NEURAL NETWORKS AND DEEP LEARNING
Homework 4

Laura Iacovissi

August 19, 2020

1 Introduction

In the following report it is described the design process of the `trained_model.py` script as it has been submitted. The program performs a reconstruction task on the `MNIST` dataset, using an Autoencoder with encoding dimension equal to 32 (i.e. hidden layer size), implemented with `PyTorch`.

1. In the first section the research of the optimal set of hyperparameters is analysed;
2. in the second section a description of the Training and Testing performance is provided. The Autoencoder is applied on both the original `MNIST` dataset and a corrupted version;
3. in the third section the influence on of the encoded space dimension value on the latent representation of the `MNIST` digits.

2 Optimal hyperparameters research and Cross Validation

The architecture of the Autoencoder used is made of two symmetric networks, the encoder and the decoder. The encoder has three convolutional layers and two linear layers, the decoder is equal but reversed.

The optimal set of hyperparameters has been found performing a Grid Search and testing them using a 3-fold Cross Validation. The splitting of the training set into folds has been done in a randomized way, to avoid any possible bias in data.

Doing some preliminary studies on the behaviour of the loss function with some specific values for the hyperparametes, possibly significant domains have been extracted:

- **Starting Learning Rate:** it has been observed that a too high value for the starting learning rate (LR) do not allow the Gradient Descent algorithm to converge to a solution. The selected set for this parameter is $\{10^{-2}, 10^{-3}, 10^{-4}\}$;
- **Weight decay:** given the value for the LR, the weight decay (WD) must be lower: the considered set for WD is $\{10^{-3}, 10^{-4}, 10^{-5}\}$.

Another relevant parameter is the **encoding dimension** (ED): its effects have been explored outside the Grid Search, since it will be evaluated looking at both the reconstruction error and the latent representation of the images. The considered values are $2^2, 2^3, 2^4, 2^5$.

During the optimal parameter setting research, the number of epochs was fixed to 20 in order to have an estimate of the errors shape in a short time.

Other (fixed) parameters used were: batch size = 1000 (the minibatch GD is used), activation function = ReLU function, GD optimizer = Adam, loss function = MSE.

noise type	MSE
original	0.026960546
$\mathcal{N}$(0,0.1)	0.027442785
$\mathcal{N}$(0,0.3)	0.032414775
$\mathcal{N}$(0,0.5)	0.041436195
$\mathcal{N}$(0,1)	0.062495653
$\mathcal{N}$(0,2)	0.08463218
occlusion	0.05018206
uniform	0.09382705

(a) ED=4.

noise type	MSE
original	0.008543696
$\mathcal{N}$(0,0.1)	0.00962089
$\mathcal{N}$(0,0.3)	0.021008287
$\mathcal{N}$(0,0.5)	0.037717797
$\mathcal{N}$(0,1)	0.07703511
$\mathcal{N}$(0,2)	0.13436298
occlusion	0.032860734
uniform	0.18368684

(b) ED=32.

Table 1: MSEs for the reconstructions shown in Figures 8, 7.

The final architecture of the Autoencoder was chosen by analysing the loss functions for all the possible combinations of the hyperparameters on every fold. All the loss functions that showed an high value of the loss (more than 10^{-1}) or an high oscillation on at least one of the splits was discarded. The remaining sets were evaluated looking at the error values (their actual values and the difference between the testing and the training one).

The best set of hyperparameters found is $\text{LR} = 10^{-2}, \text{WD} = 10^{-5}$ for all the values of ED.

3 Training and Testing

Given the set of optimal parameters, the Autoencoder has been trained on the whole training set adding the Early Stopping condition with max epochs = 2000, a window of 100 epochs and a tolerance of ± 0.0001. The double sign of the tolerance has been used to avoid both overfitting and a long plateau in the loss function.

This re-train step gave us the final weights setting and an estimation of the error and accuracy reached.

3.1 Results on Original Images

During the training, the performance of the model is also tested on a test set of 10^4 digits: the results for each encoding dimension are shown in Figures 1, 2, 3, 4.

The MSE value is clearly decreasing as the ED increases, which was expected since the digit compression is able to store more information when the latent space has a larger dimension. Given this observation alone, the best model seems to be the one with ED=32.

3.2 Results on Corrupted Images

The performances of the Autoencoder with ED=4, 32 under random noise have been investigated. The following noises have been used:

- Gaussian: random noise sampled from $\mathcal{N}(0, \sigma)$ for $\sigma \in \{0.1, 0.3, 0.5, 1, 2\}$;
- Uniform: random noise sampled from $U[0, 1)$;
- Occlusion: a portion of size 14×14 of the image, randomly chosen, has been set to zero.

It can be observed that under low noise conditions the ED=32 model reaches lower values for the MSE and the reconstruction is closer to the original image w.r.t. the results obtained with ED=4 (see Tab. 1, Fig. 8, 7).

However, when the noise is more affecting, the simpler representation obtained with ED=4 works better: probably this is possible since a smaller hidden layer forces the network to focus only on the features that really characterise the digit, without focusing on the details. Hence, the smaller representation could be useful when this behaviour of the Autoencoder is preferred.

4 Latent space representation

A visual representation of the latent space reduced to two dimensions has been obtained by using a PCA (`scikit-learn`). In Figures 5 and 6 are shown the latent space projection for ED = 4, 32. Each number is associated to a specific color, so that it is possible to observe how much the Autoencoder's representations of the digits are able to discriminate among different numbers.

Ideally, a perfect network would give as output a latent space with completely separated clusters. Taking in mind that this kind of representation is not giving an exhaustive description of this separation, some considerations can still be done.

The latent space with ED=4 shows clusters highly separated, the ED=32 has more mixed groups: this observation is coherent with the behaviour of the Autoencoder under heavy random noise. The ED=32 representation of the digits seems to be less separated, so changing some pixels makes harder for the model to assign the right label to the image. On the contrary, the ED=4 simpler representation seems to be able to better generalize the numbers' main features.

A Appendix: Images

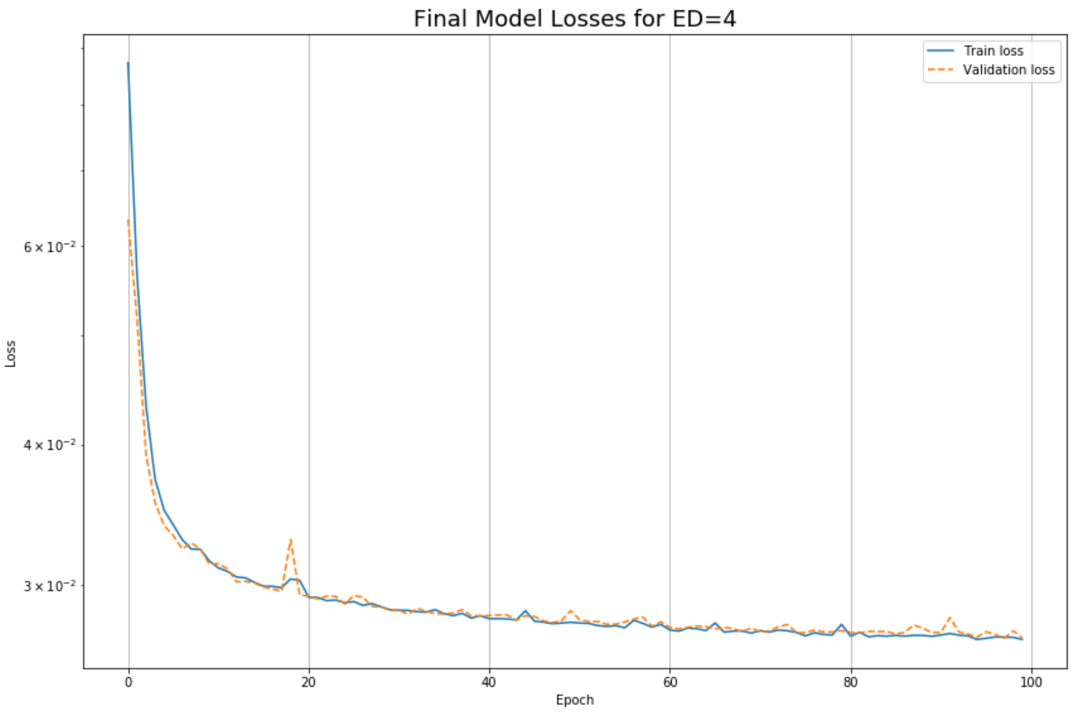


Figure 1: Loss per epoch of the optimized model.

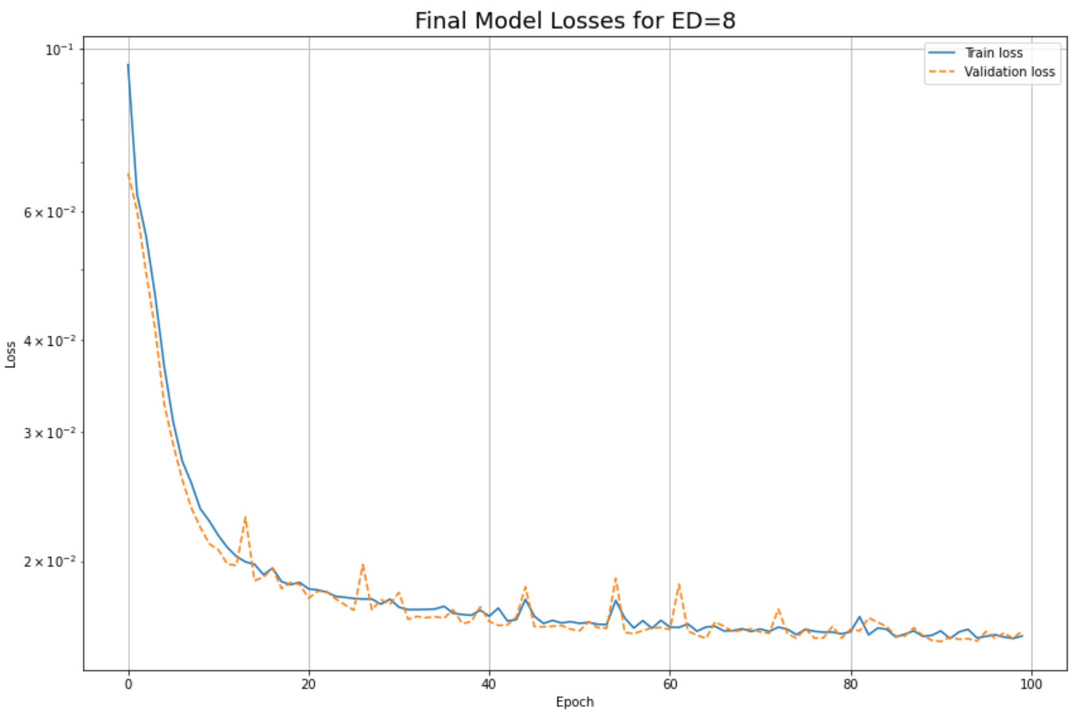


Figure 2: Loss per epoch of the optimized model.

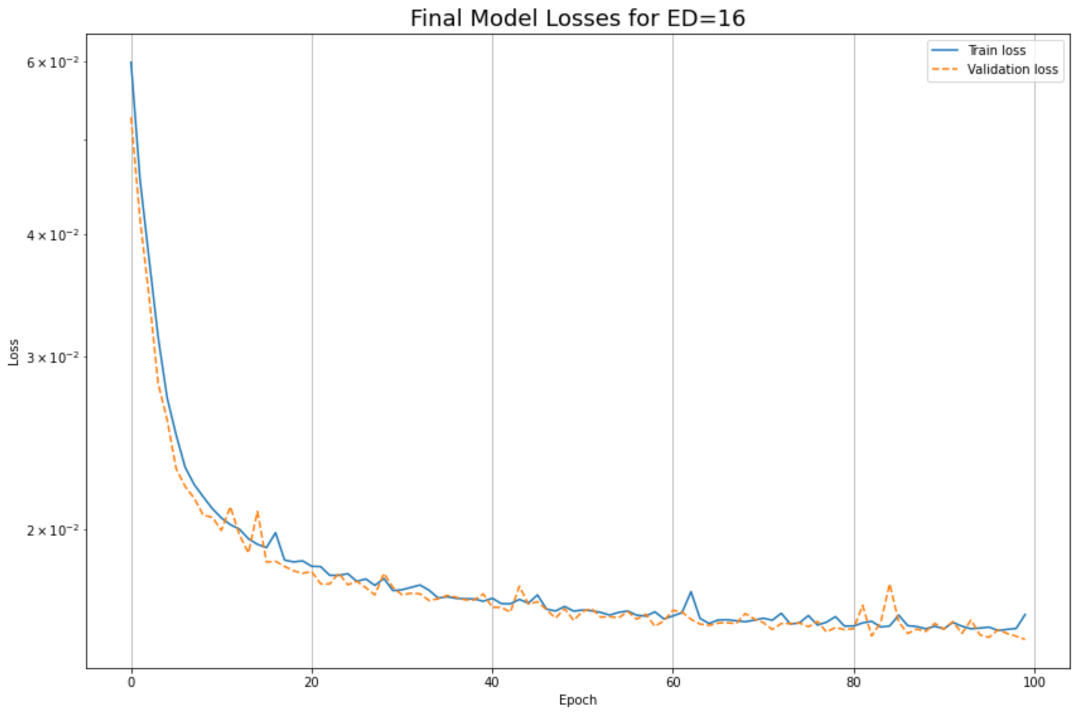


Figure 3: Loss per epoch of the optimized model.

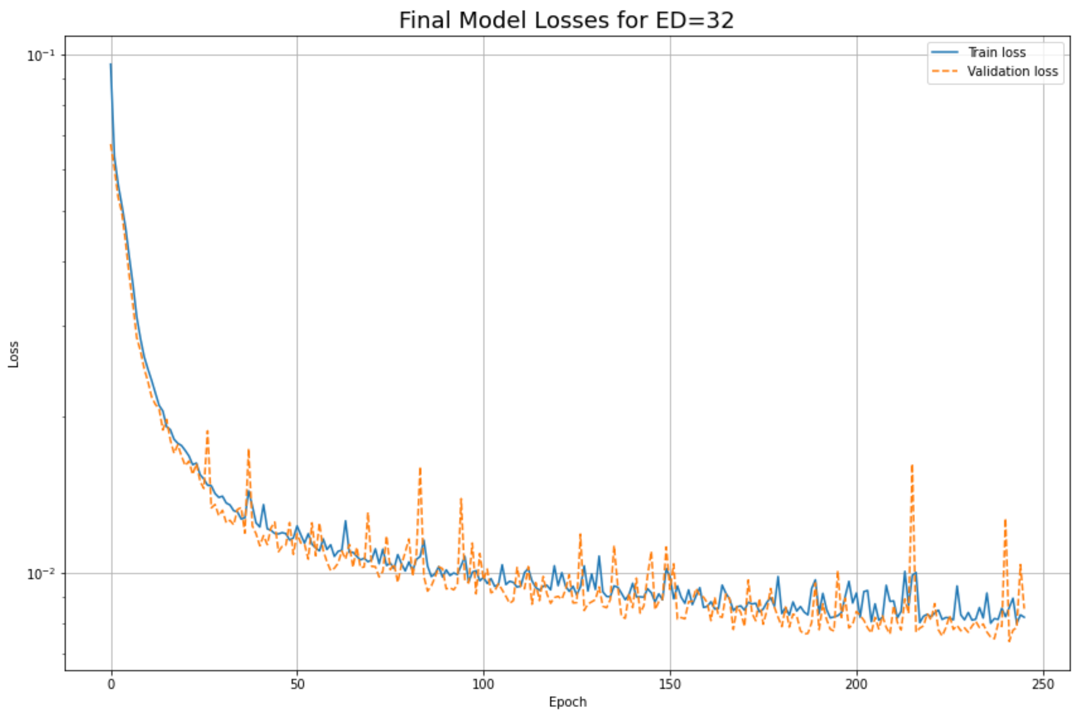


Figure 4: Loss per epoch of the optimized model.

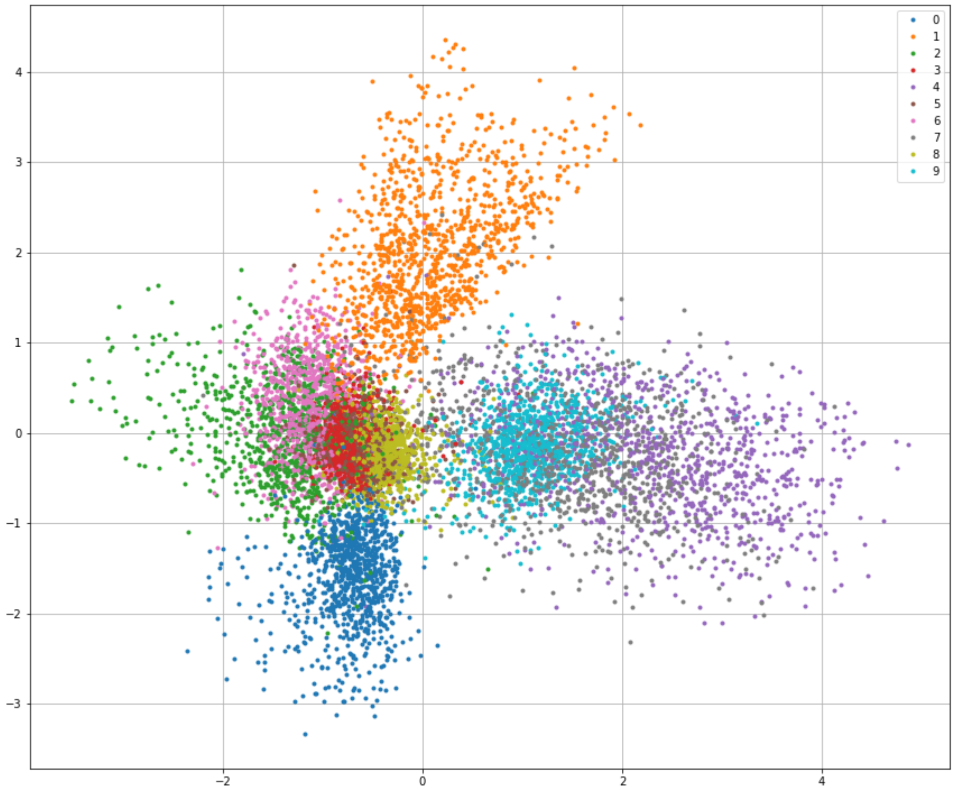


Figure 5: Latent space representation for ED=4, flattened in 2D using PCA.

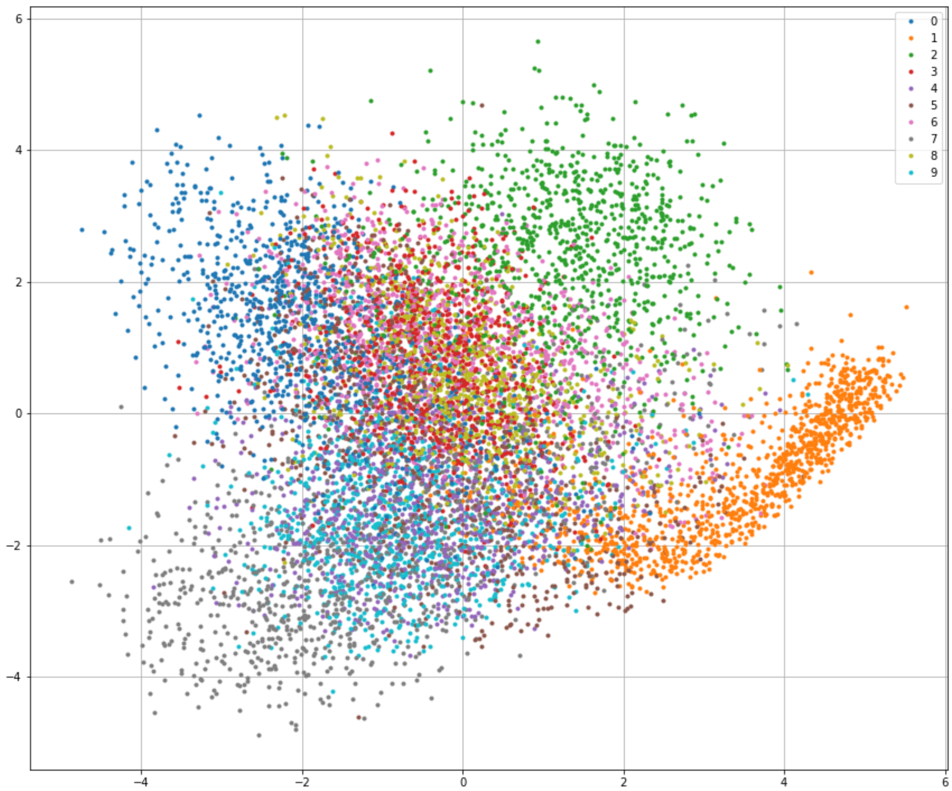
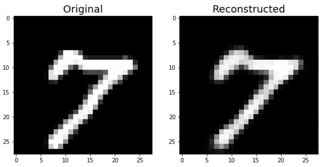
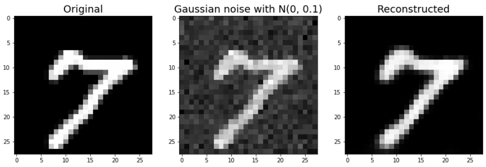


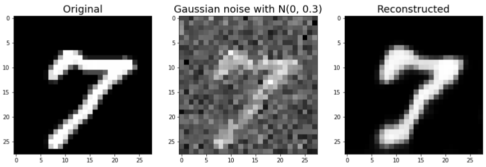
Figure 6: Latent space representation for ED=32, flattened in 2D using PCA.



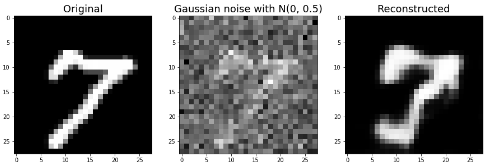
(a) Test without noise.



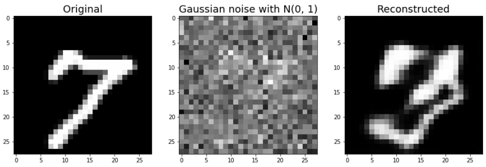
(b) Test with $\mathcal{N}(0, 0.1)$ noise.



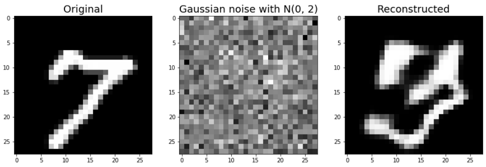
(c) Test with $\mathcal{N}(0, 0.3)$ noise.



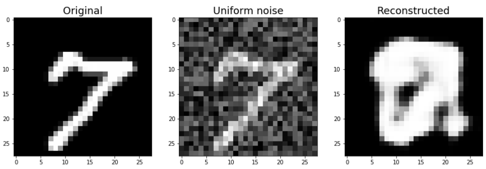
(d) Test with $\mathcal{N}(0, 0.5)$ noise.



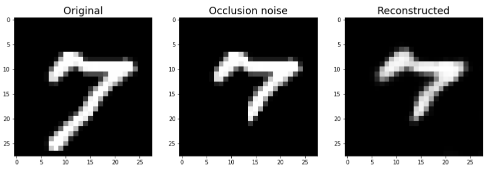
(e) Test with $\mathcal{N}(0, 1)$ noise.



(f) Test with $\mathcal{N}(0, 2)$ noise.

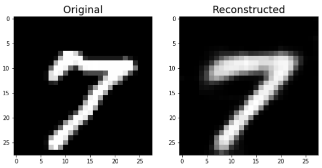


(g) Test with uniform noise.

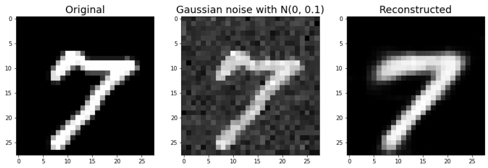


(h) Test with occlusion noise.

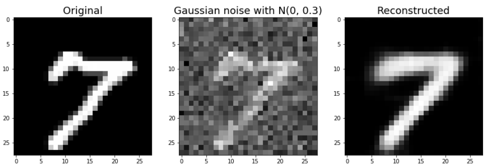
Figure 7: Reconstruction examples for ED=32.



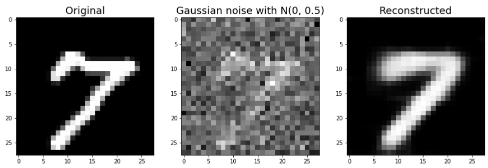
(a) Test without noise.



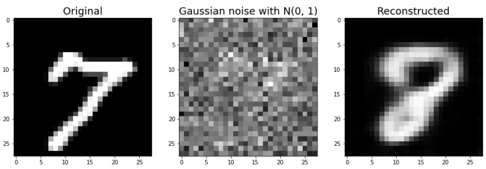
(b) Test with $\mathcal{N}(0, 0.1)$ noise.



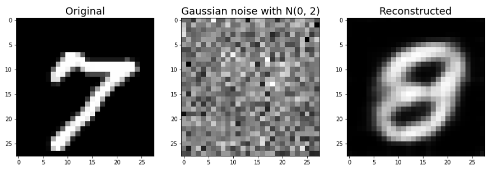
(c) Test with $\mathcal{N}(0, 0.3)$ noise.



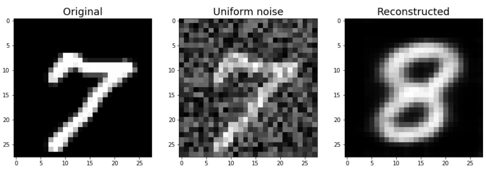
(d) Test with $\mathcal{N}(0, 0.5)$ noise.



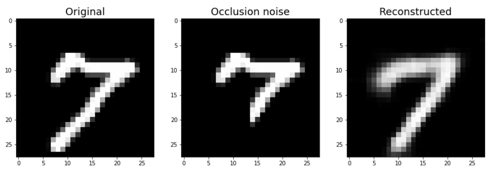
(e) Test with $\mathcal{N}(0, 1)$ noise.



(f) Test with $\mathcal{N}(0, 2)$ noise.



(g) Test with uniform noise.



(h) Test with occlusion noise.

Figure 8: Reconstruction examples for ED=4.