

Neural Network and Deep Learning

Homework 4

Federico Agostini

1 Introduction

In this exercise unsupervised learning is explored using autoencoders on the MNIST dataset. Hyperparameters are optimized; in particular a sistematic serch over the size of the hidden layer is done. The trained model is then tested on both original images and corrupted versions.

2 Parameters tuning

Tuning process is done in two separate phases; in the first one the optimizer is tuned, and then the influence of the size of the hiddeen layer is explored. The same network architecture proposed during the laboratory is used.

The optimizer chosen is Adam; in order to tune the learning rate and weight decay (L2 penalty), a grid search with cross validation with 3 folds is implemeted with 20 training epoches. Learning rate values are $[5e-02, 1e-02, 5e-03, 1e-03, 5e-04, 1e-04]$, while the weight decay spans in the list $[5e-04, 1e-04, 5e-05, 1e-05, 5e-06, 1e-06]$. Only the training set is used in this phase. Best combinations of parameters with respect to the validation loss (MSE) are shown in Tab. 1.

Table 1: *Best values for the learning rate and weight decay of the Adam optimizer.*

Learning rate	Weight decay	Train Loss	Validation Loss
0.0100	0.000005	0.021793	0.022443
0.0100	0.000010	0.022037	0.022501
0.0050	0.000005	0.022173	0.022681
0.0050	0.000010	0.022197	0.022696
0.0050	0.000050	0.023330	0.023907

Fixed those hyperparameters, another search over the size of the hidden layer is done. In this case, the whole training set is used for training, while the loss is estimated with the test set (the mean value of the last 5 epoches is used). In order to avoid overfitting, a scheduler is also added to reduce the learning rate in case the test loss does not decrease for 5 consecutive iterations (`ReduceLROnPlateau` on Pytorch). The number of training epoches is now set to 50. Fig. 1a shows the test loss for different size of the hidden layer, while Tab. 2 has the numerical values of the final loss. The best model results to be the one with an embedding dimension of 20 and Fig. 1b represents its training and test loss as function of the epoch.

Figure 1

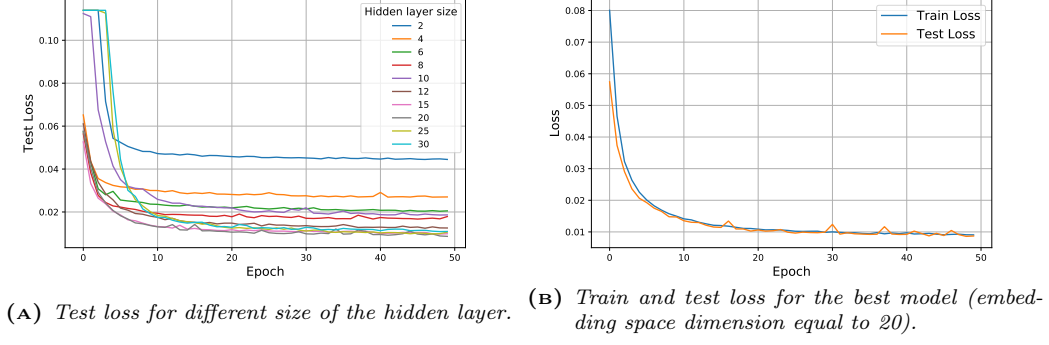


Table 2: Test loss as function of the size of the embedding layer.

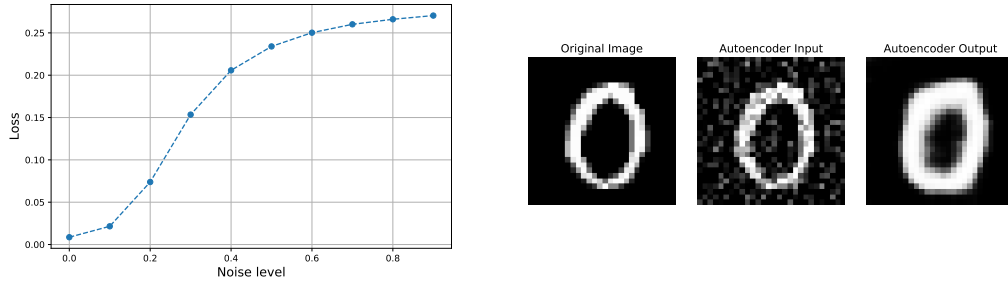
Size	Test Loss	Size	Test Loss
20	0.009258	8	0.017079
15	0.010073	10	0.018720
25	0.010108	6	0.020458
30	0.011145	4	0.027136
12	0.012720	2	0.044586

3 Reconstruction capability

The trained final model with 20 as encoded space dimension is then tested. On the MNIST test set the MSE is 0.0086. Images are the corrupted either adding normal noise or deleting a portion of the image itself.

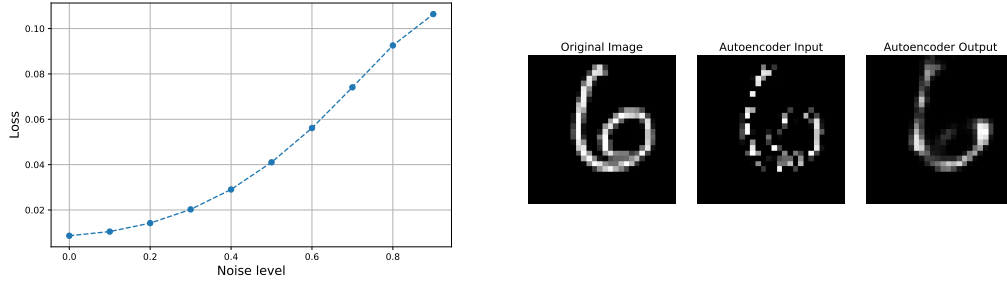
Noise Noise applied is proportional to $\mathcal{N}(0, 1)$, with a prefactor $\alpha \in [0, 1]$. Fig. 2 represents the MSE loss as function of the noise level, along with an example of the reconstructed image of the autoencoder.

Figure 2: MSE loss as function of the noise level and example of the autoencoder reconstruction.



Occlusion The occlusion level represents the fraction of pixel deleted. Fig. 3 shows the MSE loss as function of the occlusion level along with an example of a reconstructed image.

Figure 3: *MSE loss as function of the occlusion level and example of the autoencoder reconstruction.*



4 Encoding space

The representational space of the best model has dimension 20; in order to visualize it in 2 dimensions, t-SNE is used. Fig. 4 shows how different digits are grouped in the encoding space (2000 samples are plotted). It can be noticed that different digits clearly form separated clusters, so nearby points in the encoding space produce similar images.

Figure 4: *Encoding space visualization using t-SNE to reduce the dimensionality from 20 to 2.*

