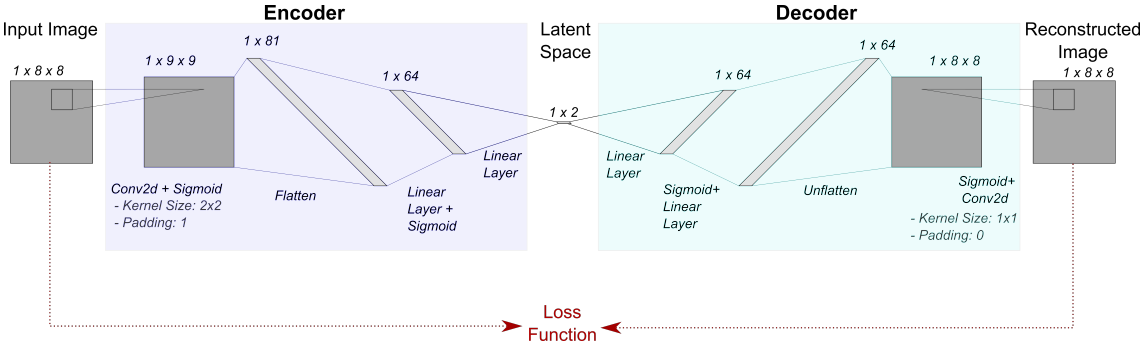
Exercise Sheet 10

1. Exercise: Autoencoder

An autoencoder is a type of artificial neural network used to learn efficient data patterns in an unsupervised manner. Its main function is to learn a compact representation of input data, enabling dimensionality reduction. Autoencoders work by compressing the input into a latent-space representation and then reconstructing the output using this representation. The network teaches itself to minimize the differences between the original data and its reconstruction. With minor modifications to their architecture, autoencoders can also be utilized for generative tasks. Using the Pytorch library the following Autoencoder architecture is to be implemented and trained in this exercise sheet.



(a) Load the digits dataset provided by the scikit-learn (sklearn) library. Split the dataset into random train and test subsets, with the train subset accounting for 80% of the total data.

(b) Normalize the feature values of the data so that they fall within the range of $[-1, 1]$.

(c) **Model Definition:** Implement and initialize the Autoencoder architecture shown above utilizing the Pytorch library. Create a class by implmenting the following functions.

```
class Autoencoder(nn.Module):
    def __init__(self, feat_dim, latent_dim):
        '''Initialize the network using the architecture
        dimension and operations.'''


    def forward(self, x):
        '''Implement the forward path of the neural
        network and return the reconstructed image.'''
        return y
```

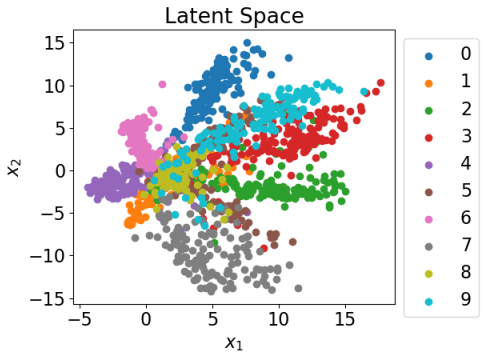
The forward computation is structured with convolution and linear layer operations, using the sigmoid function for nonlinearity. Refer to the figure above for the specific network configurations.

(d) **Loss Definition:** The Mean Squared Error (MSE) loss function should be employed for the training task. It quantifies the discrepancy between the input image and the reconstruced image.

(e) **Optimizer Definition:** PyTorch provides many different built-in optimization algorithms like Stochastic Gradient Descent (SGD) which can be used to update the weights of your model. Define a SGD-optimizer for the parameters of the initialized model with the learning rate $lr = 0.1$.

(f) Train the Autoencoder's parameters by iterating through the training dataset for a total of 15 epochs, ensuring to update your model weights at each prediction step. Shuffle the order of your training dataset for each epoch. Save your loss values and the latent space representations in an appropriate data structure.

(g) Plot the latent space representation of the training data as shown in the figure below. Provide an analysis and interpretation of what the plot indicates.



(h) Generate a plot for three different digits that displays both the original input image and its corresponding image reconstruction, as examplary shown in the following figure.