Deep Learning (Homework 3)

Due date : 06/08/2018

- 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 animation faces dataset. The dataset contains 51223 animation face image without labels parsed from the internet. All of the images are given with size of 96×96 as shown below. You will use this dataset to carry out the task of unsupervised image generation.

1. **Variational Autoencoder**

   This exercise will guide you to construct a Variational Autoencoder (VAE) for image reconstruction.

   $$\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})$$

   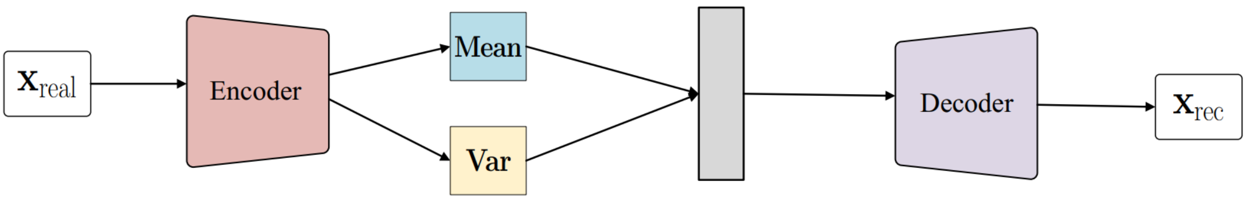

   Figure 1: Structure of VAE

   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. You need to design the network architecture and analyze the effect of different settings including dimension of latent z and minibatch size. Finally, plot learning curve (variational lower bound).

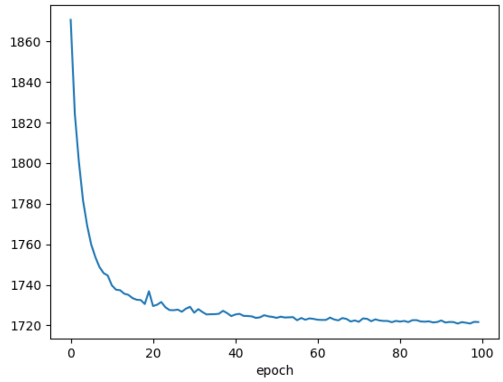

Figure 2: Learning curve of VAE

Figure 3: Reconstruction samples using VAE

Figure 4: Real samples in dataset

ii. Show some examples reconstructed by your model.

iii. Sample the prior $p(z)$ to generate some examples when your model is well-trained.

Figure 5: Samples drawn from VAE

2. **Generative Adversarial Network**

In this exercise, you will implement a Deep Convolutional Generative Network (DCGAN) to synthesis images by using the same animation face dataset. Set the number of height, width and channel according to your data.

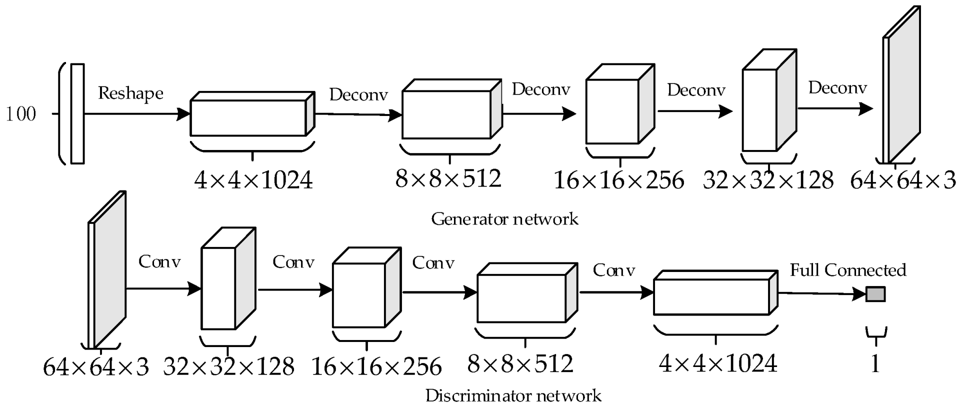

Figure 6: Structure of DCGAN

i. Data augmentation is used to enhance GAN training, describe how you preprocess the dataset and perform the data augmentation (such as resize, crop, rotate and flip). Show your improvement.

ii. Construct a DCGAN with vanilla GAN objective, plot the learning curve of both the generator and discriminator, and some samples generated by your model.

$$\max_D \mathcal{L}(D) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}} \log D(\boldsymbol{x}) + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}} \log(1 - D(G(\boldsymbol{z})))$$
$$\min_G \mathcal{L}(G) = \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}} \log(1 - D(G(\boldsymbol{z}))$$

Figure 7: Samples of DCGAN

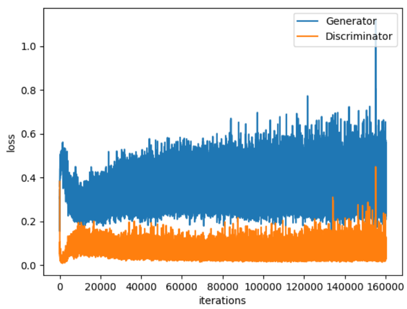

Figure 8: Learning curve of DCGAN

iii. Follow (ii), construct a DCGAN with Least Square GAN (LSGAN) objective , plot the learning curve of both the generator and discriminator, and some samples generated by your model. Least Squares GAN paper can be downloaded here.
(Hint: LSGAN removes the log operation in loss function and sigmoid operation in D)

$$\max_D \mathcal{L}(D) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}[(D(\boldsymbol{x}) - 1)^2] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}}[(D(G(\boldsymbol{z})))^2]$$
$$\min_G \mathcal{L}(G) = \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}}[(D(G(\boldsymbol{z}) - 1)^2]$$

Figure 9: Samples of LSGAN

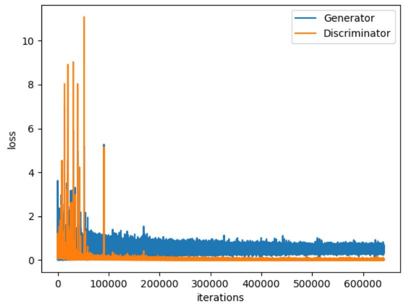

Figure 10: Learning curve of LSGAN

iv. According to (ii) and (iii), compare the results and do some discussion about log loss and L2 loss.

v. Compare the results between VAE and GAN and explain their differences.

Hints

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

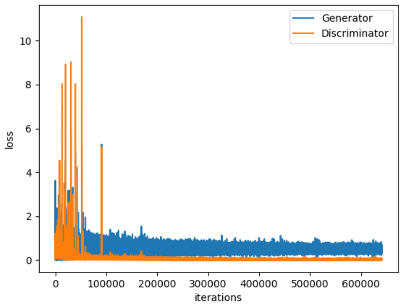

Figure 11: LSGAN loss (original)

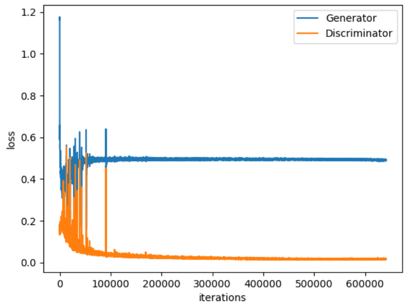

Figure 12: LSGAN loss (smoothed)

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 rates for G and D. Usually, learning rate of D could be smaller than G.

e. Suggest training the model for about 100 to 150 epochs (depends on when the model converges).

f. Batch normalization is useful for training neural network.