Report

Sajjad Zangiabadi

Summary:

- This project focuses on the use of autoencoders, an essential tool in the field of deep learning aimed at reconstructing images that come from two distinct sets named MNIST and CIFAR-10. The aim here is to build an autoencoder model that excels in accepting the average of an image from both MNIST and CIFAR-10 feeding this average to the model and then skillfully crafting reconstructions of images from both the MNIST and CIFAR-10 datasets.

Data Description

- For this project two familiar image collections are in use: MNIST alongside CIFAR-10. With 70,000 grayscale pictures of hand-penned numbers each measuring 28x28 pixels the MNIST collection stands out; on the other side CIFAR-10 brings together 60,000 color pictures (RGB) spread across 10 varied categories each with a dimension of 32x32 pixels.
- Before the autoencoder can use the images from the two datasets they have to go through a few key steps of preparation. This is to make sure they match up when it comes to their size and channel setup
- MNIST Image Transformation:
  - **Resize**: The size of MNIST images gets changed from the first size of 28 by 28 pixels to a bigger size of 64 by 64 pixels.
  - **Grayscale to RGB**: MNIST images are originally in grayscale and have just one channel, they get transformed by adding two more channels to them so they can fit the RGB setup used by CIFAR-10 images.
  - **Normalization**: The images are normalized using the mean and standard deviation unique to the MNIST dataset.

- CIFAR-10 Image Transformation:
  - Resize: 32 * 32 pixels to 64 * 64
  - Normalization
- Mixed Dataset Construction:
  - To manage both MNIST and CIFAR-10 images together, a new dataset class called MixedDataset was designed. This class ensures that the autoencoder receives the averaged image, and the loss function receives both the averaged image and the original CIFAR-10 and MNIST images.

Model Architecture:

- The autoencoder model developed for this project consists of an encoder and two decoders, implemented using PyTorch. The architecture is set up to efficiently process and extract features from the input image, producing two distinct output images.
- Encoder:
  - Convolutional Layers
  - ReLU Activation Function
  - Batch Normalization
  - MaxPooling
  - Fully Connected Layers

- Decoders
    - Within the model's structure there are two distinct decoders. The job of the first is to generate images from the MNIST dataset while the second takes on the role of creating images that belong to the CIFAR-10 dataset.
    - Up Sampling Layers
    - Convolutional Layers
    - ReLU Activation Function
    - Residual Block (just second decoder has it)

Loss Function:

- The performance of the autoencoder is evaluated using a custom loss function, `ImageLoss`, which is specifically designed to handle the reconstruction of both MNIST and CIFAR-10 images. The `ImageLoss` function incorporates the Mean Squared Error (MSE) loss, a common choice for image reconstruction tasks due to its effectiveness in measuring the differences between pixel values of the reconstructed and original images.

Result: