Deep Learning for Computer Vision (2018 Spring)

HW4

B03611026 郭冠軒

Problem 1 : VAE

1. Describe the architecture & implementation details of your model.

Encoder architecture:

Action	Details
Convolution + Batch Normalization	Filters = 64, Kernel size = 4, Strides = 2, Activation function = leaky ReLU
Convolution + Batch Normalization	Filters = 128, Kernel size = 4, Strides = 2, Activation function = leaky ReLU
Convolution + Batch Normalization	Filters = 256, Kernel size = 4, Strides = 2, Activation function = leaky ReLU
Convolution + Batch Normalization	Filters = 512, Kernel size = 4, Strides = 2, Activation function = leaky ReLU
Flatten	
Dense (obtain z mean)	Units = 1024
Dense (obtain z log variance)	Units = 1024
Random Normal (obtain epsilon)	
Add & Multiply (obtain z)	Dimension = 1024

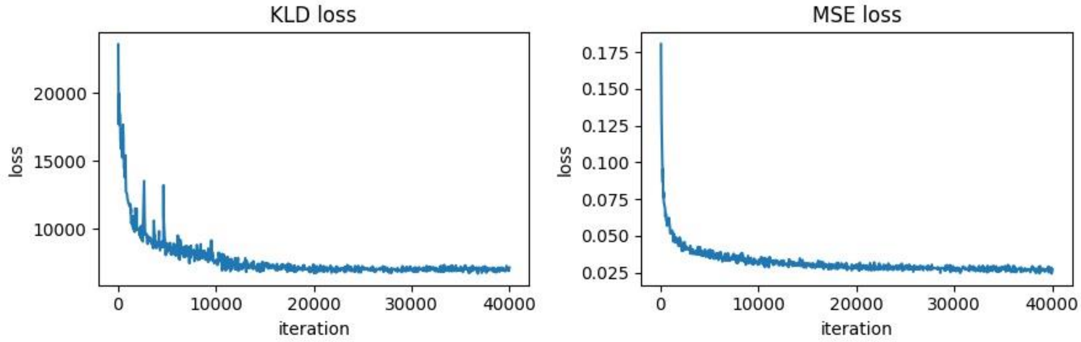
Decoder architecture:

Action	Details
Reshape	Shape = [-1, 4, 4, 64]
Deconvolution + Batch Normalization	Filters = 256, Kernel size = 4, Strides = 2, Activation function = leaky ReLU
Deconvolution + Batch Normalization	Filters = 128, Kernel size = 4, Strides = 2, Activation function = leaky ReLU
Deconvolution + Batch Normalization	Filters = 64, Kernel size = 4, Strides = 2, Activation function = leaky ReLU
Deconvolution	Filters = 3, Kernel size = 4, Strides = 2, Activation function = tanh

While training the VAE model, the loss I optimized is the summation of MSE loss and KL divergence loss, and the λ term I chose is 10^{-6}.

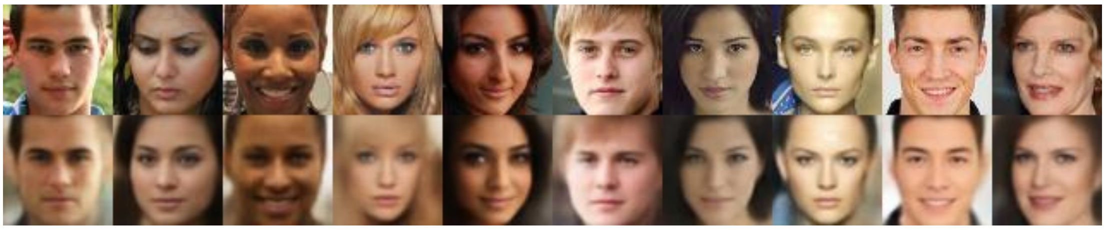
$$total\ loss = MSE\ loss + (\lambda \cdot KLD\ loss)$$

2. Plot the learning curve of your model.



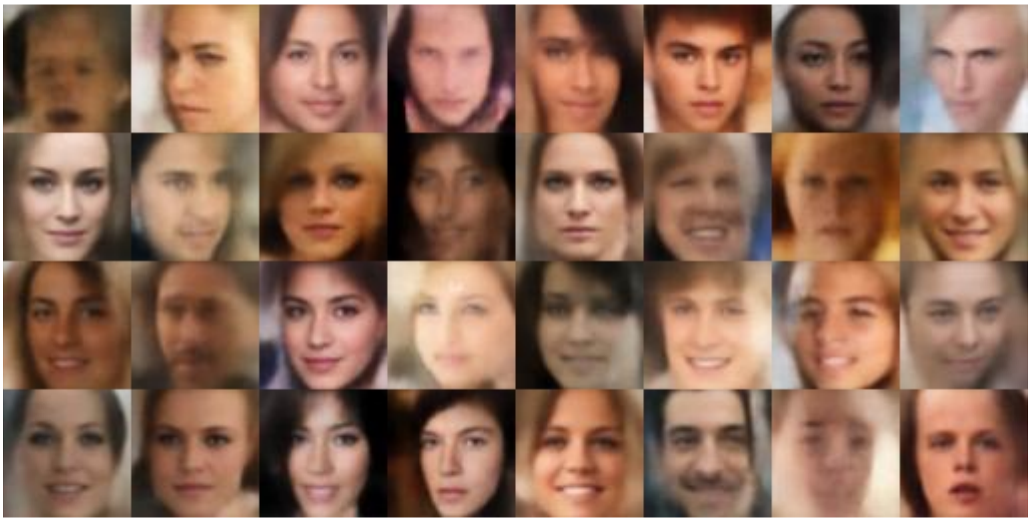
3. Plot 10 testing images and their reconstructed results of your model, and report your testing MSE of the entire testing set.

   The first row of the figure below shows the original images, and the second row indicates the reconstructed images.



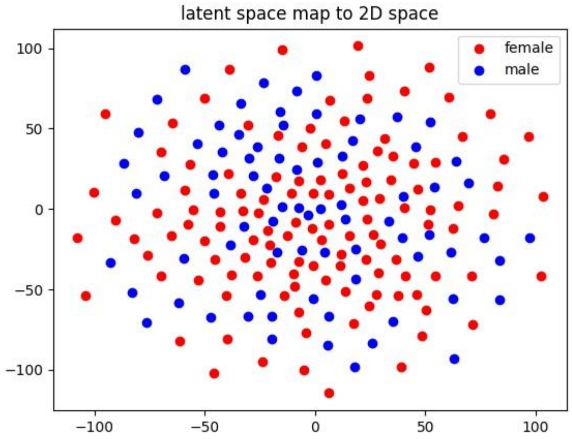
   The MSE of the entire testing set is <u>0.028471</u> (the images are scaled to the range between -1 and 1).

4. Plot 32 random generated images of your model.



5. Visualize the latent space by mapping test images to 2D space and color them with respect to an attribute of your choice.

   The attribute I chose is “male”. The blue points indicate the “male” attribute, and the red points indicate the “female” attribute.



6. Discuss what you’ve observed and learned from implementing VAE.

   The value of λ term affect the quality of both reconstructed images and randomly generated images.

   If the value of λ is too large, the generated images seem to be blurry. Conversely, if the value of λ is too small, the generated images will be much clearer, but the images generated from the randomly sampled Gaussian distribution vectors won’t look like normal human faces.

Problem 2 : GAN

1. Describe the architecture & implementation details of your model.

   Generator architecture:

Action	Details
Dense and Reshape	Shape = [-1, 4, 4, 1024]
Deconvolution + Batch Normalization	Filters = 512, Kernel size = 5, Strides = 2, Activation function = ReLU
Deconvolution + Batch Normalization	Filters = 256, Kernel size = 5, Strides = 2, Activation function = ReLU
Deconvolution + Batch Normalization	Filters = 128, Kernel size = 5, Strides = 2, Activation function = ReLU
Deconvolution + Batch Normalization	Filters = 3, Kernel size = 5, Strides = 2, Activation function = tanh

Discriminator architecture:

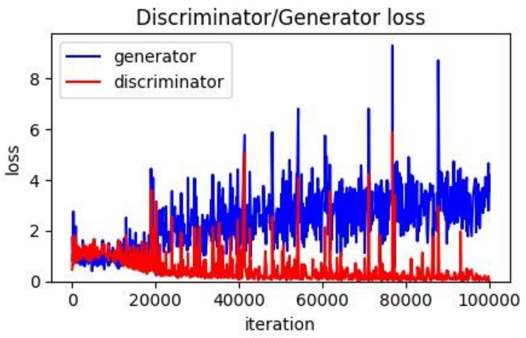
Action	Details
Convolution + Batch Normalization	Filters = 64, Kernel size = 5, Strides = 2, Activation function = leaky ReLU
Convolution + Batch Normalization	Filters = 128, Kernel size = 5, Strides = 2, Activation function = leaky ReLU
Convolution + Batch Normalization	Filters = 256, Kernel size = 5, Strides = 2, Activation function = leaky ReLU
Convolution + Batch Normalization	Filters = 512, Kernel size = 5, Strides = 2, Activation function = leaky ReLU
Flatten	
Dense	Units = 1, Activation function = leaky ReLU

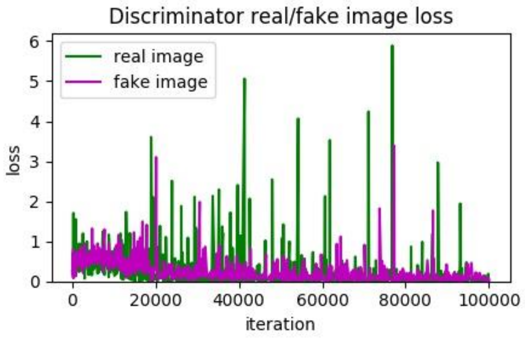
Optimizers:

Generator - Adam (learning rate = 2e-4, beta1 = 0.5)
Discriminator - Adam (learning rate = 2e-4, beta1 = 0.5)

2. Plot the learning curve of your model and briefly explain what you think it represents.

Here are the learning curves during the training process.

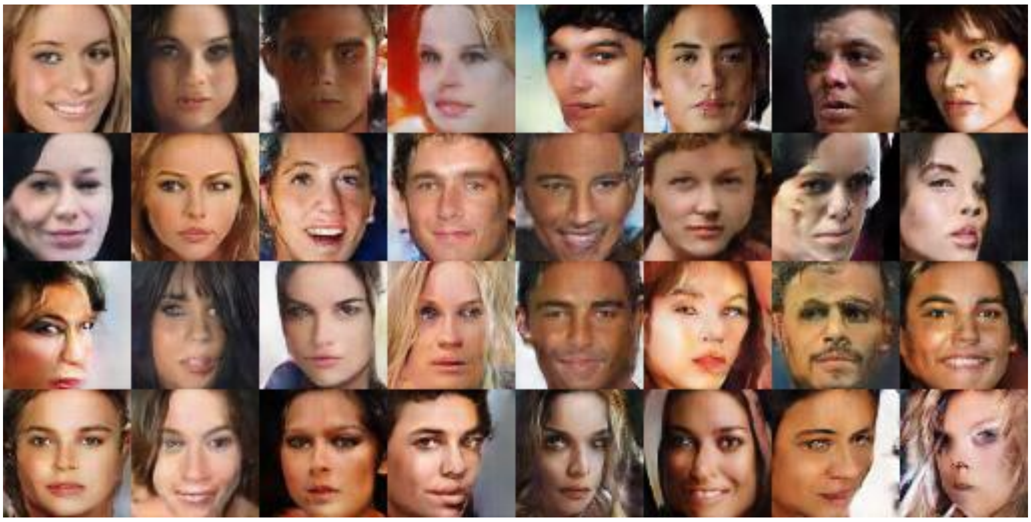




I found that the loss difference between the generator and the discriminator had become greater and greater while training, and the quality of images generated from the generator did not either rise or drop much. That means the discriminator were too powerful after training for a few iterations, and it was hard for the generator to improve itself.

Also, I recorded the discriminator's losses of real and fake images. Comparing to the losses of real images, the losses of fake images are lower in average. Perhaps we may say that it is easier for the discriminator to tell the fake images generated from the generator than the real images.

3. Plot 32 random generated images of your model.



4. Discuss what you’ve observed and learned from implementing GAN.

   The model I implemented on this task is DCGAN, which uses convolutional layers in its generator(G) and discriminator(D).

   GAN is a minimax game between G and D. If the learning ability of G/D is much stronger than D/G, the whole network will be unbalanced, and it may be hard to improve each other in the training process. As a consequence, I built G and D with the same amount of layers, and the structure of the model is nearly symmetric.

   Also, I updated G and D separately in the training process, since I discovered that they learned much slower if I jointly updated them.

5. Compare the difference between images generated by VAE and GAN, discuss what you’ve observed.

   VAE learns to generate images via converting a Gaussian distribution vector into human faces. It optimizes itself by minimizing MSE, so the generated images may be blurry.

   GAN learns to generate images through the competition between G and D. The generator’s goal is to generate images that can deceive the discriminator, and there may be some distortion or partial face with awkward colors appear on the generated images. Also, the quality of the images is affected by the ability of the discriminator.

Problem 3 : ACGAN

1. Describe the architecture & implementation details of your model.

Generator architecture:

Action	Details
Concatenate noise and label	Shape = [-1, 100 + 1]
Dense and Reshape	Shape = [-1, 4, 4, 1024]
Deconvolution + Batch Normalization	Filters = 512, Kernel size = 5, Strides = 2, Activation function = ReLU
Deconvolution + Batch Normalization	Filters = 256, Kernel size = 5, Strides = 2, Activation function = ReLU
Deconvolution + Batch Normalization	Filters = 128, Kernel size = 5, Strides = 2, Activation function = ReLU
Deconvolution + Batch Normalization	Filters = 3, Kernel size = 5, Strides = 2, Activation function = tanh

Discriminator architecture:

Action	Details
Convolution + Batch Normalization	Filters = 64, Kernel size = 5, Strides = 2, Activation function = leaky ReLU
Convolution + Batch Normalization	Filters = 128, Kernel size = 5, Strides = 2, Activation function = leaky ReLU
Convolution + Batch Normalization	Filters = 256, Kernel size = 5, Strides = 2, Activation function = leaky ReLU
Convolution + Batch Normalization	Filters = 512, Kernel size = 5, Strides = 2, Activation function = leaky ReLU
Flatten	
Dense (from Flatten) Real / Fake	Units = 1, Activation function = leaky ReLU
Dense (from Flatten) With / Without label attribute	Units = 1, Activation function = leaky ReLU

Optimizers:

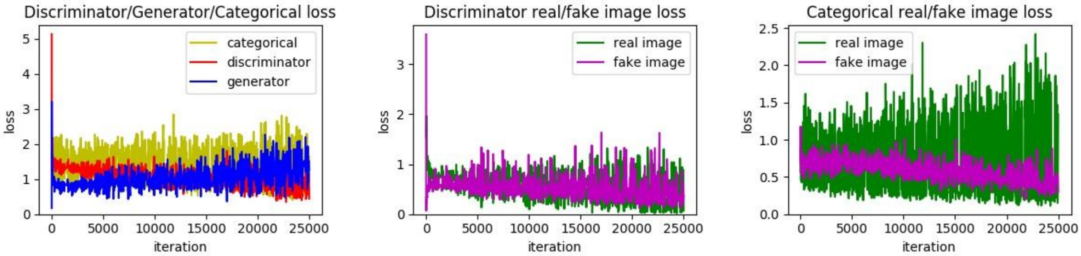
Generator - Adam (learning rate = 8e-4, beta1 = 0.5, beta2 = 0.9)
Discriminator - Adam (learning rate = 2e-4, beta1 = 0.5, beta2 = 0.9)

The attribute to disentangle:

Smiling

2. Plot the learning curve of your model and briefly explain what you think it represents.

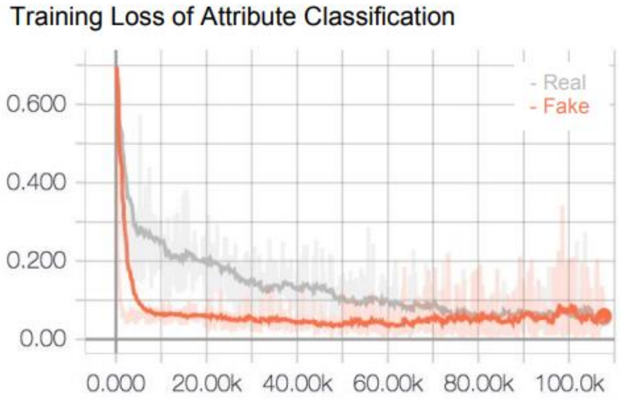
Here are three kinds of losses in the task, including generator losses, discriminator losses, and categorical losses.



From the figure on the left, we may find that the generator loss had been gradually increasing during the training process, and the discriminator loss had been decreasing. This situation is similar with the case I mentioned in problem 2-2, also caused by the unbalance ability between G and D.

Let’s take a look at the other two figures. If we focus on the loss difference between real and fake images, the two discriminator losses are very close, but the two categorical losses are not close at all, and the losses of real images fluctuate severely.

My interpretation is, the discriminator did not learn well how to classify the label attribute of images in the beginning of training, and what it had learned is about the simple patterns which are also learned by the generator, so the categorical losses of fake images are lower, and the losses of real images are not stable. But as the learning process going on, both of the two losses should become smaller, just like the figure that TA posted on page 27 of the homework instruction slides (the figure below).



3. Plot 10 pairs of generated images of your model, each pair generated from the same random vector input but with different attribute. This is to demonstrate your model's ability to disentangle feature of interest.

The first row are the images with the "smiling" attribute, the second row are the images without the attribute.