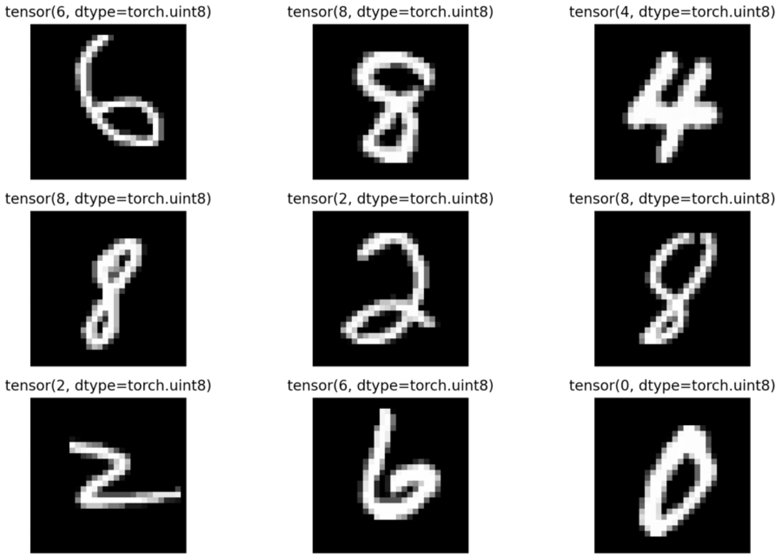
TP03 :

Question 1 :



Question 2 :

The weights of the decoder are usually the transposed weights of the encoder. What is the advantage? Indeed, the encoder part of an Auto-Encoder is supposed to extract the features, then the decoder must use the features to recreate the original image. By using the transpose of the encoder weights for the decoder, we avoid re-training the decoder and we can immediately have the original image. Since the weights learned to obtain the features can be used to find the image by transposing them.

Question 3 :

For the test campaign, we first used a 2-layer hidden network composed of 2D convolutions, Batch Normalisation and LeakyReLU activations. Then we used an autoencoder that was taught both to classify using only the encoder part and also to reconstruct the input image using the encoder and decoder module.
We have made a few tests on both of these modules and here are the results:

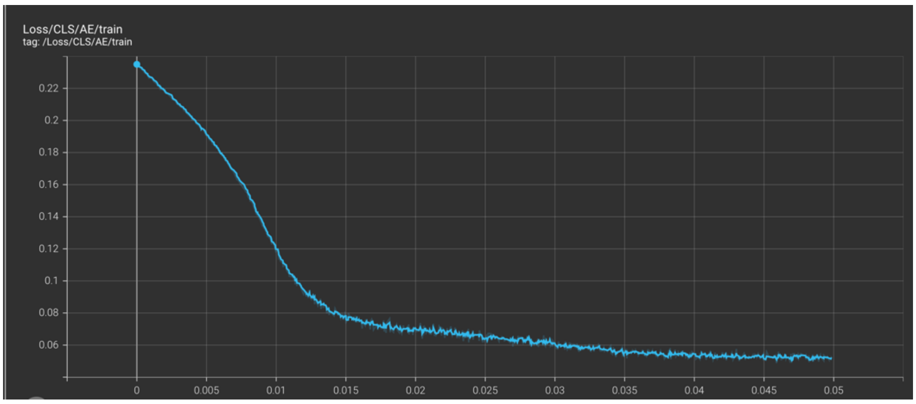


Figure 1. AutoEncoder Classify Loss

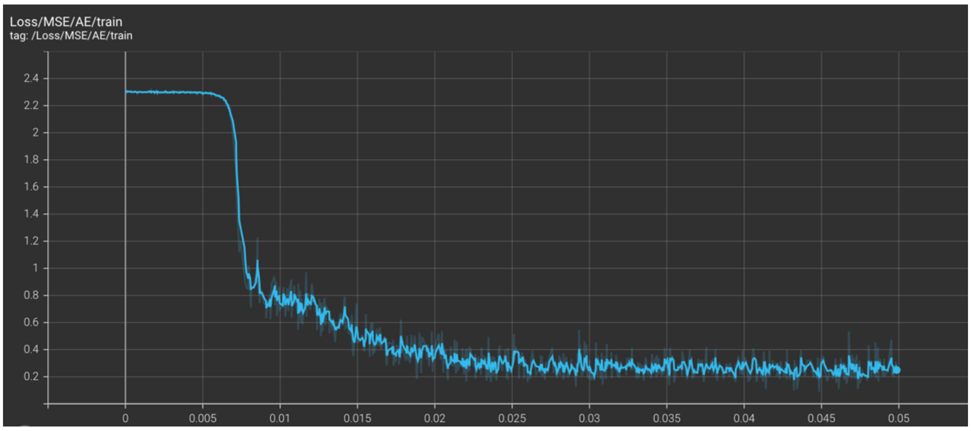


Figure 2. Autoencoder MSE Loss

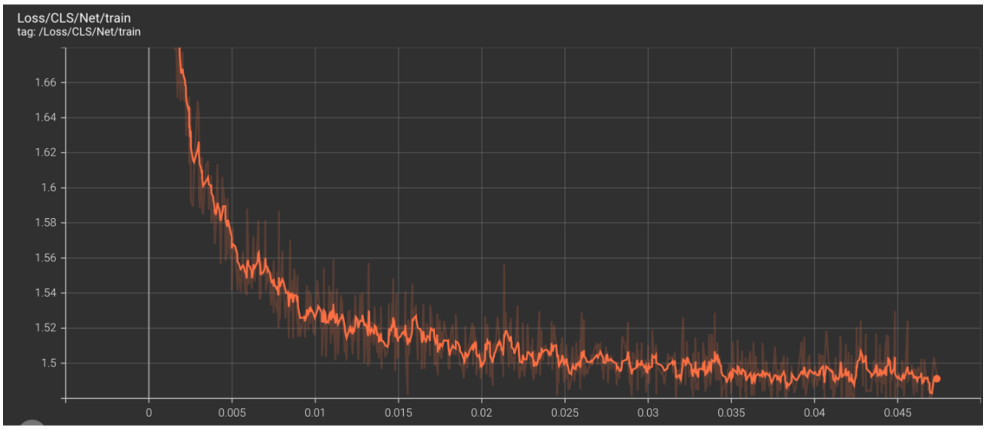


Figure 3. Simple Net Loss

We see that that the Autoencoder is better at classifying than the Simple Net We obtain a low loss with the AE on the classifying task (around 0.3) whereas we have a value of 1.4 with the Simple Net.

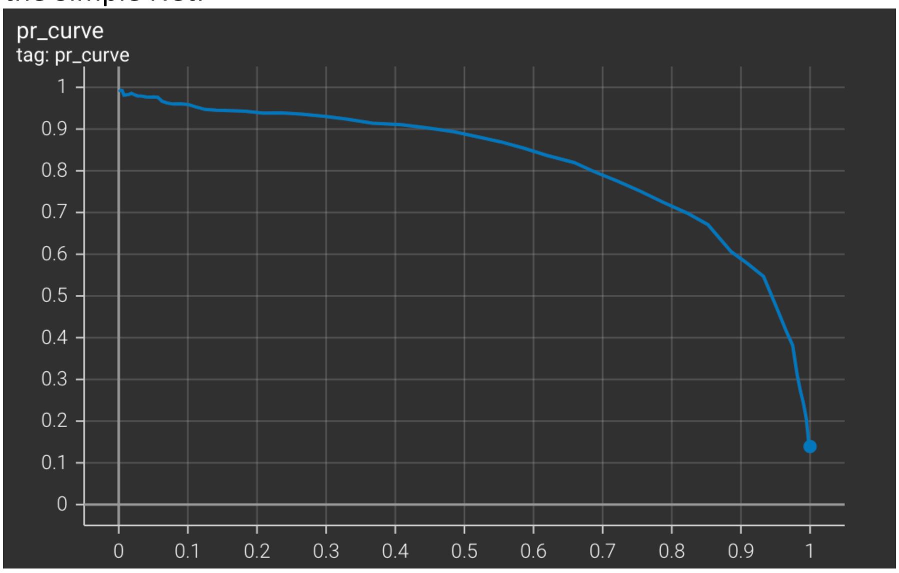
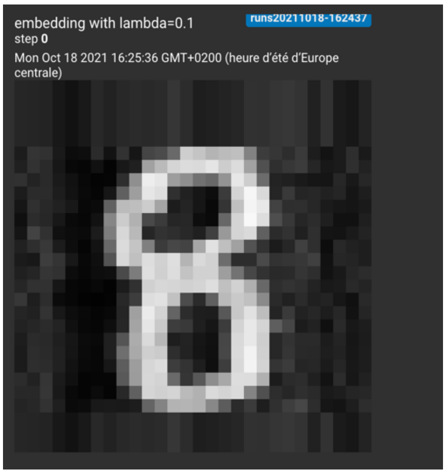
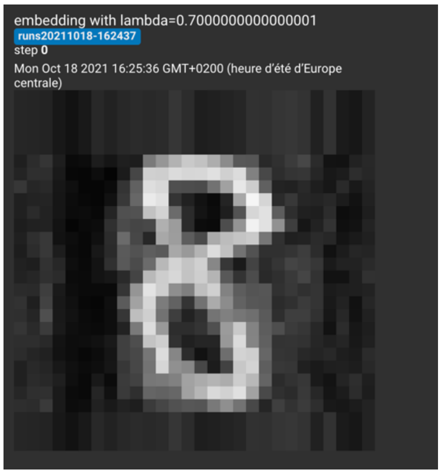


Figure 4. PR curve for Auto encoder

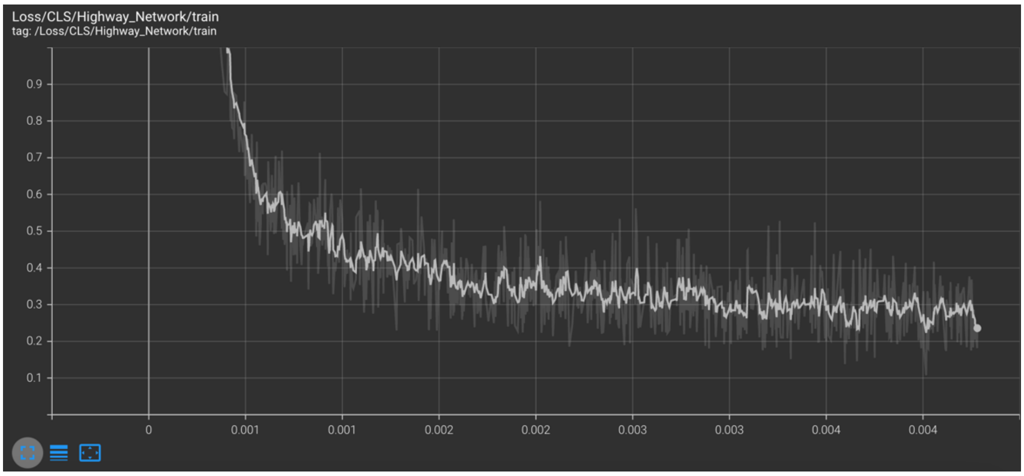
We also have computed the PR curve for the Auto Encoder, the goal of the PR curve is to find a balance between Precision and Recall. A huge area under the curve is a sign that the model is working. Here we see that we always have a value between 0.9 and 1 so this is good.

Finally, we also have ran some test on the interpolate images to see the image in the latent space with different lambda values:





Question 4:



On the classifying task the results are comparable for the Highway Network and the Autoencoder. The reason might be the data that are not very complex so we see less the differences between the models.