Project I: Deep autoencoder based on dense neural networks

Deep Learning

Authors

Juan Muñoz Villalón
100502334@alumnos.uc3m.es

Mario Golbano Corzo
100504162@alumnos.uc3m.es

Elena Almagro Azor
100504241@alumnos.uc3m.es

Universidad Carlos III de Madrid

November 7, 2024

Index

In this lab, we implemented a deep autoencoder based on dense neural networks for the MNIST and FMNIST datasets.

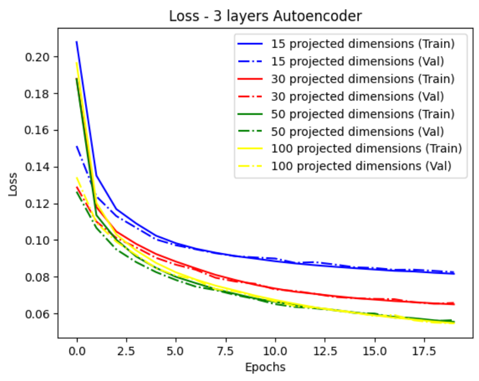
1 Autoencoder

The autoencoder is a fully connected neural network designed for dimensionality reduction and reconstruction, consisting of an encoder with N layers leading to a bottleneck and a mirrored decoder. It uses ReLU activation. The extended class adds training functionality with an MSE loss, Lasso regularization, Adam optimizer, and methods for PSNR calculation and visualizing reconstructions. This supports efficient training and evaluation over multiple epochs, aiming to learn low-dimensional representations while mitigating overfitting. We tested architectures with 3 and 5 layers, evaluating the impact of different projected di- mensions (15, 30, 50, 100).

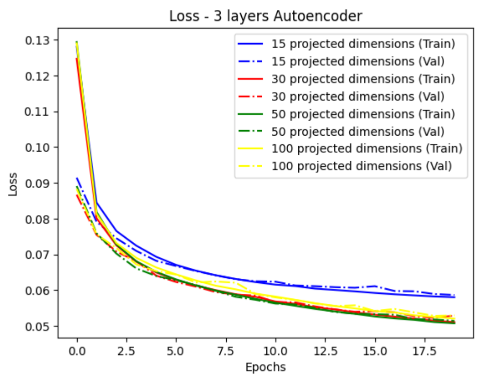
1.1 Autoencoders' Results

1.1.1 Loss over epochs

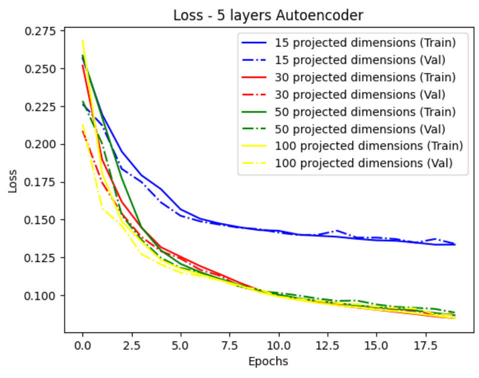
In Figure 1, we observe similar behavior for the 3-layer and 5-layer autoencoders across datasets: the loss decreases as training progresses, indicating learning. Smaller projected dimensions (e.g., 15) show higher losses, suggesting that the autoencoders struggle to retain sufficient information. Larger projections lead to lower losses, but improvements beyond 50 dimensions are minimal. In the FMNIST dataset, the model with 100 projected dimensions even underperforms compared to the 30 or 50 dimension models, indicating potential overfitting, with smaller projections (30 or 50) being more suitable.



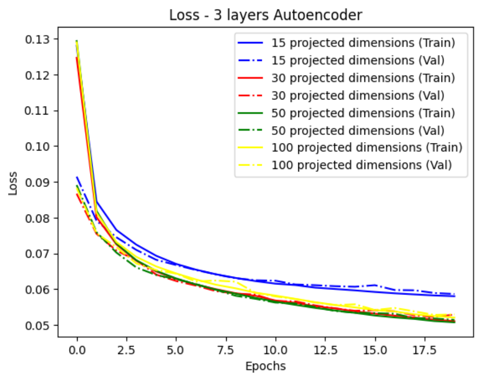
(a) 3 Layer autoencoder - MNIST



(b) 3 Layer autoencoder - FMNIST



(c) 5 Layer autoencoder - MNIST

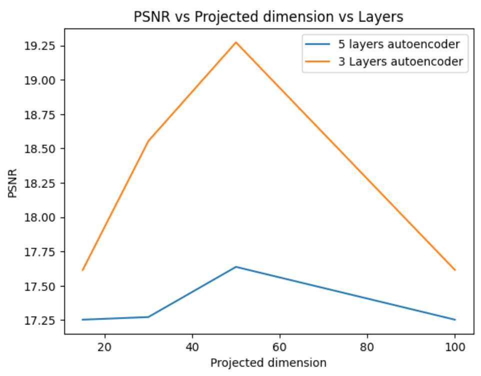


(d) 5 Layer autoencoder - FMNIST

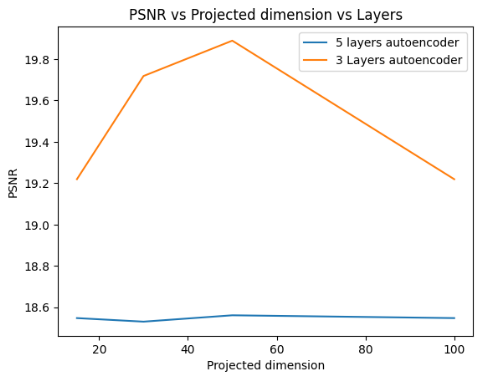
Figure 1: Training and validation loss for the different autoencoders and datasets.

1.1.2 Peak Signal-to-Noise Ratio

Figure 2 supports the previous observations: as the projected dimension increases in 3 and 5-layer autoencoders, PSNR improves, peaking at 50 dimensions for both MNIST and FMNIST datasets. Beyond this point, PSNR decreases, suggesting that higher dimensions may capture redundant details, negatively impacting performance. Interestingly, the 5-layer autoencoder shows lower PSNR compared to the 3-layer one, likely due to increased complexity. For tasks like image reconstruction with MNIST and FMNIST, the simpler 3-layer model with 50 dimensions appears more effective, emphasizing that added depth may not always improve performance.



(a) MNIST



(b) FMNIST

Figure 2: PSNR vs projected dimension for both datasets.

1.1.3 Regularization

Throughout the project, we applied regularization techniques beyond Lasso, such as dropout layers and monitoring validation loss. Figure 1 shows no increase in validation loss over 20 epochs, making early stopping unnecessary. Dropout was tested on the initial model (3-layer autoencoder with MNIST), but resulted in a significant decrease in PSNR (Figure 3), leading to its omission. The PSNR drop was likely due to the temporary reduction in the model's reconstruction ability caused by deactivating nodes during training.

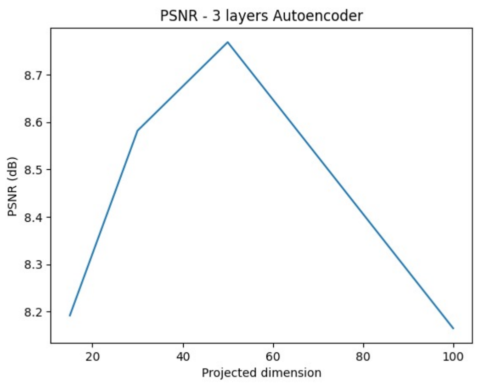
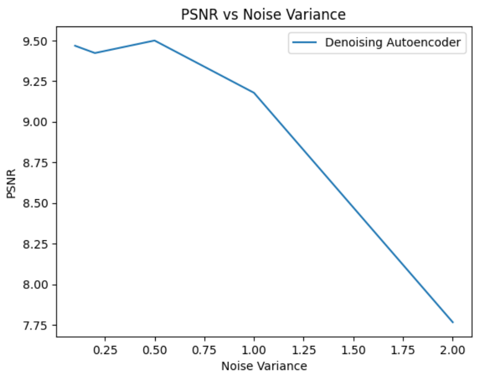


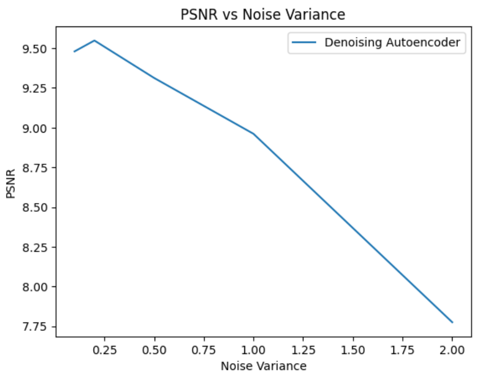
Figure 3: PSNR for 3-layers autoencoder (MNIST) with dropout

2 Denoising Autoencoder

The Denoising Autoencoder class extends "autoencoder_3_extended" to reconstruct clean data from noisy inputs using a denoising approach. The architecture reuses the one that returned the best psnr on the previous study, which is the 3 layer and projected dimension 50 model. During training, Gaussian noise is added to the inputs, and the loss is calculated using Mean Squared Error (MSE) with Lasso regularization, just as before. Figure 4 shows PSNR of the models trained with noise of different variances (0.1, 0.2, 0.5, 1, 2) evaluated at images with a Gaussian noise of variance 1. We can see that the model trained with low noises has better performance, as the images with high noise variance do not give much information. Moreover, we can see that the models trained with 0.5 (MNIST) or 0.2 (FMNIST) variance noise have the best PSNRs. This means that these models are best for noisy images, since 0.5 or 0.2 variance noise is not noisy enough for not adding information, but is noisy enough for making the model robuster.



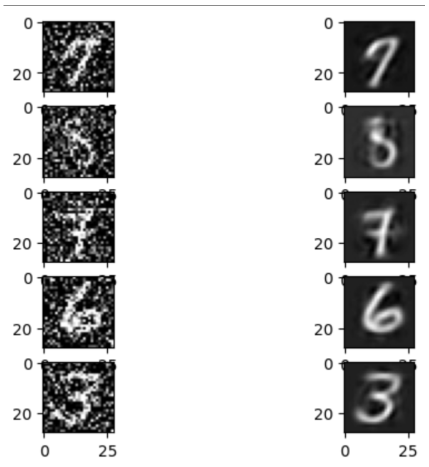
(a) MNIST



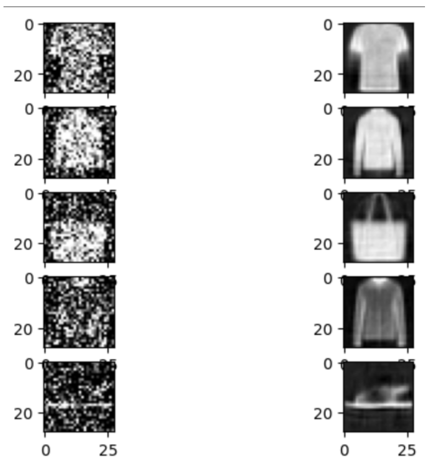
(b) FMNIST

Figure 4: PSNR for denoising autoencoders

The final step was to evaluate our models using two different datasets. We selected the model with the higher PSNR to reconstruct images from the "testloader," after adding Gaussian noise with a variance of 1 to these images. This allowed us to assess the model's ability to handle noisy inputs and produce accurate reconstructions. In the following figure 5, we can observe the positive results of the reconstructed images compared to the noisy ones, recovering the essential features of the input images. We can observe that the reconstructed images are pretty similar to the original ones, which demonstrates the model's robustness in handling noisy data.



(a) MNIST



(b) FMNIST

Figure 5: Reconstructed images from both datasets for a Gaussian noise with var = 1