Deep Learning 2020	**(Due: 20/12/2020)**

Submission Assignment 5A

Instructor: Jakub Tomczak *Name:* Maurits Dick,Ho,Yu-Feng, *Netid:* MDK258,YHO700

For this The 5th and final assignment for the course Deep Learning we chose to do the extra assignment A. On latent variable models VAE's and GAN's. After constructing the two models with the help of Pytorch frameworks, we trained the model on two separate data sets. The MNIST hand written digit data set. And the Imagenette a somewhat smaller rendition of the Imagenet data set containing 10 classes. As is the case with most deep learning problems an objective function is fundamental in training models. In the cases for VAE 'Variational Autoencoders' the objection function can be found in equation 0.1. For our second model Generative Adversarial Networks (GANs) the objective function can be found in equation 0.2

$$logp_\theta(x) \geq E_{z\sim q_\phi(z|x)}[logp_\theta(x|z)] - KL(q_\phi(z|x)||p_\lambda(z)) \tag{0.1}$$

$$min_G max_D V(G,D) = E_{x\sim p_{real}}[logD(x)] + E_{z\sim p(z)}[log(1-D(G(z)))] \tag{0.2}$$

1 VAE

A simplified representation of a VEA can be found in figure 1. First a batched input tensor of images is given as an input to this a encoder is applied in our case we used a series of fully connected MLP. This is uses to create a latent distribution $p(z|z)$, from this we able a sampled latent representation $z \sim p(z|x)$. Now with this representation a decoder is applied here we again used a MLP structure. On this the loss function from 0.1 is used. After training the encoder and decoder random noise can be given as input and after sampling the output should be a generated representation of images given in the training set.

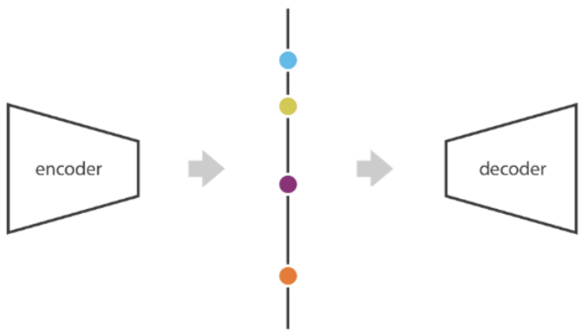

Figure 1: A simplified representation of a VEA

2 GAN'S

Our second model Generative Adversarial Networks (Gans). using a different method to grain the ability to generate images. Here instead of using encoders and decoders. A generator and a discriminator is trained. The generator learns to generated plausible images from random noise the discriminator learns to distinguish the generator's fake data from real data. After training the generator would hopefully be able to create new images from random noise within the set of trained data. A simplified representation of this model can be found in figure 2. In our created model we again used MLP layers with leaky ReLU activation. We toke precautions to make sure the overall complexity of both model were equal by having both have more or less the same amount of trainable parameters.

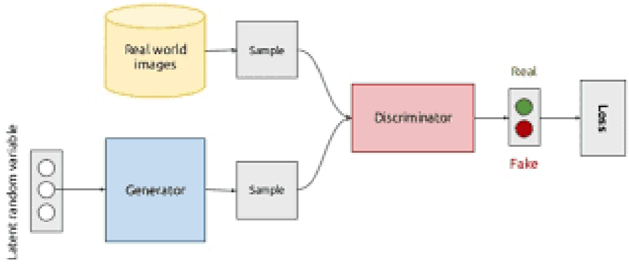

Figure 2: A simplified representation of a Gans

3 Analysis

In this section we will be looking for way to analyse the validity of our created models. This will first be done by looking at the loss learning curve for the VAE and GAN's. After this we will Implement the Frechet Inception Distance (FID) for evaluating GANs and VAE's. A lower FID indicates better-quality images; conversely, a higher score indicates a lower-quality image.

Lets start off by looking at the learning curve for the MNIST data. In figure 4 is loss over 10 epochs is plotted, monotone declaing. FID score was Average FID: 19.52 with Std 4.5. In Lucic et al. (2018) the best Score was average FID on MNIST was 2.60. Our score in compression isn't great but when looking at the generated image by eye they looking better good 6. The GAN's preformed better on the FID scale with Avr 12.7, Std 3.5. When comparing the two batches you can quite clearly make out that the GAN's produces less blurry images. Last but not least, in order to make two models comparable, we structured the models with similar numbers of weights. Total parameters in VAE are 1,474,784. In GAN's, the parameters are 1,486,352 in discriminators and 1,460.225 in generators.

Furthermore, we randomly chose two points, using latent space of VAE and GAN's to generate picture. Then we linearly interpolate 10 points between two points and similarly generate 10 pictures, which represent the transformation between two original pictures 7. The same operation is also for GANs model 8.

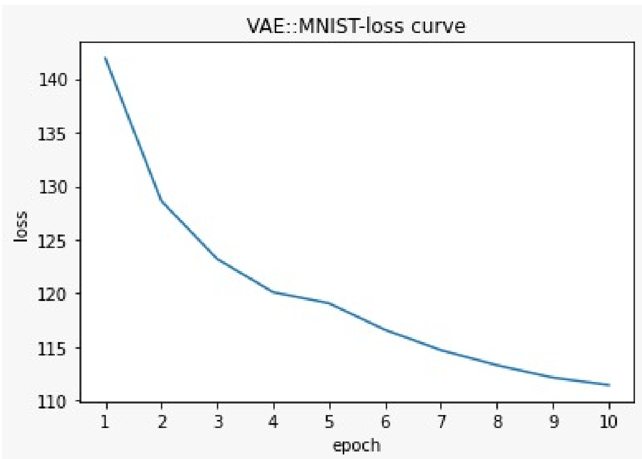

Figure 3: Loss learning curve MNIST for the VAE model

4 Discussion

Regarding Imagenette, the color image dataset, we generated the similar outcomes but the resolution was way more worse. For example using VAE, the generative image is hard to identify 9. The possible reason is the parameters in neural network structure need to be tuned, including numbers of hidden layer, weights and activation functions.

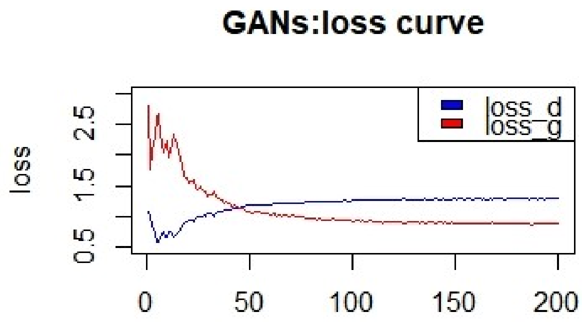

Figure 4: Loss learning curve MNIST for the GAN's model

Figure 5: Batch of digits generated using our VEA model

References

Lucic, M., Kurach, K., Michalski, M., Gelly, S., and Bousquet, O. (2018). Are gans created equal? a large-scale study.

Figure 6: Batch of digits generated using our GAN's model

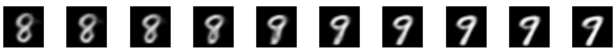

Figure 7: Interpolations of VAE in MNIST

Figure 8: Interpolations of GAN's in MNIST

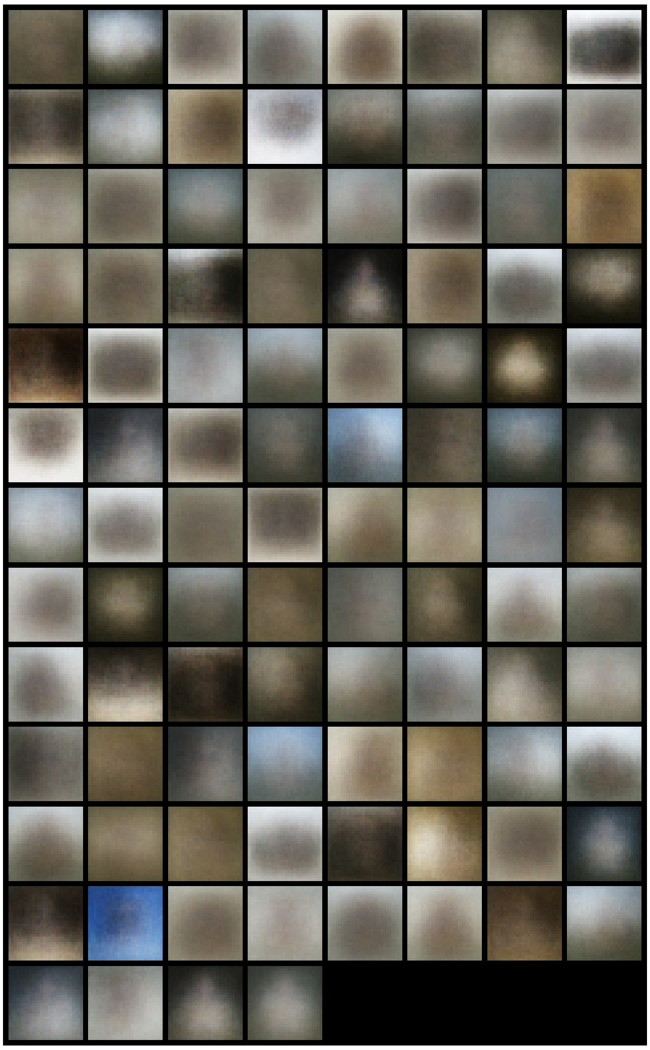

Figure 9: Batch of digits generated using our VAE model in Imagenette