# LAB 8 - EXPLORING LATENT SPACES

*Supritha Konaje - 32864477*
ssk1n21@soton.ac.uk

## 1. EXPLORING THE LATENT SPACE OF A VAE

### 1.1. Systematically sample a VAE

The code snippet to generate latent image is given below:

```
enc.eval()
dec.eval()
cnt = 21
latent_x = torch.linspace(-4, 4, cnt)
latent_y = torch.linspace(4, -4, cnt)
image_height = 28
image_width = 28

large_image = np.zeros((588, 588))
for i, x in enumerate(latent_x):
  for j, y in enumerate(latent_y):
    z = torch.tensor((x, y)).
        unsqueeze(0).to(device)
    output = dec(z)
    output = output.cpu().detach().
    reshape(image_height,image_width)
        large_image[j*image_height:(j+1)*
        image_height, i*image_width:(i+1)*
        image_width] = output

plt.imshow(large_image, cmap=plt.
    get_cmap('gray'))
plt.show()
```

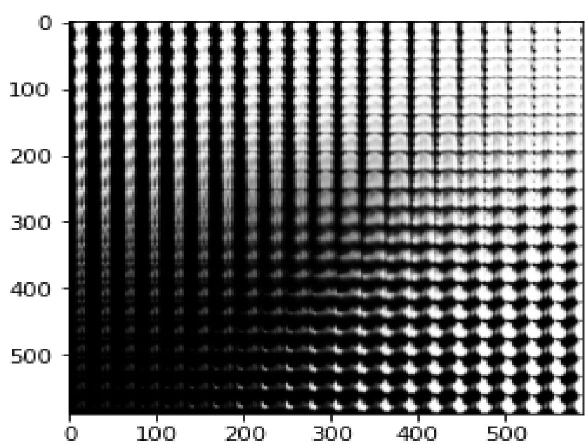


**Fig. 1**. Representation of Variational Encoder

## 2. EXPLORING THE CODE SPACE OF A STANDARD AUTO-ENCODER

### 2.1. Systematically sample an Autoencoder

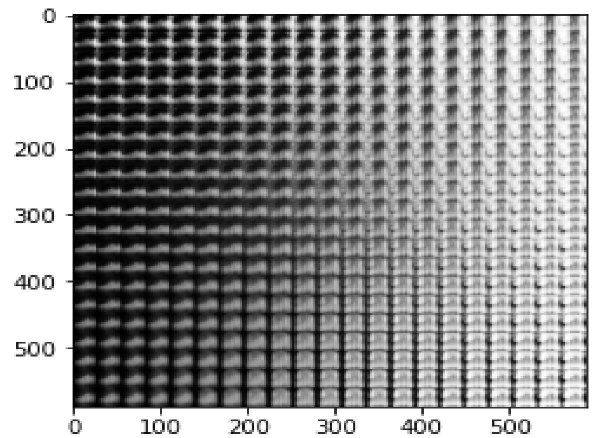


**Fig. 2**. Representation of AutoEncoder

### 2.2. Compare the latent spaces of the VAE and autoencoder

Variational-autoencoders are not only trained based on how successfully they reconstruct their input, but they are also penalised if their latent space distribution deviates from the assumed (prior) probability distribution. It's possible to tweak the relative relevance of these various goals.

Autoencoders employ neural networks to compress input into a latent space and then reconstruct the input from there.