DATA 255 Lab 2

Part 1: How to Explore the GAN Latent Space When Generating Faces (25 pts)

1. Implement SR-GAN on your own using PyTorch.
   (Ref: https://arxiv.org/abs/1609.04802)

2. Use the ImageNet dataset to run a few epochs to train the model. You may use the script provided here to generate training data: Dataset. However, there is some issue with this code, hence the drive link for the generated images is shared below. Please use this link to access the dataset for training SRGAN. (https://drive.google.com/drive/folders/13txSH8LU64amnvY0hkxMxMI6F6ssbErG?usp=sharing) .

   Note: you should only use this repo for dataset code and reference. Model and results (20 pts) + report (5 pts)

Part 2 NLP (25 pts)

The Part 2 is an in-class competition.
Original Competition: https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge
Using the file: jigsaw-toxic-comment-classification-challenge.zip

The .zip file contains the Jigsaw Toxic Comment Classification dataset as provided in the original Kaggle competition. Data is organized as zip files containing .csv files. The training data is organized by ID, text, and label.

Content Warning: The dataset contains text that may be considered profane, vulgar, or offensive.

Step 1. Load the dataset

Step 2. Preprocess the data as you see fit

Step 3. Utilize a model implementing a Natural Language Processing strategy

Step 4. Train your model.

Step 5. Display the results of your model on the Test dataset by showing the predicted labels against their true labels

Submission Details: part2.ipynb file (Write your comments as markdown) Model and results (10 pts) + class ranking (10 pts) + report (5 pts)

Part 3 PDF RAG LLM with Langchain (30 points)

1. Create a Retrieval-Augmented Generation (RAG) LLM to consume PDF documents and allow users to prompt questions based on pdf documents to upload to RAG. To support the creation of this project, your team will leverage any LLM model, such as ChatGPT, Llama2, Mistral pre-trained model and pdf data collected from various sources.
2. Use the pdf data set in the canvas.
3. Implement the RAG model with Langchain (llama index) and FAISS (or ChormaDB, etc) (15 points). Test thoroughly, build quality metrics for RAG, fine-tune the RAG, and improve (document) the metric results in your report.
4. **This will be a class competition**. You will be graded based on the BLEU score. (10 points)
5. Report (5 pts)

Deliverables:

As files submit the following to Canvas

1. Python notebook containing the work done to complete part1, part2, part3
2. Report for report1, report2, report3

Report for each part

Write a report elaborating on your experience with each part. Cover the following topics:

- Discuss the model architectures used/experimented with, describing each element, loss/object function, different experiments, and your understanding.
- Provide an overview of the NN utilized. Report on the design process and any issues or challenges you faced during training and/or data selection and what worked and what did not work. Any parameter tunings.

Other Notes:

- Feel free to utilize whatever open source content is available to you, however, be sure to reference the original authors.