AML Assignment 2

Paarth Iyer - MCS202218

VAE

The architecture, for both the encoder and decoder, is a sequence of linear layers with the activation functions being leaky ReLU.
The latent dimension was set to be 50.

$$Encoder : 784\ (28 * 28) \rightarrow 1000 \rightarrow 1000 \rightarrow 200 \rightarrow 50(mu) + 50(logvar)$$
$$Decoder : 50 \rightarrow 200 \rightarrow 1000 \rightarrow 1000 \rightarrow 784\ (28 * 28)$$

Results

While training process, it was observed that the classes with similar features, like T-shirt/top and pullover seemed to have similar look. This could be explained by those having similar shape and structure in the small images. This could be seen in the images generated from random latents. The images which look like upper wear are hard to categorize given the classnames Tshirt/top, pullover, shirt and coat. Higher quality details are generally not observable, but the model manages to create and replicate the overall structure of objects.
Looking at the latent interpolations for images in the dataset, it can be seen that while interpolating, some of them temporarily change to a class which none of the initial two images belong to. This tells us that the latent space may have a good clustering of classes with not too much empty space between them.

GAN

The model architecture here is a DCGAN.

The discriminator has 4 layers of (convolution, batch norm, leaky ReLU).
The number of channels is as follows : $64 \rightarrow 128 \rightarrow 256 \rightarrow 512$. This is followed by a Flatten operation, after which two dense layers ($512 * 2 * 2 \rightarrow 10 \rightarrow 10$) output a 1 dimension prediction.

The generator takes in a latent dimension of 128. It has a dense layer (which maps the latent dim to a 256*3*3 tensor), followed by 3 conv transpose (channels : $256 \rightarrow 256 \rightarrow 128 \rightarrow 64$). After this a kernel size 1 convolution is applied to merge the channels, and a Tanh activation function at the end.

The weights for the convolutions are initialized normally with mean = 0 and stddev = 0.02. The Adam optimizer is used with lr = 0.0002 and $\beta_1 = 0.5$. These values are taken from the DCGAN paper.

Results

The randomly generated samples look pretty good with most of them being identifiable as numbers from the dataset. It looks like all the classes have some generations, suggesting that a mode collapse has not happened.

The latent space interpolations between two randomly generated seem to result in legible images, with the number seeming to morph naturally.

Conditional GAN

The architecture used for the CGAN is also similar to above, but the conv transpose layers have been replaced by an upsample and a convolution (see : `https://distill.pub/2016/deconv-checkerboard/`).

The discriminator gets the class information by passing the one hot class through a dense layer to make it dim 50 and then to 784, after which it is added to the image as a new channel.
For the generator, the above is repeated so that the class info is attached to the image as a new channel before the upsample convolutions start.

The DCGAN values, as stated in the GAN section, are used again for this model.

Results

The images generated with classes look very good and are easily classifiable into the given class. There is a good amount of variation in the classes, which may suggest that a mode collapse has not happened inside the classes themselves.

An observation : Conditional GAN has much better generation than GAN. One possible reason for this could be that supplying the model with class information allows it to internally separate them, as compared to GAN, where the model may not know the differences between similar looking samples from different classes.