Report for exercise 4 from group H

Tasks addressed:	4
Authors:	Ahmad Bin Qasim (03693345)
	Kaan Atukalp (03709123)
	Martin Meinel (03710370)
Last compiled:	2019–12–02
Source code:	https://gitlab.lrz.de/ga53rog/praktikum-ml-crowd

The work on tasks was divided in the following way:

Ahmad Bin Qasim (03693345)	Task 1	33%
	Task 2	33%
	Task 3	33%
	Task 4	33%
	Task 5	33%
Kaan Atukalp (03709123)	Task 1	33%
	Task 2	33%
	Task 3	33%
	Task 4	33%
	Task 5	33%
Martin Meinel (03710370)	Task 1	33%
	Task 2	33%
	Task 3	33%
	Task 4	33%
	Task 5	33%

Report on task 1, Principal component analysis

First part:
Energy of the two components: energy of pc1: 0.9929871280253524 energy of pc2 0.0070128719746476165
Second Part: Rerun it using columns rather than rows
Third Part: They are walking in loops. energy of pc1: 0.47330561274983274 energy of pc2 0.3759408098565421 Total Energy is around 84%. Two components are enough to capture the most energy. It is enough because in comparison to the original plot the reconstructed plot looks similar and still captures the most information.

Report on task 2, Diffusion Maps

Part 2:
Energy values: pc3 0.28867937283177014 pc2 0.3292051971173148 pc1 0.3821154300509151
The first two Principal components cover only 71.1% So we would lose 28.9% of the data. In comparison to that we can cover almost the whole dataset by taking the first three principal components. Consequently, it makes much more sense to take all principal components instead of two.

Report on task 3, Training a Variational Autoencoder on MNIST

1. We used a linear activation function to approximate the mean and standard deviation of the posterior distribution because, the mean and standard deviation values are unbounded. Other activation functions bound the output value to a certain range. We considered different activation functions but discarded it because of the bounded output. [1]

Table 1: Different activation functions

Activation Function	Bounds
Sigmoid	(0,1)
Tanh	(-1,1)
Relu	max(0,x)

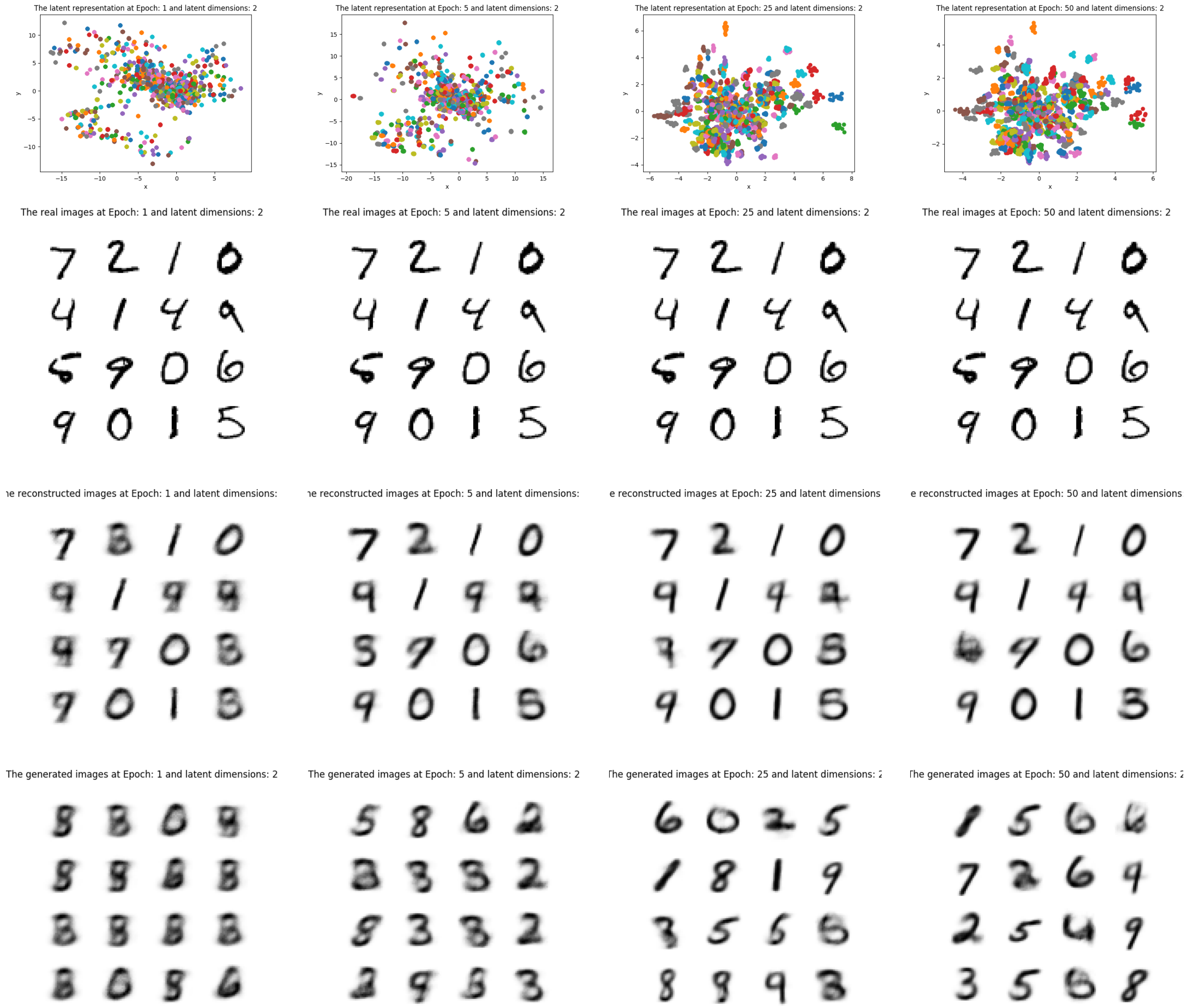
2. If the reconstructed image are much better then the generated images then, it means that the model has been overfitted to the input data distribution. This can happen when the KL-divergence loss (latent loss) of the approximated posterior distribution does not converge while the reconstruction loss of the input distribution converges.

   In order to solve this problem, the reconstruction loss and the KL-divergence loss can be weighted to give more weight to the latter.

3. After training the Variational Autoencoder model, We plotted the desired graphs. The latent space of the encoder is 2 dimensional. [1]. In addition to the configurations provided in the exercise sheet, related to the model, the following changes were made:

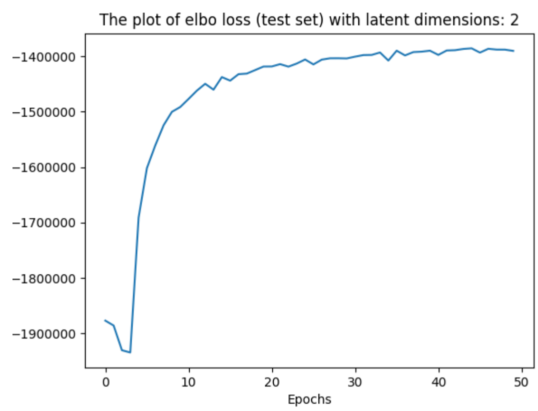
   - The output of the decoder is not the mean of likelihood but the predicted data points. This modification within the decoder can be considered as probabilistic as well because it is equivalent to modeling likelihood as Gaussian with identity covariance
   - Binary cross entropy is used to calculated the reconstruction loss with the predicted data points from decoder and the input data as the target.
   - The last layer of decoder uses a Sigmoid activation as the data in mnist is grayscale.

Figure 1: results obtained with a 2 dimensional latent space



4. As the VAE is trained for more epochs, the loss decreases. [2]

Figure 2: ELBO loss



5. While initializing the VAE model class object, a configuration dictionary can be sent as an argument to the init function, to dynamically change the model. There are a number of configurations that can be set using the dictionary. [2].

Table 2: The VAE class initialization arguments

Argument	Description
latent_vector_size	The dimensions of the latent vector
print_output	Print the output plots (True or False)
batch_size	Batch Size to be used
learning_rate	Learning rate
epochs	Number of epochs
train_dataloader	Pytorch dataloader for training set
test_dataloader	Pytorch dataloader for test set
dataset_dims	The dimensions of the dataset
mode	The dataset mode (mnist or mi)
test_count	The number of test set data points

The mnist images generated using 32 dimensional latent space are more sharper. [3]. The loss in this decreases more rapidly and within 50 epochs, the total loss per epoch reaches a lower value compared to the VAE trained with 2 dimensional latent space. [4]

Figure 3: results obtained with a 32 dimensional latent space

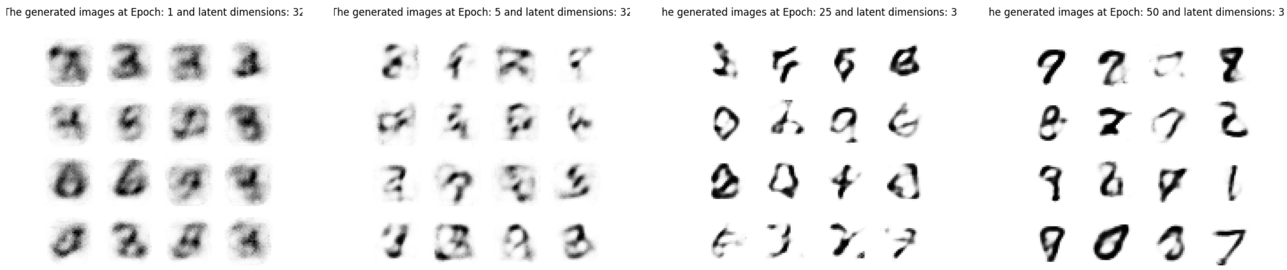
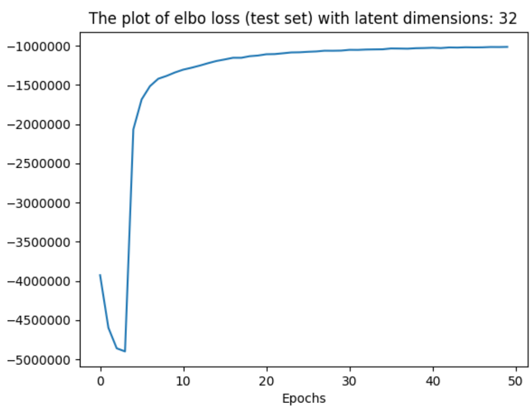


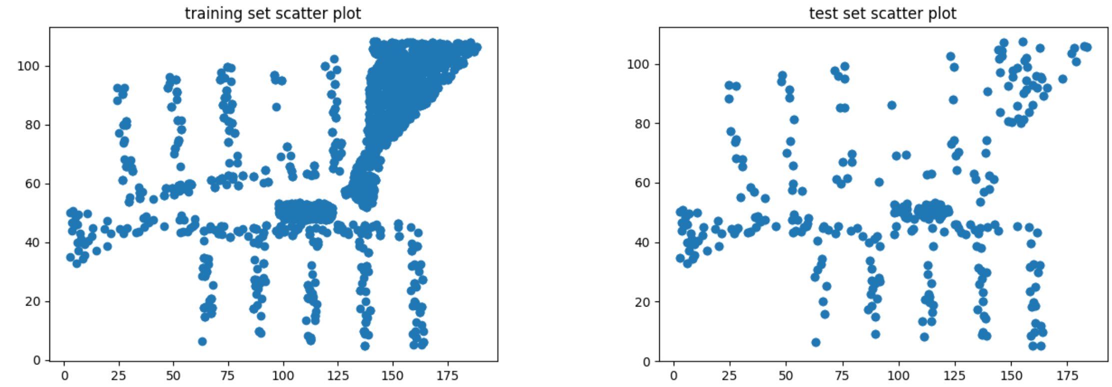
Figure 4: ELBO loss



Report on task 4, Fire Evacuation Planning for the MI Building

1. The training and test datasets reflect the positions of the students and employees in the campus hallways.

Figure 5: The scatter plot of the training and test sets



2. The same VAE implementation is used for training a VAE on the FireEvac data. The 'mode' argument is set to mi and the 'dataset_dims' argument is set to 2, as the FireEvac dataset is 2 dimensional. A number of modifications have to be made to train the FireEvac dataset:
   - The VAE is trained for 1000 epochs.
   - The learning rate of the adam optimizer is set to 0.0001 because the higher learning rate of 0.001 used in the previous task proves to be too high for training the FireEvac data. It results in a jittery model training.
   - Mean Squared Error is used for calculating the reconstruction loss. The predicted data is the output of the decoder and the target is the input data.
   - Relu activation function is used for the last layer of decoder, as the FireEvac data consists of positive unbounded coordinates.
3. The distribution of the data reconstructed from test set, is similar to test set distribution. [6]

Figure 6: ELBO loss