw2_2_<StudentID>.zip** contains
    - hw2_2_<StudentID>.ipynb
    - submission.zip
  - **HW 2-3 is for Image captioning:**
    Please submit one zip file named **hw2_3_<StudentID>.zip** contains
    - hw2_3_<StudentID>.ipynb
    - submission.json
- Implementation will be graded by
  - Completeness
  - Algorithm correctness
  - Discussion and analysis
- Only Python implementation is acceptable.
- Kindly suggest using GPU to run the code! You can refer to the Colab tutorial in E3.
- DO NOT PLAGIARIZE. (We will check program similarity score.)