

Neural Networks

Carlos III University of Madrid

March 11, 2024



Practice 1 – Autoencoders

Pablo Gradolph Oliva - 100458456

Raquel Parajuá Delgado - 100454359

Table of Contents

1. Introduction.....	3
2. MNIST Database	3
a. Validation of the 3-layer network.	3
b. Validation of the 5-layer network.	4
3. FMNIST Database	5
4. Denoising Autoencoder	5
5. Conclusion	6

1. Introduction.

In this project we have implemented a deep autoencoder based on dense neural networks over both MNIST and FMNIST databases. An autoencoder is used to learn efficient coding of unlabeled data, which takes an input image and encodes it to get a low dimensional embedding of the image to be then decoded in order to reconstruct the original image as similar as possible.

To do this, we've first validated a 3-layer network over the MNIST database, changing the dimensions (15,30,50,100); then the same but with a 5-layer network, changing the dimensions as well. To all this, Lasso regularization and Ridge regularization were applied simultaneously (Elastic Net) in order to prevent the overfitting that occurs as a result of the model learning the specific patterns from the training data set so well it fails to generalize to new unseen data, helping improving this generalization issue by adding a penalty to the loss function, forcing the model to learn smoother or smaller weights. By using Elastic Net, we combine both properties of Lasso regularization such as the ability of characteristics selection with the ability of Ridge regularization to penalize weights without eliminating them completely. We then repeated the same exact procedure but over the FMNIST database.

After carrying out this, we selected the 3-layer network autoencoder (since it was faster to run) both the lowest dimension and highest one to compare them and determine possible differences, and applied noise over the input image, in order to implement a denoising autoencoder and obtain the original input image but noise free. The same was done for the FMNIST database.

Presented below are the different results obtained for each step and procedure. In this report no images from the code will be attached given the amount there would be, but we'll be commenting the obtained results and observations which complement the Jupyter notebook attached with all the corresponding images.

If for some reason, you cannot see the images in the attached Jupyter Notebook, please access this GitHub repository to see them:

<https://github.com/PabloGradolph/Neural-Networks/blob/main/Project1/Project1.ipynb>

2. MNIST Database.

a. Validation of the 3-layer network.

For a 3-layer network we decided to work with 10 epochs each time because we thought it would be enough to see the desired results and test the efficacy of the model. Each 4 epochs we are going to have a visual representation of the results, which is going to complement the numerical results obtained as the loss values.

For all the dimensions, we can see how the more epochs have passed the lower the loss value is and the better the output image is. This is expected and indicates that the model is improving its capacity to represent data and generalize to new ones.

Comparing the results between the different projected dimensions we can also observe some divergences between them. Having a lower or a higher dimension or latent space means a difference in the amount of information the model captures from the input images and the loss or gain of compression of the information, and therefore a variation in the representation of the output images.

A smaller latent space restricts the amount of information that can be retained during the encoding process, and although this might be useful when forcing the autoencoder to learn the most essential features of the data, it may also lead to a significant loss of information, making it difficult to accurately reconstruct the input data. A larger latent space provides more capacity for the autoencoder to learn and store information about the input data which allows the model to capture a wider range of variations in the data, resulting in reconstructions that are more faithful to the original images.

This can be observed both in the numerical values of the loss and in the visual representation of the data. When we increase the dimension, the loss decreases, and the accuracy of the images is better.

Also, our hypothesis for when adding noise to the images and using a denoising autoencoder, is that the lower dimensions are going to work more accurately because the autoencoder learns the most essential information or features of the image and somehow “avoids” part of information as would be the noise in this case, taking only the most important features for the reconstruction of the image.

b. Validation of the 5-layer network.

For this network we've decided to rise the number of epochs to 13, which we thought was enough to see the desired results even though we are aware that even a higher number would've been better, but due to the time it takes to run we decided 13 would be a fair amount. This need to increase in epochs is due to the higher complexity of a 5-layer network than one of 3 because there are more parameters in the first one. This means it has more capacity to learn sharp and smooth details of the data but needs more time (epochs) to do it.

Also, with more layers, the model has a greater tendency to overfit, especially if the training data set is relatively small. Overfitting can lead to the model learning the "noise" of the training data rather than the true underlying features, which can result in higher loss when evaluating on unseen data. To avoid this, we've decreased *lambda_L1* and *weight_decay*, because otherwise the loss was too high. By doing this we're allowing the 5-layer network autoencoder to adjust better to the data and reducing the loss. This could be because of a combination of less regularization, a lower tendency to overfitting and a faster convergence of the model.

3. FMNIST Database.

As already mentioned, we have repeated all the above steps but for the FMNIST database and we have observed two key points:

The first is that the loss function gives us slightly higher values than for the previous database, and the second is that we do not see so clearly a decrease in the loss function as we increase the projected dimension, as was the case previously.

This may be mainly due to the differences between the databases. MNIST contains images of handwritten digits that are relatively simple and consistent in terms of shape and size. On the other hand, Fashion-MNIST (FMNIST) contains images of fashion items (clothes, shoes, bags, etc.), which are inherently more complex and varied. This additional complexity in FMNIST can make it more challenging for an autoencoder to learn to reconstruct the images with the same accuracy it achieves with MNIST, resulting in a higher loss for FMNIST.

In addition, the improvement in loss with increasing latent space dimensions may be more pronounced in MNIST due to its lower complexity compared to FMNIST. In a more complex dataset such as FMNIST, incremental improvements in latent space capacity may not be sufficient to capture all the additional complexity in the data, which could explain the lower variability in loss across different latent space dimensions.

The behaviour between the 3-layer and the 5-layer autoencoder for this database is very similar to the previous case, it has been necessary to adjust the regularization parameters as we have explained before, and it is necessary to increase the number of epochs for the 5-layer case to see a clear decrease of the loss function as we advance in epochs.

4. Denoising Autoencoder.

We've decided to apply noise to the 3-layer network to ease the procedure, and from it, the lowest and highest dimension in order to make a good comparison. This was done the same way over the MNIST and FMNIST database.

Here, to evaluate the efficacy of the model, added to the numerical value of the loss and the visualization of the images, we are going to have the PSNR as a metric to evaluate the quality of a restored image in comparison to the original one and quantify how deteriorated the image is due to the noise applied.

PSNR should increase with the number of epochs, this is because as the autoencoder trains, it learns to reconstruct the inputs more faithfully from the latent space. For a denoising autoencoder, this means learning how to effectively denoise input images and reconstruct a clean version of the image. An increasing PSNR indicates that the difference between the clean and reconstructed images is decreasing, which is a sign of better reconstruction. Therefore, loss should decrease as before and PSNR should rise.

We can observe in our results that this is indeed what happens in most cases, the PSNR increases with the number of epochs for both dimensions in each data set. However, we can observe some "ups and downs" of this value in some cases which can

just be because of some problem with overfitting and regularization techniques and could be solved by adjusting hyperparameters of the model.

Regarding our previous hypothesis that stated that lower dimensions were going to work better than the higher ones for a denoising autoencoder, we observed the following:

In the MNIST database, for the autoencoder with a dimension = 15 the loss is lower and the PSNR higher than those values of the autoencoder with dimension = 100. We believe this might be because of what was mentioned before, that with lower dimensions the autoencoder learns the most essential features of the image by the encoder compressing more the information and therefore taking only that fundamental information for reconstructing the images, avoiding any other-not important information (noise).

However, we don't observe this for the FMNIST database. This can be because, as we have already explained above, the data in this second database are more complex. This can mean that when the latent space is too small (e.g., dimension = 15), it may not sufficiently capture the complexity of the data, resulting in lower quality reconstructions compared to a larger latent space (e.g., dimension = 100). In MNIST, even with a low-dimensional latent space the autoencoder can effectively learn the essential features needed to perform accurate reconstructions.

This underscores the importance of tailoring the model architecture and hyperparameters to the specific characteristics of the dataset you are working with.

5. Conclusion.

This project allows us to understand how an autoencoder performs in image reconstruction and highlights the significance of properly adjusting model architecture and regularization hyperparameters to maximize efficacy across different datasets.

Overall, we've observed reasonable and predictable results throughout the project and a good execution of our model was achieved, and for those results that didn't exactly match predictions were reasoned and a fair-minded explanation was found.