Deep Learning Mini-Project

Roi Teichman, Elad Sznaj

Spring 2024

**Wet Part:**

Section 1.3.1:

1. **Overview of our AutoDecoder Implementation:**

**Architecture**:

* **Latent Space Dimension**: We used a latent space of size 64 ().  
  The latent dimension determines how much information the model compresses the image into. A smaller latent space compresses more, while a larger one retains more information. We chose 64 as a reasonable compromise between compression and reconstruction quality.
* **Fully Connected Layer**: The first layer in the decoder is a fully connected layer that reshapes the latent vector into a 128-channel feature map of size 7x7. This serves as a bridge between the latent space and the convolutional layers.
* **CNN Decoder**: The decoder uses **transposed convolutional layers** to progressively upsample the 7x7 feature maps into a full-size 28x28 image:
  + **Layer 1**: ConvTranspose2d(128, 128, kernel\_size=3, stride=2, padding=1, output\_padding=1) upsamples from 7x7 to 14x14.
  + **Layer 2**: ConvTranspose2d(128, 64, kernel\_size=3, stride=2, padding=1, output\_padding=1) upsamples from 14x14 to 28x28.
  + **Layer 3**: Conv2d(64, 32, kernel\_size=3, stride=1, padding=1) reduces channels to 32, keeping the size at 28x28.
  + **Layer 4**: Conv2d(32, 1, kernel\_size=3, stride=1, padding=1) outputs a single-channel (grayscale) image of size 28x28.
  + **Activation Functions**: Each layer is followed by a **ReLU** activation to introduce non-linearity, except the final layer, which uses a **Sigmoid** to constrain pixel values between 0 and 1.
* **Dropout**: We use **dropout** with a probability of 0.5 after the first two transposed convolution layers to prevent overfitting and improve generalization.

**Training Parameters**:

* **Optimizer**: We used **Adam** as the optimizer with a learning rate of 0.0005. Adam adapts the learning rate for each parameter, making it well-suited for deep learning tasks.
* **Learning Rate Scheduling**: We included a **learning rate scheduler** (StepLR) to reduce the learning rate by half (gamma=0.5) every 5 epochs. This helps the model converge more smoothly as training progresses.
* **Loss Function**: We used **Mean Squared Error (MSE)** to measure the reconstruction error between the original and reconstructed images.
* **Weight Decay**: We applied a small amount of **weight decay** (1e-6) to regularize the model and reduce overfitting.

1. **Evaluation:**

* **Training Loss**: During the last epoch, our training loss was approximately 4271.79, indicating the model's performance on the training set.
* **Test Loss**: Your test set evaluation loss was 2.8731. This shows how well the model generalizes to unseen data. The relatively low test loss suggests that the model is able to generalize to unseen test images.