DS5007 Deep Learning Lab 6 - GNN, GAN and Diffusion

Max Marks: 10
Deadline: 11/04/2025, 12:00 PM

Instructions

1. Provide well-commented, indented code with meaningful variable names.
2. Write the task description in separate text blocks before the corresponding code block.
3. Carefully follow the task requirements and use only the specified libraries or approaches.
4. Ensure all plots have appropriate axis labels, titles, and legends.
5. Submit a single Jupyter Notebook (.ipynb) file named YourName_YourRollNo_Assignment6.ipynb.

Objective:

This lab assignment is designed to give you hands-on experience with three important deep learning models:

1. **Graph Neural Networks (GNNs)**
2. **Generative Adversarial Networks (GANs)**
3. **Diffusion Models**

You will implement, train, and evaluate these models on different datasets.

Tasks & Marks Distribution

Task 1: Graph Neural Network (GNN) Implementation (3 Marks)

1. **Dataset:** Use the **Cora dataset**.
2. **Task:** Implement a **Graph Convolutional Network (GCN)** for node classification.
3. **Requirements:**
    - Load the dataset and preprocess it (normalize adjacency matrix, convert to sparse tensors).
    - Define a **2-layer GCN model**
    - Train the model and report **test accuracy**.
    - Visualize node embeddings using **T-SNE**.

Task 2: Generative Adversarial Network (GAN) Implementation (3 Marks)

1. **Dataset:** Use the **CIFAR-10** dataset.
2. **Task:** Implement a **Deep Convolutional GAN (DCGAN)** to generate images.
3. **Requirements:**
    - Load and preprocess the dataset (normalize to `[-1, 1]`).
    - Implement a **DCGAN** with:
        - Generator (`Conv2DTranspose` layers).
        - Discriminator (`Conv2D` layers).
    - Train the GAN and generate **sample images** after every few epochs.
    - Plot the **loss curves** for generator and discriminator.

Task 3: Diffusion Model Implementation (4 Marks)

1. **Dataset:** Use the **CIFAR-10** dataset.
2. **Task:** Implement a **Denoising Diffusion Probabilistic Model (DDPM)** for image generation.
3. **Requirements:**
    - Load and preprocess the dataset (rescale to `[-1, 1]`).
    - Implement:
        - **Forward diffusion process** (gradually add Gaussian noise).
        - **Reverse diffusion process** (U-Net denoising model).
        - Train the model and generate **samples at different noise levels**.
    - Compare generated images with real images.

Note:

- Include **plots, loss curves, and generated samples** for each task.
- For GNNs, **Spektral** is recommended (install via `pip install spektral`).