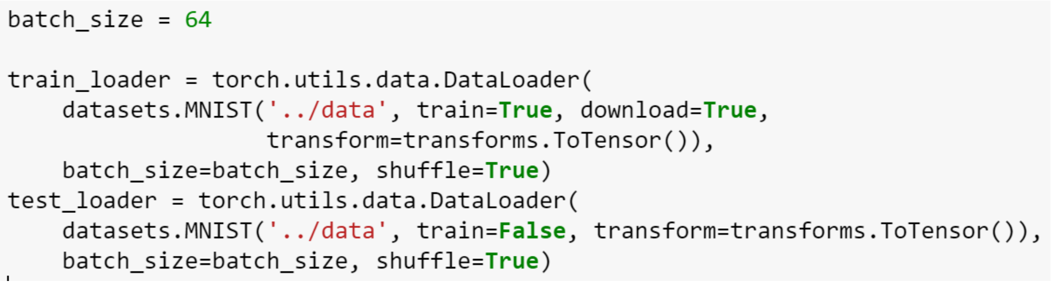
Train a VAE on the MNIST dataset (handwritten digit). You can get the MNIST dataset with the library torchvision, using:



```
batch_size = 64

train_loader = torch.utils.data.DataLoader(
    datasets.MNIST('../data', train=True, download=True,
                   transform=transforms.ToTensor()),
    batch_size=batch_size, shuffle=True)
test_loader = torch.utils.data.DataLoader(
    datasets.MNIST('../data', train=False, transform=transforms.ToTensor()),
    batch_size=batch_size, shuffle=True)
```

You can iterate over this object, that yields tuples x, y, x being batch of pictures (size batch_sizex1x28x28) and y being the number that is actually drawn in the picture

In terms of architecture, you can use either only fully connected layers, or convolutions for the encoder and deconvolutions for the decoder (transpose convolution in pytorch).

Use a 2-dimensional latent space

1. What is the loss of the problem?
2. Train the model and monitor the training and validation losses over time
3. Try to generate new samples
4. Plot the latent representation, colored with the label of the points. What do you observe ?
5. Can you think of a modification of your model that would allow to generate exactly the number you want ? i.e, a model allowing conditional generation ? Train such a model (redo questions 2,3,4,5 for this one).