MLP Autoencoder

ELEC 475

Lab 1

September 24th, 2024

Ryan Zietlow 20347719

Simon John 20348233

Model Details

In this lab a fully connected four-layer Multi – Layer Perceptron (MLP) encoder is used. The model consists of an input layer, encoder and decoder that works on flattened 28 by 28 grayscale pictures from the MNIST dataset. The input layer takes a flattened image from the dataset and normalizes each pixel value of the image between 0 and 1 to help the model in learning better and faster by making the data more consistent.

The encoder consists of two fully connected layers. The first fully connected layer has an input size of 784 and an output size of 392, meaning it shrinks the dimensionality of the image to 392 units. An activation function called the ReLU (Rectified Linear Unit) is applied, it introduces nonlinearity and helps the model learn difficult features. The second fully connected layer of the encoder has an input size of 392 and an output size of 8 (bottleneck layer). This bottleneck layer acts as a compressed representation of the input data and reduces the dimension of the image to 8 units.

The decoder mirrors the encoder and has two layers as well. The third fully connected layer has an input size of 8 and an output size of 392. The decoder expands the bottleneck back to 392 units, and the ReLU function is applied. The fourth fully connected layer has an input size of 392 and an output size of 784. This layer fully restores the dimensionality of the image back to 784 and applies a sigmoid function to map output values between 0 and 1 to make sure that the reconstructed images stay valid.

Training Details

The optimization method used is Adam, it tweaks the model’s parameters by adjusting the learning rate based on the gradients. It has a learning rate value of 0.001 and a weight decay value of 0.00001. The learning rate scheduling used is the ReduceLROnPlateau. It’s used to adjust the learning rate during training based on the performance of the model, specifically the training loss. If the training loss stops improving, then the learning rate is reduced. The loss function used is the Mean Squared Error (MSE) Loss. It measures the averaged squared difference between the original image and the reconstructed image. The training procedure consisted of 50 epochs, which is the number of complete passes through the dataset, and a batch size of 2048, which is the number of samples processed before the model's parameters are updated. During each epoch four steps occur. Firstly, a forward pass where the input images from the MNIST dataset are put through the encoder to get the reconstructed image. Next, after the forward pass, a loss computation occurs where the loss between the initial and reconstructed image is measured using the MSE Loss function. Third, a backward pass occurs where the gradients of the loss are computed. Finally, the model is adjusted based off the gradient that was found in the backward pass. The model is saved after every epoch and the training loss is plotted to help visualize.

Results

Based on the results received from the images and the loss curve it seems that the model worked well with no surprises along the way. As seen below in the loss curve there is a steep decline during the first two epochs, indicating the model was learning very rapidly and effectively at the start. This is followed by a drop off and then a flattened curve at a loss of around 0.01 after 10 epochs which signifies that the model was stabilizing then and was making minimal improvements there on after. These results display that an effective reconstruction of the initial image from the MNIST dataset was achieved.