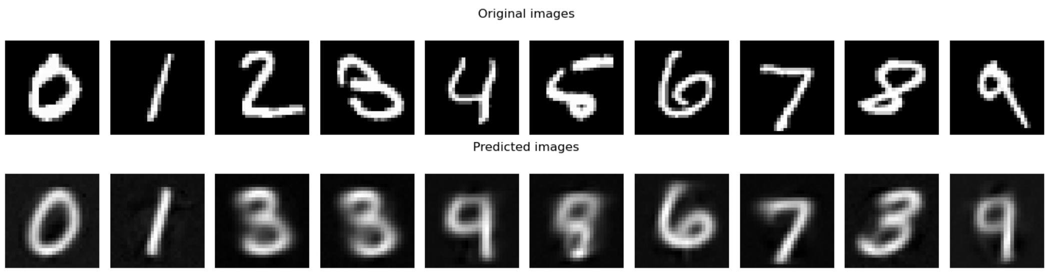
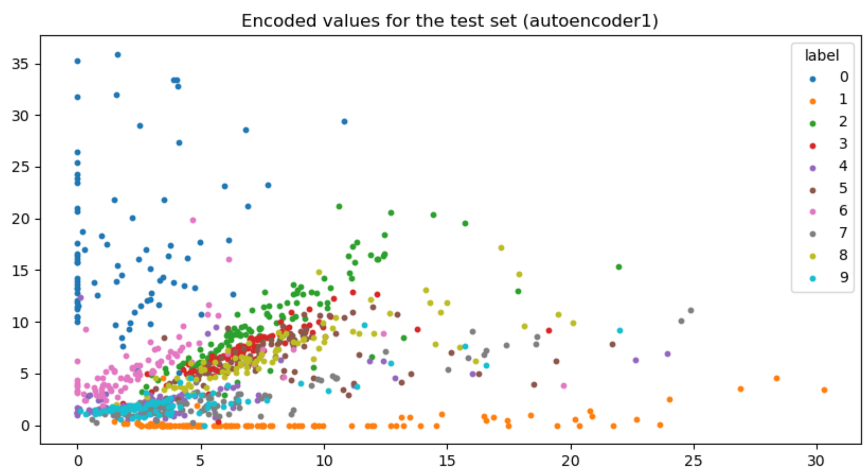
2 Part 2: Fully Connected Autoencoder

2.1 Autoencoder 1

Autoencoder 1 uses only two neurons in the bottleneck layer and is hence not particularly good at reconstructing images from the test set. In Figure 1a we see the a reconstruction of the first observation of each digit in the test set. We see that digits 0 and 1 are very well reproduced which can be explained by inspecting the scatterplot in Figure 1b. Here observe that zeros and ones are clearly separated with the rule of $V_1 \approx 0$ for zeros and $V_2 \approx 0$ for ones. One could argue that the decision boundary is are $V_1 = V_2$ since the rest of the digits are close to this line. Digits 6, 7 and 9 are also quite well reproduced in Figure 1a, but from observing the scatterplot it is hard to pick values that predicts the correct digit, especially for 7 and 9.



(a)

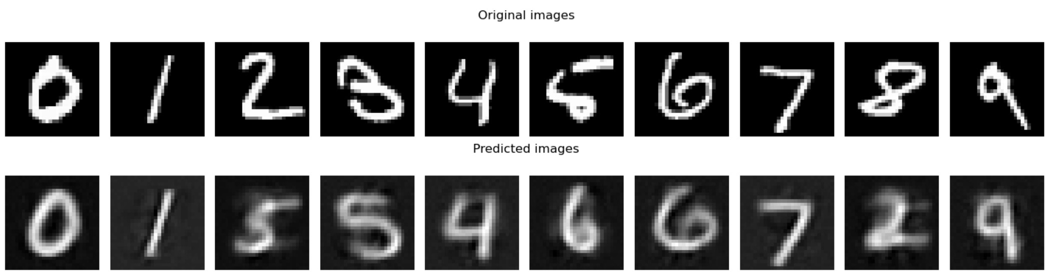


(b)

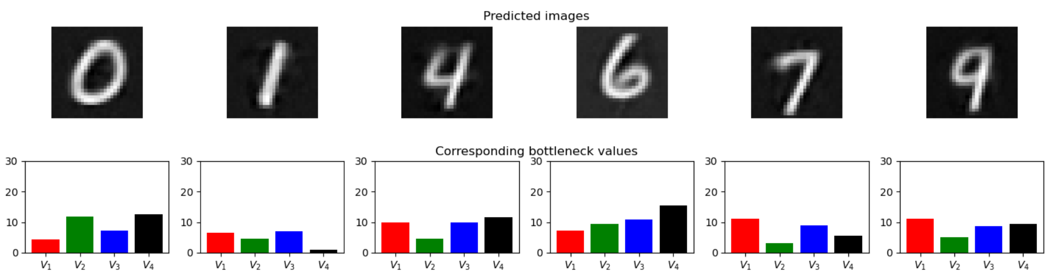
Figure 1: (a) Reconstruction from test set using autoencoder 1. Top row are original images and bottom row are reconstructed images. (b) A scatterplot over the encoded values for the test set using autoencoder 1. The x-axis and y-axis represents the values of the bottleneck neurons V_1 and V_2 respectively.

2.2 Autoencoder 2

Autoencoder 2 uses four bottleneck neurons and performs better than autoencoder 1 at the task of reconstructing the digits. We see in Figure 2a that autoencoder 2 in addition to zeros, ones, sixes, sevens and nines, also reconstruct digit 4 well. To find the rule used by the encoder I calculated the mean of each bottleneck neuron for each digit that was well classified to how they differ. These mean values are represented by the bars in the bar-plots in fig:Ng4, which are positioned under the corresponding prediction performed using these mean values. By looking at the bar-plots we see that each digit that are well-classified, are positioned in a separate part of the four dimensional space spanned by V_1, V_2, V_3, V_4. For example, zeros seem to go under the rule $V_1 < V_3 < V_2 < V_4$, sevens are possibly restricted to the space where $V_2 < V_4 < V_3 < V_1$ and so on. The digits that are not well-classified are probably assigned sub-spaces where multiple observations are positioned when using these particular settings. However, the network should be able to separate the different digits well with the right settings and training, given that the bottleneck layer creates a four dimensional space.



(a)



(b)

Figure 2: (a) Reconstruction from test set using autoencoder 2. Top row are original images and bottom row are reconstructed images. (b) Top row showing a decoding prediction by inspection of the most well reconstructed images of the test set using autoencoder 2. Bottom row showing the corresponding bar-plot that illustrates the values of the bottleneck neurons used for the prediction.