CS671: Deep Learning and Applications

(JAN-JUNE 2025)

Programming Assignment-2

Date: 25-Mar-2025

Submission Deadline: 05-Apr-2025, Sunday (till 23:59 hrs)

PART-I:

Representation Learning with Autoencoders and Denoising Autoencoders.

Dataset: CIFAR-10

- The dataset consists of **60,000 color images (32x32 pixels) from 10 classes**, with **50,000 training images** and **10,000 test images**.
- Each image is an RGB image with **3 channels**.

Model-1: Standard Autoencoder (AE)

You are required to build a **standard Autoencoder (AE)** to learn meaningful representations from the CIFAR-10 dataset.

1. The encoder should consist of **convolutional layers** for feature extraction.
2. The decoder should use **transpose convolutions (ConvTranspose2D)** to reconstruct images.
3. Train the model using **Mean Squared Error (MSE) loss**.

Task:

1. Train the AE on CIFAR-10 and evaluate reconstruction quality on test data(using SSIM,PSNR,MAE,MSE).
2. **Visualize latent space representations** using **t-SNE or PCA**.

Presentation of Results:

- Visualize **original vs. reconstructed images**.
- Plot **average error (y-axis) vs. epochs (x-axis)**.

- Compare **latent space structure** for clean vs. noisy input images.
- Discuss the **effectiveness of autoencoders in feature learning**.

Model-2: Denoising Autoencoder (DAE)

A **Denoising Autoencoder (DAE)** is trained to **reconstruct original images from corrupted versions**.

- Apply **Gaussian noise** (mean=0, variance=0.1) to **input images** before feeding them into the encoder.
- The **architecture remains the same** as AE, but the model learns to **remove noise** instead of just reconstructing inputs.

Task:

1. Train the DAE with **different levels of noise (e.g., Gaussian noise with $\sigma = 0.1, 0.3, 0.5$)**.
2. Compare the **reconstruction quality of DAE vs. AE** on **clean vs. noisy inputs**.
3. Evaluate **denoising performance** by testing on unseen noisy images.

Presentation of Results:

- **Visualize original, noisy, and reconstructed images**.
- **Plot the average error vs. epochs** for both AE and DAE.
- **Compare feature extraction quality between AE and DAE** using a **classifier trained on their latent representations**.
- **Discuss when AE vs. DAE is preferable** for feature learning.

PART-II

Variational Autoencoder and Latent Space Inference

Model-2: Variational Autoencoder (VAE)

Extend the autoencoder model into a **VAE**, incorporating a probabilistic latent space.

- Implement the **reparameterization trick** using **mean and variance outputs**.
- Train the VAE with **KL-divergence loss + reconstruction loss**.
- Use CIFAR-10 for training.

Task:

1. Train the VAE on CIFAR-10.
2. Perform **latent space interpolation** by generating transitions between two random images.
3. Conduct **latent space arithmetic** (e.g., “dog” - “cat” + “bird” = ?).
4. **Infer the trained VAE on unseen test images** and analyze generation quality(using SSIM,PSNR,MAE,MSE).

Presentation of Results:

- **Visualize latent space interpolations** and **latent space arithmetic**.
- Compare **VAE vs. AE reconstructions**.
- Plot the **KL-divergence loss and reconstruction loss over epochs**.
- Provide **qualitative observations on latent space structure**.

PART-III

Masked Autoencoder for Feature Learning

Model-3: Masked Autoencoder (MAE) with CNN

Implement a **Masked Convolutional Autoencoder (MCAE)**:

- **Mask random patches in feature maps** instead of raw image patches.
- Use **CNN layers** for encoding masked images.
- Train on CIFAR-10 and compare with VAE.

Task:

1. Train the **Masked CNN Autoencoder (MCAE)** with different masking ratios.
2. Compare its performance with **VAE** in terms of **feature extraction and reconstruction quality**.
3. Evaluate **classification accuracy** by using frozen embeddings from MCAE for **linear classification**.

Presentation of Results:

- **Compare reconstruction quality between MCAE, AE, and VAE**.
- **Decision region plots for classification tasks using learned embeddings**.
- **Heatmaps showing which parts of an image are most important for reconstruction**.
- **Observations on whether CNN-based masking retains global information like ViTs**.

Submission Requirements

1. **Code:** Submit a well-documented **Jupyter Notebook (`.ipynb`)**.
2. **Report (`.pdf`)** detailing:
    - **Problem definition & methodology.**
    - **Hyperparameter tuning & architecture choices.**
    - **Results, visualizations, and interpretations.**
    - **Key findings from latent space analysis.**

Guidelines for the Submission:

1. Upload the Assignment Report in PDF format with the following name : **<Group_number>_Assignment2_Report.pdf**

2. Upload the code files in a single zip file with the following name: **<Group_number>_Assignment2_Code.zip**

****