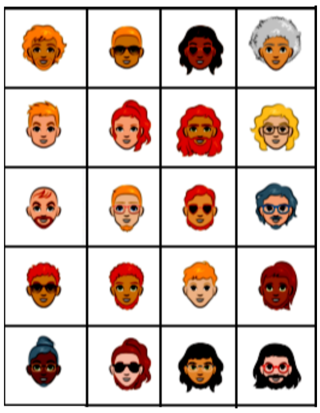
Deep Learning (Homework 3)

Due date : 6/16/2019

- Homework submission – Please zip each of your source code and report into a single compress file and name the file using this format : HW3_StudentID_StudentName.zip (rar, 7z, tar.gz, . . . etc are *not* acceptable)

In this homework, you will use the cartoon character faces dataset. The dataset contains 10000 face images. All the images are with size of 500 × 500. You will use this dataset for the unsupervised image generation as shown bellow



1. **Cartoon Character Generation**

   In this exercise, you will construct a Variational Autoencoder (VAE) for image reconstruction by the provided animation faces dataset.

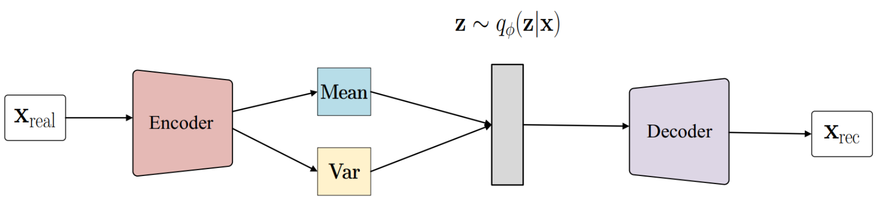


Figure 1: Structure of VAE

   VAE paper can be downloaded here. You should preprocess the images such as resizing or cropping by yourself before implementation.

   i. Describe in details how to preprocess images (such as resize). Implement a VAE for image reconstruction by using convolution layers or fully connection layers. You need to design the network architecture and show it in the report. Finally, plot the learning curve in terms of loss function or negative evidence lower bound.

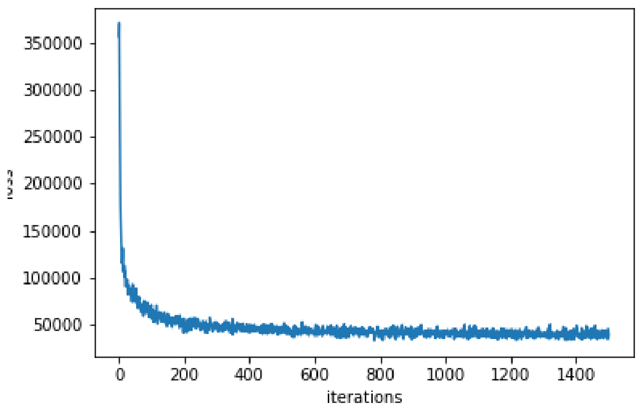


Figure 2: Learning curve of VAE

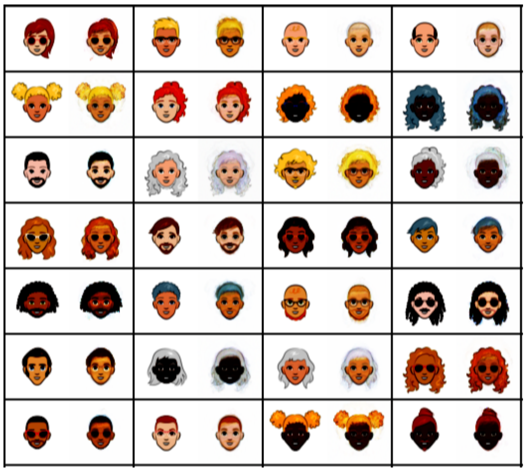


Figure 3: Reconstruction samples using VAE, left one in each pair is ground truth, right one is reconstructed image

ii. Show some examples reconstructed by your model.

iii. Sample the prior $p(\mathbf{z})$ to generate some examples when your model is well-trained with convergence.

Figure 4: Samples drawn from VAE

2. **Style Transfer**

In this exercise, you will implement a cycleGAN to transfer the American cartoon character into the Japanese animation style, and vice versa. There are 10000 images in both folder "cartoon" and folder "animation". CycleGAN paper can be downloaded here. The example pytorch codes are provided as cyclegan_train.py and cyclegan_test.py. You only need to fill up TODO parts we marked in the code.

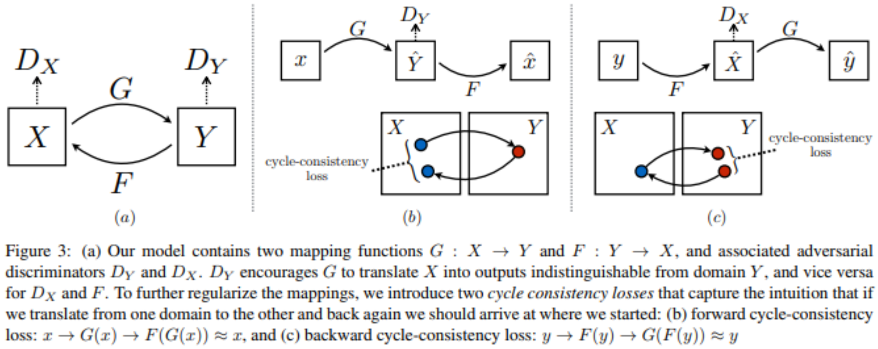


Figure 5: Structure of cycleGAN

i. Construct a cycleGAN with the loss function below. Plot the learning curve of both generators and discriminators. You can sum up the loss of two generators and plot in one curve.

$$
\begin{aligned}
&\mathcal{L}_{\mathrm{GAN}}(G, D_Y, X, Y) = \mathbb{E}_{y\sim p_{\mathrm{data}}(y)}\left[\log D_Y(y)\right] + \mathbb{E}_{x\sim p_{\mathrm{data}}(x)}\left[\log(1 - D_Y(G(x))\right] \\
&\mathcal{L}_{\mathrm{cyc}}(G, F) = \mathbb{E}_{x\sim p_{\mathrm{data}}(x)}\left[\|F(G(x)) - x\|_1\right] + \mathbb{E}_{y\sim p_{\mathrm{data}}(y)}\left[\|G(F(y)) - y\|_1\right] \\
&\mathcal{L}(G, F, D_X, D_Y) = \mathcal{L}_{\mathrm{GAN}}(G, D_Y, X, Y) + \mathcal{L}_{\mathrm{GAN}}(F, D_X, Y, X) + \lambda\mathcal{L}_{\mathrm{cyc}}(G, F) \\
&G^*, F^* = \arg\min_{G,F}\max_{D_x, D_Y}\mathcal{L}(G, F, D_X, D_Y)
\end{aligned}
$$

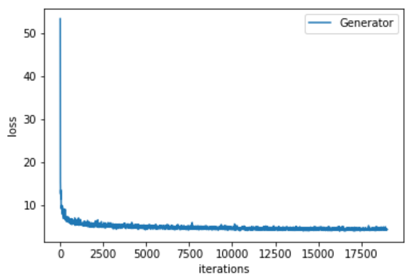


Figure 6: Loss curve for generators

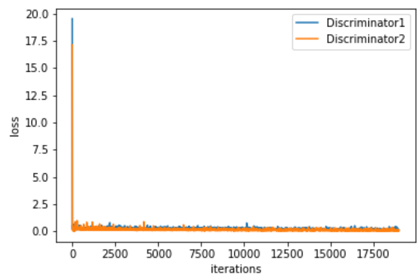


Figure 7: Loss curve for discriminators

ii. Please sample some cartoon images in animation style and animation images in cartoon style. Show your results and make some discussion in the report.

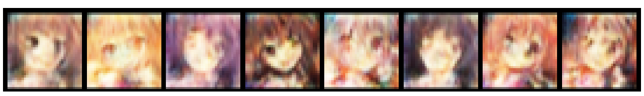


Figure 8: Samples of cartoon images in animation style

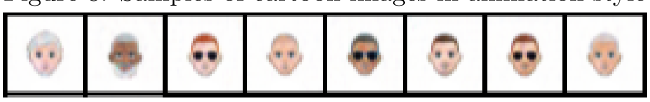


Figure 9: Samples of animation images in cartoon style

iii. Briefly describe what is mode collapse. According to (ii), is mode collapse issue serious in this task? Why?

Hints

a. Training procedure of GANs is unstable. When visualizing the loss curve, you can do moving average every N steps to smooth the curve and to observe the trend easily.

b. You can decide the number of deconvolution/convolution layers in generator/discriminator by yourself.

c. Optimizer such as Adam or RMSProp is recommended. If you use SGD, it may take more iterations to converge.

d. You may set different learning rate for G and D. Usually, learning rate of D could be smaller than G.

e. If you don't have much time or a powerful GPU, we suggest you to use smaller size of images (such as 32x32) and fewer epochs.

f. Batch normalization is useful for training neural network.