Assignment # 2

Generative AI **Deadline: October 21, 2024**

Instructor: Dr. Akhtar Jamil

Instructions

Note: There is no extension in the deadline. Please submit on or before the deadline. There is a **grace time of 2 hours** after the submission deadline expires. You must verify that your submissions are correct. Any submission received after this slack time will be considered late and NO marks will be awarded. There should be no comments provided on each line of code. You must provide detailed comments about functions that you create; that is enough to understand the function.

Deliverables:

1. Original Source file, including all code.

2. **Technical Report.** It must be written in Overleaf. Use IEEE Conference Paper Format for composing the report.

IEEE Conference Paper Format

Your report should include:

1. Introduction: A brief overview of the assignment.
2. Methodology: Description of the dataset, preprocessing steps, and model architecture.
3. Results: Training and validation loss and accuracy, examples of completed sentences.
4. Discussion: Analysis of sentence coherence, how the model's predictions improve over time, and any challenges encountered.
5. Conclusion: A summary of your findings.
6. Prompts
7. References

ZIP File Submission

Put all your files inside one folder and name it according to the following convention:

RollNo_Name_Ass1.ZIP

You must upload this ZIP file

Question # 1 VAE and Simple GAN Implementation

Objective

In this task, you will implement a Variational Autoencoder (VAE) and a simple GAN to generate fake signatures. You will train the VAE and a GAN network separately, generate new signatures by sampling from the latent space, and evaluate them using metrics like reconstruction loss.

The dataset might be smaller, so you can increase the dataset by augmenting the signature dataset using techniques like scaling, rotation, noise addition, etc. Ensure that the augmented dataset provides sufficient diversity for the VAE and GAN to learn meaningful latent representations.

Test the model on test dataset and see if the generated images match with those of original images.

Data Set

You can use the same data set that was shared as part of Assignment #1 consisting of signatures.

Question # 2 Custom GAN Implementation for CIFAR-10

Objective

You need to use CIFAR-10 dataset to train a custom GAN network. In this network, the generator will be the same as discussed in the class, however, instead of using discriminator to just distinguish between fake and real, you need to create a custom neural network that finds similarity between two images i.e. you pass a generated image and a real image together to the discriminator network. This will generate a similarity score. Now, the objective of GAN is still the same, discriminator will try to maximize the distance (dissimilarity) between the generated and real images while generator will want to minimize this score (as it wants to generate a similar image to that of a real one).

For this question you can limit your classes to only cats, and dogs.

Tips: For the discriminator network you need to design it in such a way that it takes two inputs and produces one output (similarity score). You can think of some special architectures such as Siamese Network.

Dataset: CIFAR-10
The CIFAR-10 dataset consists of 60,000 32x32 color images in 10 classes, with 6,000 images per class. The dataset is divided into 50,000 training images and 10,000 testing images. The classes are mutually exclusive and there is no overlap between them. For details, please visit the following website:
https://www.cs.toronto.edu/~kriz/cifar.html

Question # 3 Conditional GAN Implementation for CIFAR-10

Objective

Your task is to implement a Conditional GAN (cGAN) model using the Person Face Sketches dataset. The goal of this task is to use sketches as input conditions to generate realistic face images. The model should learn how to map a given sketch to a corresponding real face during training. The model should be able to take any input sketch (provided by the user) and generate a realistic face image based on that sketch.

Implementation Guidelines:

- You are required to refer to the original Conditional GAN paper for understanding the architecture and detailed implementation steps.
- Focus on ensuring that your model can condition on the sketch input to generate accurate and visually realistic face outputs.

Data Set

Person Face Sketches

Question #4: CycleGAN Implementation for Person Face Sketches

Objectives:

Implement the CycleGAN model based on the original paper, using the same Person Face Sketches dataset as provided in Question #3. The goal of this task is to convert a face image into a sketch and a sketch back into a real face (image-to-image translation).

At test time, the model should be able to:

- Take a sketch and generate a corresponding real face image.
- Take a real face image and generate a corresponding sketch.

The model will be trained end-to-end, and I highly recommend using Google Colab for this task. Make sure to save model weights after every epoch to avoid losing progress. If you need to restart the training for any reason, resume from the saved weights rather than starting from scratch.

If you encounter memory issues and cannot load the entire dataset at once, you can:

- Split the dataset into smaller batches.
- Reduce the number of training samples as a last resort.

Train your model for multiple epochs. Carefully monitor the training process and adjust hyperparameters (e.g., learning rate, batch size) as necessary to improve model performance.

User Interface:

You are also required to create a simple user interface. Deploy your trained model using Flask or any other API framework of your choice. The interface should be user-friendly and able to handle real-time image conversion requests. Using UI users should be able to:

- Upload a picture (or use live camera input).
- The model will convert the uploaded image into a sketch or generate a real face image from a sketch, depending on the input.

Data Set

Person Face Sketches