The Fourth Assignment : Auto Encoders

Reyhaneh Saffar 99222064

1. Data Processing

- Dataset Preparation

The data preparation stage involved the selection and preprocessing of two distinct datasets, CIFAR-10 and MNIST, to ensure they were compatible for further analysis. The preprocessing steps included:
The computation was configured to use a GPU if available, otherwise, it defaulted to the CPU. This ensures efficient processing, leveraging hardware acceleration when possible.
For the CIFAR-10 dataset, images were transformed into tensors and normalized. Normalization involved adjusting pixel values to a range centered around zero with a standard deviation of 0.5.
For the MNIST dataset, images were first converted to RGB format, resized to match the CIFAR-10 image dimensions, then transformed into tensors. They were normalized using mean and standard deviation values to align with the CIFAR-10 dataset's normalization parameters.
These steps ensured that both datasets were standardized in terms of image size and pixel value distribution, facilitating their combination in subsequent steps.

- Mean Image Computation

In this phase, the focus was on combining the two datasets by computing the mean of paired images from each dataset. The steps included:
A function was defined to compute the mean of corresponding images from the two datasets. For each pair of images, pixel values were averaged, resulting in a new image that encapsulated features from both original images.
The function was applied to the training sets of both datasets, resulting in a new set of mean images for the training data.
The same process was repeated for the test sets, yielding a set of mean images for the test data.
By averaging pixel values, this process created a new dataset that combined features from both CIFAR-10 and MNIST images, potentially enhancing the richness of the training and testing data.

- Test & Train Dataset

Finally, the mean images were structured into training and testing datasets suitable for model training and evaluation. The steps involved:
The stacks of mean images were reshaped to fit the expected input format for the model. Each mean image was processed to ensure it had the correct dimensions and format for the analysis framework.
The reshaped mean images were assigned to training and testing datasets, respectively. This structured organization ensured that the model could be trained on a dataset that included diverse features from both original datasets, and evaluated on a similarly structured test set.

2. Training Model

- Autoencoder Model Construction

The construction phase involved developing an autoencoder model, a type of neural network designed for unsupervised learning tasks such as data compression and noise reduction. The autoencoder consists of two main components: the encoder and the decoder.
The encoder compresses the input data into a latent space representation. This is achieved through a series of convolutional layers with non-linear activation functions, reducing the spatial dimensions of the input and learning a compact representation of the data.
The decoder reconstructs the original data from the latent representation. This is done using transposed convolutional layers that gradually restore the spatial dimensions to the original input size. Non-linear activation functions are also applied in the decoding process to ensure the reconstructed output closely matches the original input.
Additionally, a loss function was defined to measure the difference between the input and the reconstructed output, and an optimizer was set up to minimize this loss by updating the model parameters during training.

- Training the Autoencoder

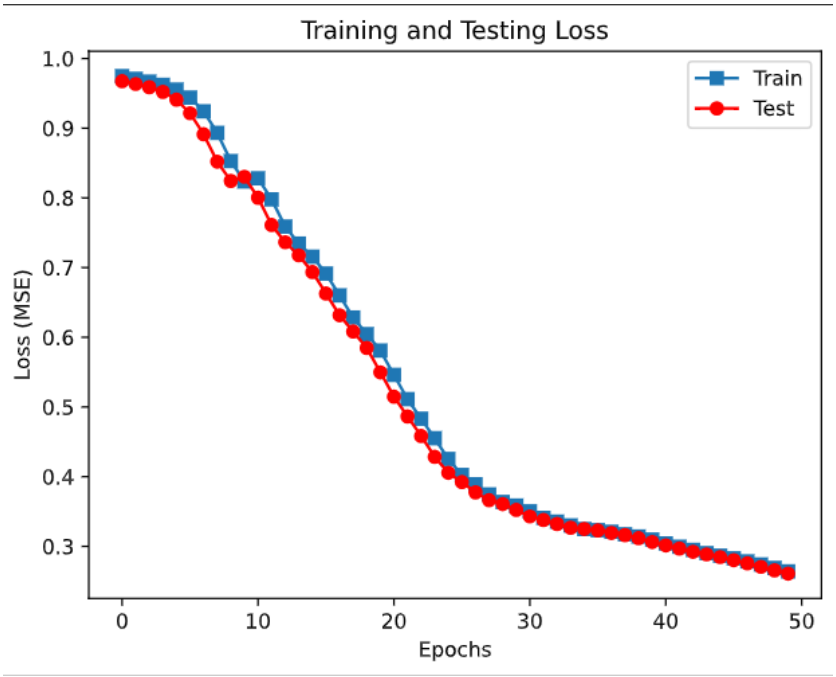
The training process involved several key steps to ensure the model learned effectively from the data:

1. Data Loading: The training and testing datasets were loaded into data loaders, which handled batching and shuffling of the data. This ensured that the model received data in manageable sizes and in a randomized order, crucial for effective training.

2. Training Loop: The model was trained over 50 epochs. In each epoch: The model was set to training mode. For each batch of training data, the model performed a forward pass to reconstruct the input. The reconstruction loss was computed by comparing the reconstructed output with the original input. A backward pass was performed to compute gradients, and the optimizer updated the model parameters to minimize the loss.

3. Evaluation Loop: After each training epoch, the model was evaluated on the test data: The model was set to evaluation mode to disable gradient computation, saving computational resources. The test data was passed through the model to obtain reconstructed outputs.The reconstruction loss was computed for the test data and recorded.

Throughout the training process, both the training and testing losses showed a steady decrease, indicating effective learning and generalization by the model. For instance, the training loss decreased from 0.9747 in the first epoch to 0.2641 in the final epoch, while the testing loss reduced from 0.9674 to 0.2609. These improvements demonstrate the model's increasing ability to accurately compress and reconstruct the input data over time.



The provided plot illustrates the training and testing losses over the 50 epochs. Both the training and testing losses showed a steady decrease, indicating effective learning and generalization by the model.

Training Loss: Decreased from 0.9747 to 0.2641.

Testing Loss: Reduced from 0.9674 to 0.2609.

3. Evaluation

The evaluation phase focused on assessing the performance of the trained autoencoder model by comparing the reconstructed images with the original images using two widely recognized metrics: Structural Similarity Index (SSIM) and Peak Signal-to-Noise Ratio (PSNR).

SSIM is used to measure the similarity between two images. It considers changes in structural information, luminance, and contrast. The SSIM value ranges from -1 to 1, where 1 indicates perfect similarity.

PSNR measures the peak error between the original and reconstructed images. It is expressed in decibels (dB). Higher PSNR values indicate better image quality with less distortion.

- Model Evaluation

The model was set to evaluation mode to ensure that it performed inference without computing gradients, which is more efficient and appropriate for evaluation. For each image in the test dataset, the original image was passed through the autoencoder to obtain the reconstructed image. SSIM and PSNR values were computed for each pair of original and reconstructed images.

- The evaluation yielded the following results:

The average SSIM score was 0.2322. This relatively low value indicates that while the reconstructed images retain some structural similarity to the original images, there are noticeable differences, suggesting that the model has room for improvement in capturing finer structural details.
The average PSNR score was 15.6239 dB. A PSNR value in this range suggests that the reconstructed images have a moderate level of distortion when compared to the original images. Higher values would indicate less distortion and better reconstruction quality.
These metrics provide a quantitative measure of the model's performance. The relatively low SSIM and moderate PSNR values suggest that while the autoencoder can reconstruct images to some extent, it may not capture all the fine details and structural similarities perfectly.

The provided visualization shows a side-by-side comparison of an original mean image and its reconstructed counterpart from the test dataset:
The image on the left is a mean image from the test dataset, representing the combined features of the original data.
The image on the right is the output from the autoencoder after processing the original mean image.

The visual comparison highlights the differences between the original and reconstructed images. The reconstructed image retains some features of the original, but the overall quality and detail are significantly lower. This visual discrepancy is consistent with the quantitative metrics, where the SSIM and PSNR values indicate that the model struggles to accurately reconstruct high-quality images.