DAMA61 - 6th assignment

Exercise 1

1-5

All the instructions from the exercises were followed.

Console output:

```
Latent dimension 10 - Reconstruction Error: 0.0014080990804359317
Latent dimension 100 - Reconstruction Error: 0.00036464014556258917
Latent dimension 250 - Reconstruction Error: 5.670843893312849e-05
```

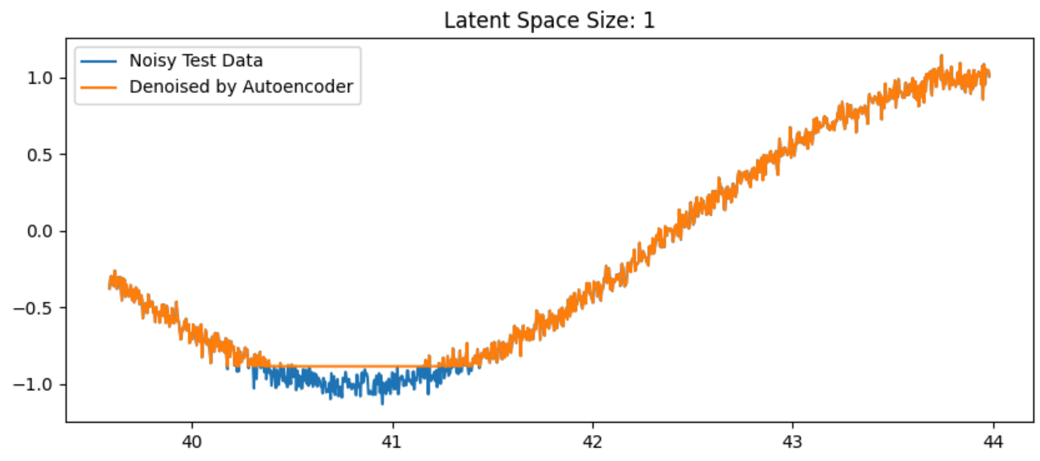
Comment:
From the results we can see that as the number of nodes in the latent space increases, the reconstruction error decreases. More specifically, with latent dimension number 10 the autoencoder struggles more to capture the essential features of the data, leading to a higher reconstruction error. Increasing the latent dimension to 100 provides a substantial improvement in the reconstruction error. This indicates that the model can now encode more information and reconstruct the images more accurately. When increasing the dimension to 250 there is even more of an improvement (the readable number is close to 0.0000567) but not as significant as the 10 to 100 dimension improvement. The 250 dimension model is the most accurate of the three, but when taking into consideration the runtime and performance the 100 dimension one is good enough for the task given.

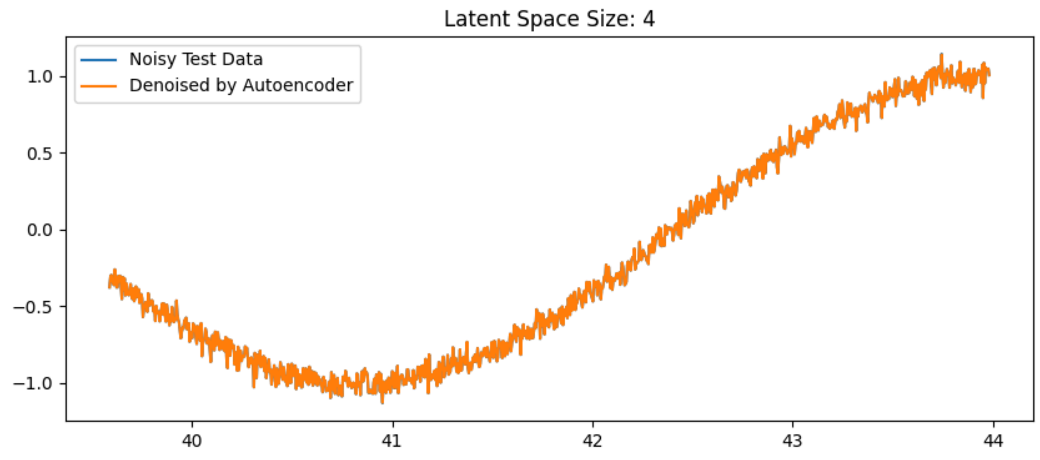
DAMA61 - 6th assignment

Exercise 2

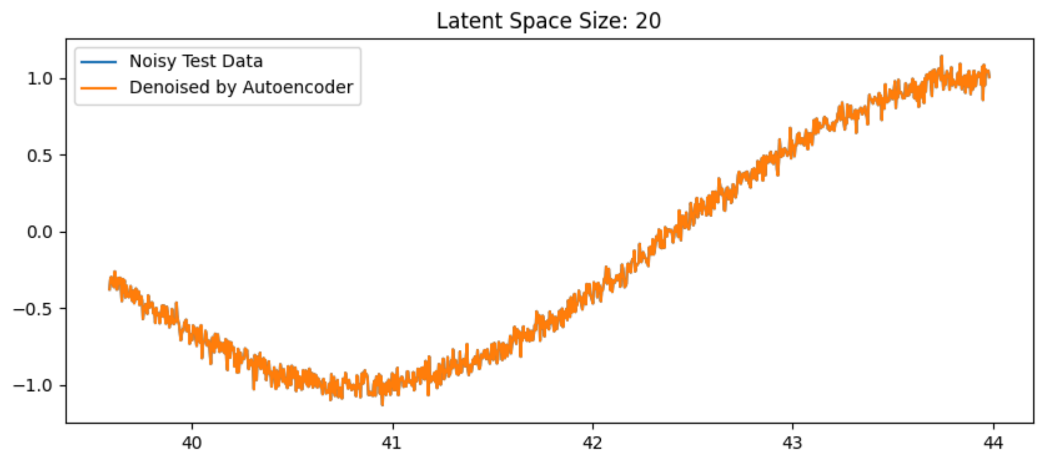
1-5

All the instructions from the exercises were followed. The code produced the following plots:





DAMA61 - 6th assignment



Comment:
All autoencoders, regardless of latent space size, manage to reduce most of the noise in the test data. The autoencoder with a latent space of 1 node struggles to capture the essential features of the original signal as seen by the plot. The latent space of 4 nodes is sufficient to effectively denoise the signal without significant loss of information. Increasing the size to 20 nodes does not visibly improve the quality of the denoised signal. Given that, it implies that a smaller latent space (close to or equal to 4 nodes) is adequate. Using fewer nodes reduces the model complexity and computational requirements, making the autoencoder more efficient without sacrificing performance.