Task 1: Image Reconstruction and Noise Removal with Autoencoders

Description:

In this task, you will explore the power of autoencoders for data compression and noise removal. The objective is to train an autoencoder to reconstruct images and then use it to clean noisy images.

Steps:

1. Load a dataset of images (e.g., MNIST, CIFAR-10, or a custom dataset).
2. Preprocess the images by normalizing the pixel values between 0 and 1.
3. Implement an autoencoder with the following:
   - **Encoder**: A series of convolutional layers to compress the input image into a lower-dimensional representation.
   - **Decoder**: A series of transposed convolutional layers to reconstruct the original image from the compressed representation.
4. Train the autoencoder to minimize reconstruction loss (e.g., Mean Squared Error).
5. Test the autoencoder on unseen images and visualize the original vs. reconstructed images.
6. **Noise Removal**:
   - Add random noise to the test images (e.g., Gaussian noise).
   - Pass the noisy images through the autoencoder to observe how well it can

Task 2: Data Generation and Latent Space Exploration with Variational Autoencoders (VAEs)

Description:

The goal of this task is to explore the generative capabilities of Variational Autoencoders. You will train a VAE to learn a latent space representation of images and use it to generate new samples.

Steps:

1. Load a dataset of images (e.g., MNIST, CelebA, or a custom dataset).
2. Implement a Variational Autoencoder with the following:
   - **Encoder**: Compress the input image into a latent space representation and estimate the mean and variance of the latent distribution.
   - **Reparameterization Trick**: Sample from the latent distribution using the estimated mean and variance.
   - **Decoder**: Reconstruct the original image from the latent space.

3. Define the loss function as a combination of:
    - Reconstruction Loss (e.g., Mean Squared Error).
    - KL Divergence to ensure the latent space follows a normal distribution.
4. Train the VAE and visualize:
    - Original vs. reconstructed images.
    - Images generated by sampling from the latent space.
5. **Latent Space Analysis**:
    - Plot a 2D or 3D representation of the latent space for a subset of the data (if latent space has low dimensions).
    - Interpolate between two points in the latent space and visualize the generated images.

Task 3: Feature Extraction for Classification Using Autoencoders

Description:

In this task, you will use the encoder part of a trained autoencoder to extract meaningful features from images and use these features for classification.

Steps:

1. Train a basic autoencoder on a dataset (e.g., CIFAR-10 or Fashion MNIST).
2. Extract features from the latent space (output of the encoder).
3. Visualize the latent features using dimensionality reduction techniques like PCA or t-SNE.
4. Use these features as input to a classifier CNN.
5. Compare the classification accuracy using:
    - Raw image data.
    - Features extracted by the autoencoder.

Task 4: Anomaly Detection Using Autoencoders

Description:

You will train an autoencoder to detect anomalies in a dataset by measuring reconstruction error.

Steps:

1. Use a dataset with both normal and anomalous samples (e.g., KDD Cup dataset for network intrusions or an industrial sensor dataset).
2. Train the autoencoder only on the normal samples.
3. Compute reconstruction error for both normal and anomalous samples.
4. Set a threshold for the reconstruction error to classify samples as normal or anomalous.
5. Evaluate the anomaly detection performance using metrics such as precision, recall, and F1-score.

Task 5: Cross-Modal Data Generation Using VAEs

Description:

In this advanced task, you will build a cross-modal VAE to map data from one domain (e.g., images) to another (e.g., captions).

Steps:

1. Use a paired dataset (e.g., MS COCO for image-caption pairs).
2. Design a VAE with two modalities:
    - **Image Modality**: Use a CNN-based encoder to extract features from images.
    - **Text Modality**: Use an RNN or Transformer-based decoder to generate captions.
3. Train the model to learn a shared latent representation for both modalities.
4. Evaluate the generated captions for unseen images.