Deep Learning – SE4050

Generative Adversarial Network

Lab Activity – 09

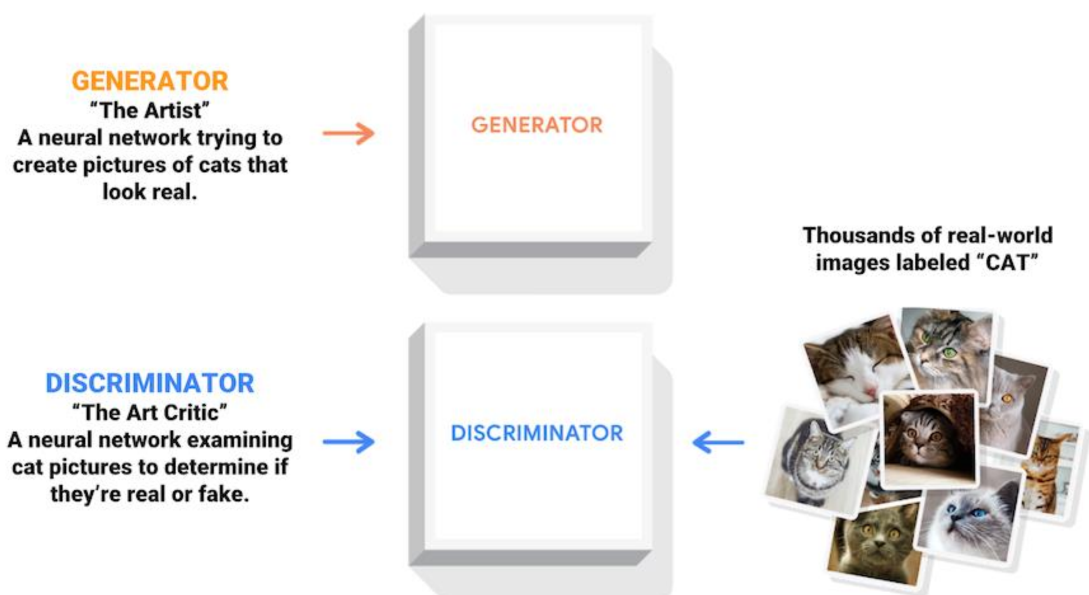
Generative Adversarial Network (GAN)

A Generative Adversarial Network (GAN) is a specific type of neural network designed for the purpose of generating realistic images, making it suitable for a wide range of applications. A GAN consists of two key components: a generator and a discriminator network.

The primary role of the generator network is to take as input a random noise vector and transform it into realistic, high-quality images. It achieves this by learning to generate images that closely resemble those found in the training data. The generator essentially creates synthetic images from random noise.

Discriminator network has a straightforward task: to determine whether a given input image is real or not. It accomplishes this by examining the visual characteristics of the image and making a binary classification decision.

The ultimate objective of training a GAN is to reach a state where the generator becomes highly proficient at producing images that are so convincing that the discriminator is left bewildered and unable to reliably distinguish between real images and those generated by the generator. This adversarial process leads to the continual improvement of the generator's image synthesis capabilities, resulting in the creation of realistic and convincing images.



Deep Learning – SE4050

Generative Adversarial Network

Lab Activity – 09

Part 1

1. Upload the GAN_MNIST_Dataset notebook into Google Colab.
2. Run the Notebook.
   Follow the instructions provided in the notebook to modify the latent space size, train the GAN for 10,000 epochs, change the optimizer, and experiment with different batch sizes.
3. Answer to the questions:

   i. Modify the latent space size: Change the noise vector size from 100 to 50 or 200. Observe how this change affects the quality and variety of generated images.

   ii. Train the GAN for 10,000 epochs: Save and visualize the generated images at every 1,000 epochs. Create a GIF or a slideshow to show how the quality of the generated images evolves over time.

   iii. Change the optimizer from Adam to RMSprop or SGD for both the generator and discriminator. Observe the impact on training performance and image quality.

   iv. Experiment with different batch sizes. What impact different batch sizes have on training?.

Part 2

1. Upload the CGAN_MNIST_Dataset notebook into Google Colab.
2. Run the Notebook.
3. Answer to the questions:

   i. Modify the CGAN to generate images for specific digits (e.g., generate only '7' or '9').

   ii. Implement label smoothing by replacing real labels of 1 with random values between 0.9 and 1 during training. Analyze how label smoothing affects the training process and the quality of generated images.

   iii. Create noise vectors corresponding to two different digits (e.g., '3' and '8'). Perform interpolation between the two noise vectors and visualize how the generated images morph from one digit to the other.

Deep Learning – SE4050

Generative Adversarial Network

Lab Activity – 09

Part 3

1. Upload the DCGAN_CIFAR_DataSet notebook into Google Colab.

2. Run the Notebook.

3. Answer to the questions:
   i. Modify the DCGAN to generate images for a specific class (e.g., cars, airplanes). Generate 10 examples of each class and analyse the quality and variety of the generated images.

   ii. Apply data augmentation (random cropping, flipping, etc.) to the real CIFAR-10 images during discriminator training. Evaluate if augmentation improves the discriminator's performance and results in better-generated images.

   iii. Implement the Inception Score to evaluate the quality of generated images. Train the DCGAN and compute the IS at various points (e.g., 2000, 5000, 10000 epochs). Compare the IS values with the visual quality of generated images.

   iv. Add more layers to the generator and discriminator to create deeper models. Train the deeper DCGAN and observe if this complexity improves the quality of generated images.

Submission

1. Create a report that includes:
   a. Include the visualized results (images, GIFs, etc).

   b. Discuss how changes in parameters (e.g., latent space size, optimizers, batch_sizes) affected the results.

   c. Discuss the quality and range of the generated images.

2. Update the answers in the relevant notebook. (GAN_MNIST_Dataset.ipynb, CGAN_MNIST_Dataset.ipynb, and DCGAN_CIFAR_Dataset.ipynb).

Create a ZIP file using your IT Number, and submit it.