**IIDS67682 - Neural Network Labs Day 2**

Today’s exercises involve the practical understanding of autoencoders. You should aim to complete at least Q1-3 below.

To get the python notebook files, you will need to go into elab and run the command ‘**fetchme vaelab**’ from a terminal. This will fetch the code for this week’s lab.

1. Look at the code in pytorch-cnnae.ipynb. Draw a block diagram of the autoencoder architecture.
2. At the end of the pytorch-cnnae.ipynb notebook a dataframe is created that contains the latent embeddings for 1000 test set digits. Use PCA and tSNE to visualise the latent digit embeddings. Do the latent embeddings reflect the class labels well?
3. Use k-means clustering to cluster the digits based on their latent embeddings. Use the PCA/tSNE visualisations and a cross-tab table to compare your cluster labels with the digit labels. Which digits tend to cluster with other digits? Do you think the results make sense in terms of the similarity of different digits?
4. In the lectures you heard about latent space arithmetic. Investigate how arithmetic works with pytorch tensors (see e.g. <https://jhui.github.io/2018/02/09/PyTorch-Basic-operations/>). How do the AE or VAE reconstructions look that are in-between two embedded digits? What happens as you move from one embedded digit to a different embedded digit in the latent space?

Hint: To create a latent embedding which is 0.2 part of img1 and 0.8 part of img2 you can use the following code:

**latent\_mixture = torch.add(torch.mul(encoder(img1),0.2),torch.mul(encoder(img2),0.8))**

You can use this code to plot images which are made up of different mixtures of img1 and img2. You will have to use the decoder to create and plot the resulting images. Some of the code in the function **plot\_ae\_outputs** can be usefully adapted for this task.