In this assignment, you will learn how to work with variational autoencoders in PyTorch.

Question 1a

Implement a fully connected VAE trained on the MNIST dataset. Use a latent space with only two latent variables and a mean-field Gaussian decoder.
If you feel self confident, I recommend you write all the code yourself. This is not too difficult if you have experience with PyTorch and you understand the concepts explained during the lecture.
If you think that you need guidance, then you can decide to copy the code from the following repository:

https://colab.research.google.com/github/smartgeometry-ucl/dl4g/blob/master/variational_autoencoder.ipynb

The given code uses convolutional architectures, you need to change it into fully connected ones. This can be done with relatively few modifications.
If you decide to do so, it is important that you study the code in detail as you will need to modify it further the code in the following questions.

Question 1b

Train the VAE you just used in the previous assignment on the fashion MNIST dataset. Evaluate performance and compare the results.

Question 2

Replace the mean-field encoder with a full covariance encoder as described in the slides. This means that the encoder network has to output both mean vector and the triangular factor matrix L (called scale_tril in the multivariate normal implementation in the PyTorch distribution library: https://pytorch.org/docs/stable/distributions.html).
Test the resulting modified VAE on MNIST and fashion MNIST. Compare performance with the mean-field version.

Question 3

Change the encoder and decoder architectures from fully connected to a ConvNet architecture of your choice. Test on the dataset CIFAR10. Interpret the results.