National Chiao Tung University

Spring 2019

Deep Learning

Instructor: Jen-Tsung Chien

Deep Learning HW3 Report

Alfons Hwu

Student ID: 0416324

alfons.cs04@g2.nctu.edu.tw

Dept. of Computer Science

Composed with LaTeXon Overleaf

Deep Learning HW3 Report

June 5, 2019

1 Self-designed VAE for image reconstruction and generating(unsupervised learning)

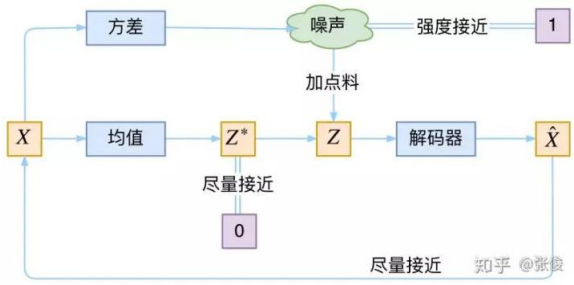
1.1 Preprocessing the images

In the preprocessing part, I merely resize the image to 64 x 64 and without normalization.
If normalization is used, the reconstructed / randomly generated image will look very dark and black.
CNN is used in this problem since it **fits better for image processing** while DNN falls short in this part because it processes 3 color channels at the same time, causing the result image somehow looked grey(The value of R G B are similar).
The CNN layers are used in the feature extraction and reconstruction part as the following code shows.

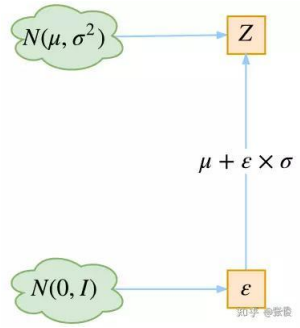
```
def __init__(self, image_channels = 3, h_dim = 1024, z_dim = 32):
        super(VAE, self).__init__()
        self.encoder = nn.Sequential(
            nn.Conv2d(image_channels, 32, kernel_size = 4, stride = 2),
            nn.ReLU(),
            nn.Conv2d(32, 64, kernel_size = 4, stride = 2),
            nn.ReLU(),
            nn.Conv2d(64, 128, kernel_size = 4, stride = 2),
            nn.ReLU(),
            nn.Conv2d(128, 256, kernel_size = 4, stride = 2),
            nn.ReLU(),
            flatten()
        )
        self.fc1 = nn.Linear(h_dim, z_dim)
```

```
        self.fc2 = nn.Linear(h_dim, z_dim)
        self.fc3 = nn.Linear(z_dim, h_dim)
        # The code of decoder part is just deconvolution, i.e. nn.ConvTranspose2d
        )
```

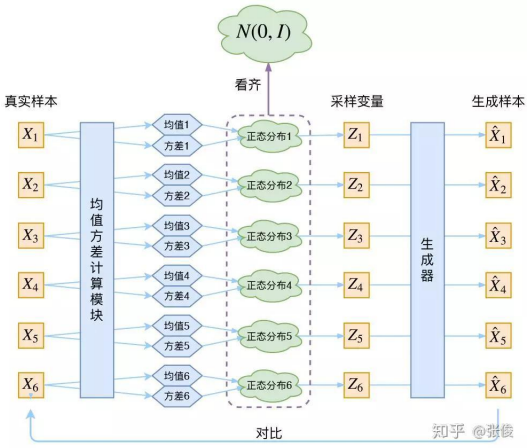
Between the feature extraction and image reconstruction lies the reparameterization part, that is by multiplying σ and shifting μ, making $N\ (0,1)$ to approximate $N\ (\mu,\sigma^2)$



(a) VAE ovaerall architecture



(b) Reparameterization to approximate $N\ (0,1)$

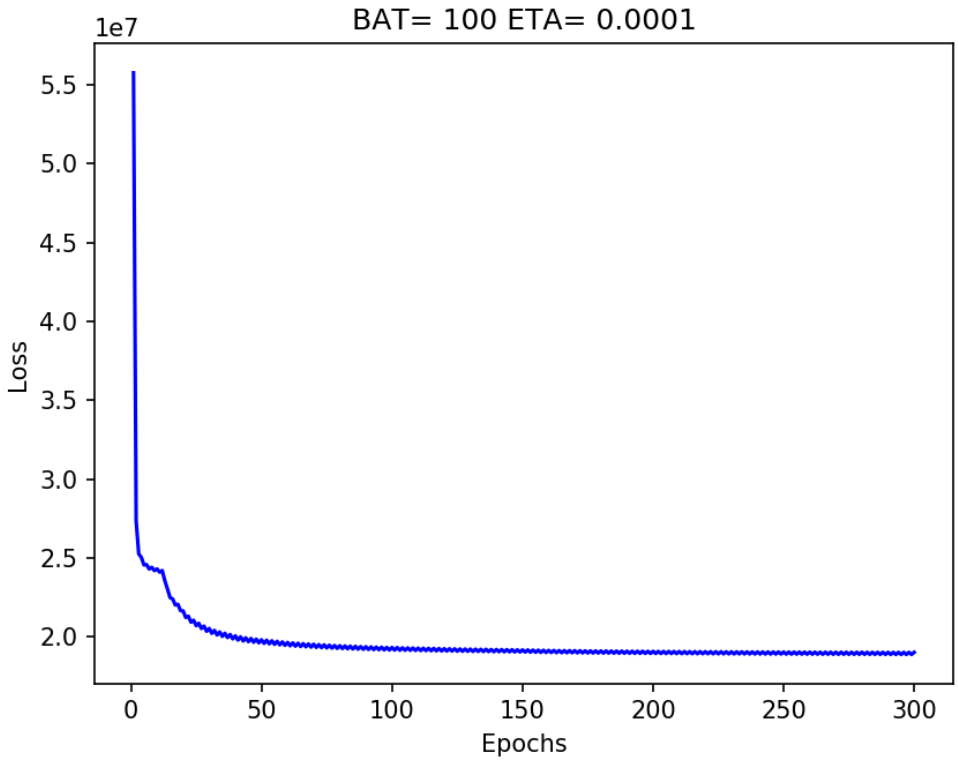


(c) Mathematical explanation

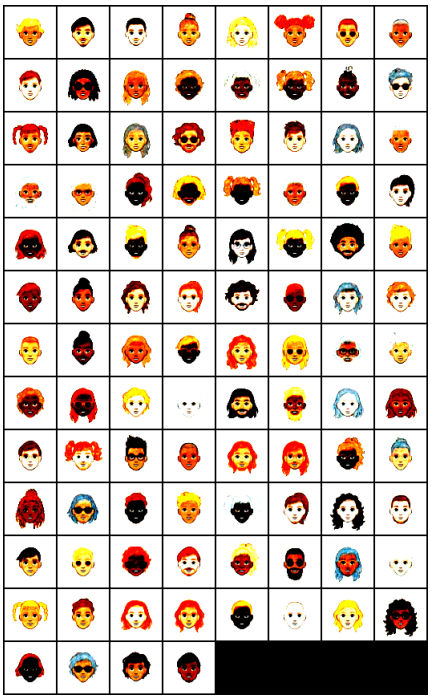
Figure 1: VAE explanation

Image source credit to `https://zhuanlan.zhihu.com/p/34998569`

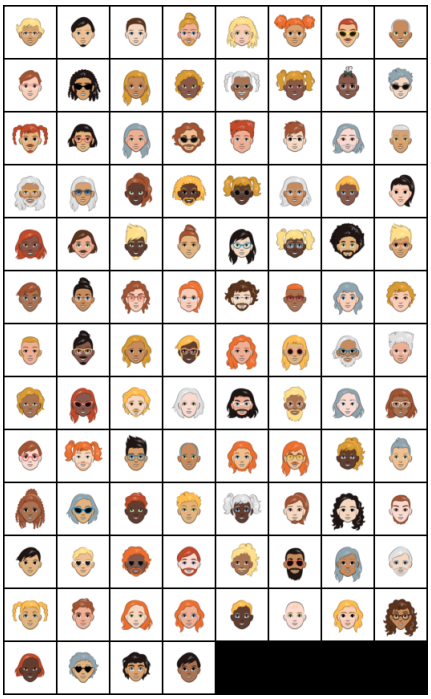
1.2 Learning curve and reconstructed images



(a) Learning curve



(b) Reconstructed image

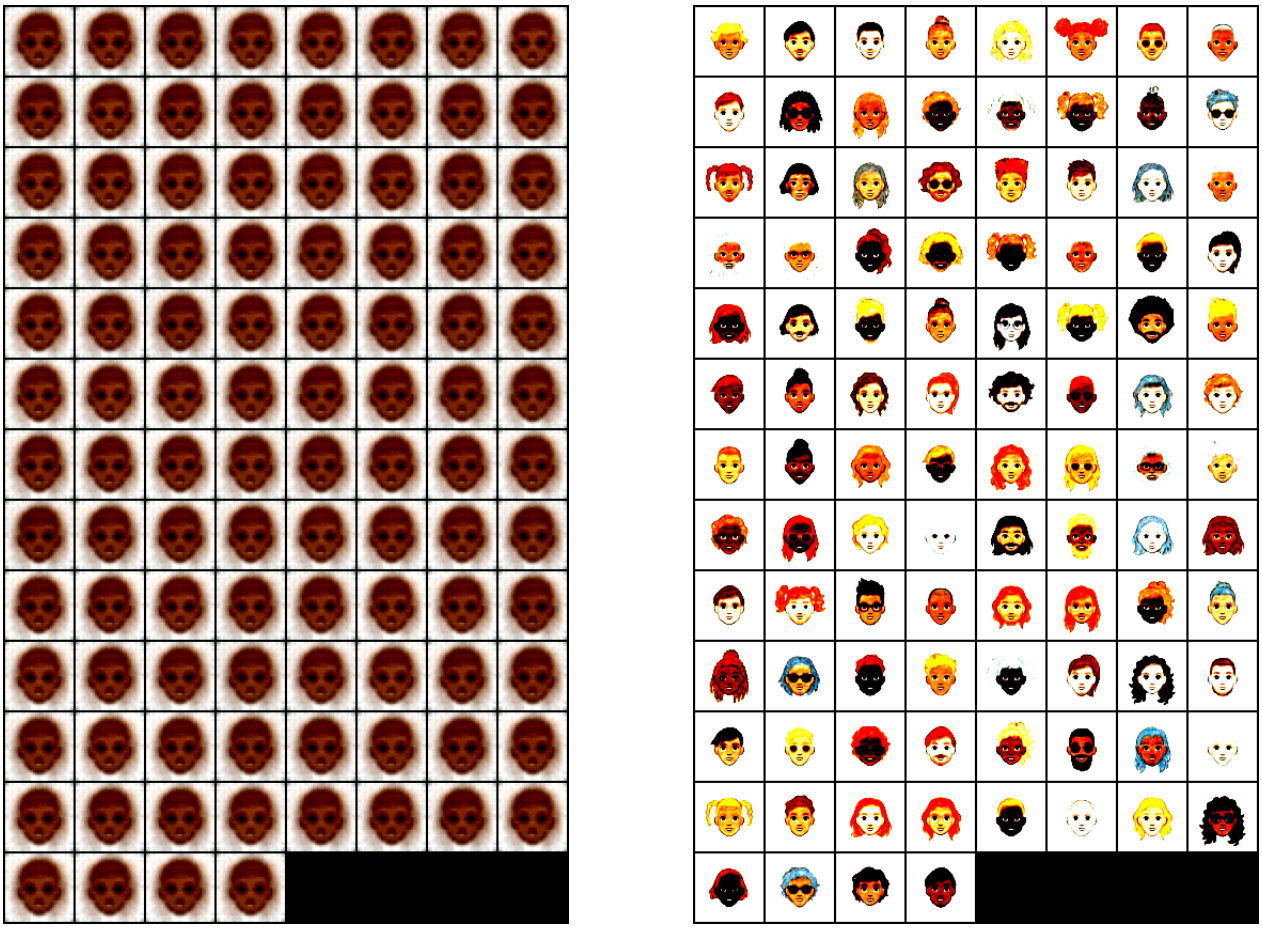


(c) Ground truth

Figure 2: VAE result evaluation

KL annealing is used in this model to improve learning different characteristics from different faces, since without it, KL Divergence and Loss will be quite small, and with small error value, NN will be hard to implement self-learning, causing the whole reconstructed image to be all the same (bulrry, averaging all the characteristics of all images).

KL annealing reference to `http://www.sohu.com/a/216987798_297288`

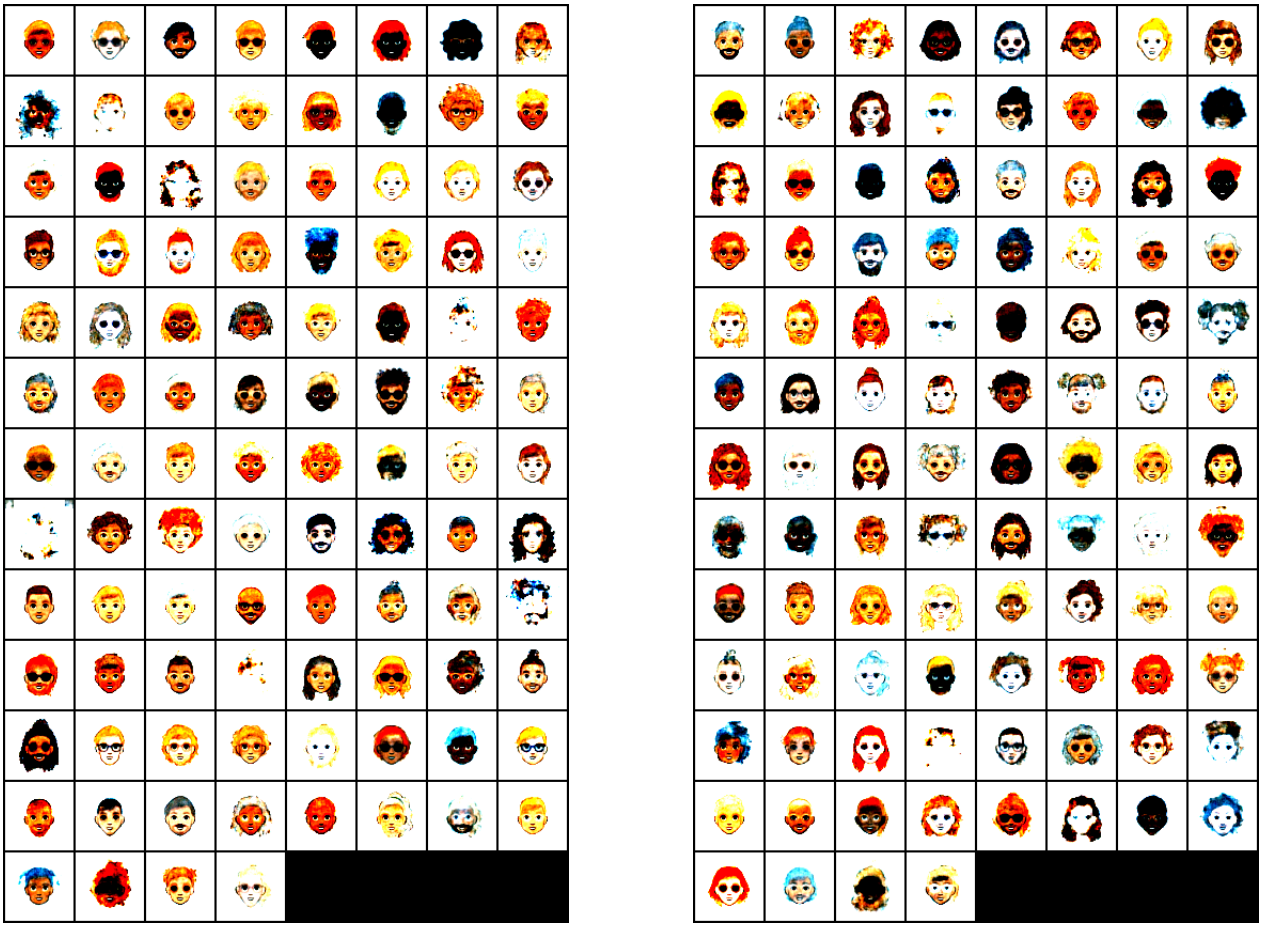


(a) Without KL annealing

(b) With KL annealing

Figure 3: Comparison

1.3 Randomly generated images



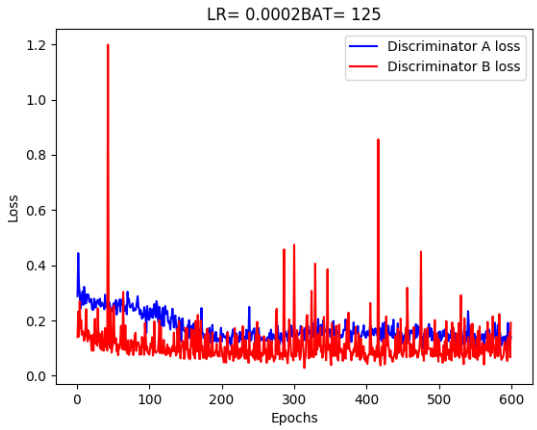
(a) Generated 1

(b) Generated 2

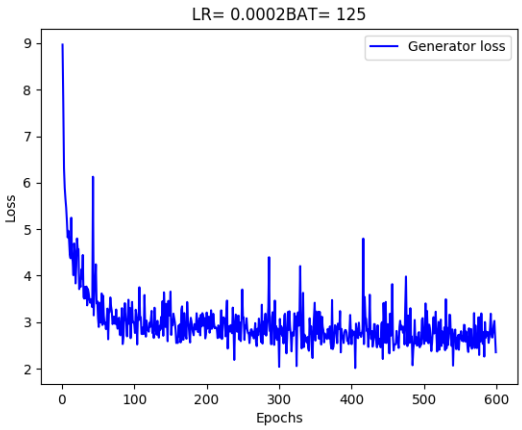
Figure 4: Generated images

2 Self-designed CGAN for style transformation(unsupervised learning)

2.1 Loss curves of discriminator and generator



(a) Loss curve of discriminator



(b) Loss curve of generator

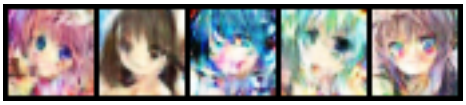
Figure 5: Loss curves

2.2 Results of style transferred images

(a) Real cartoon 1

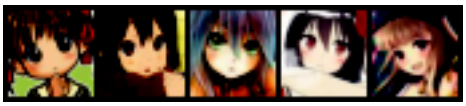
(b) Real cartoon 2

(c) Fake anime 1(From real cartoon 1)



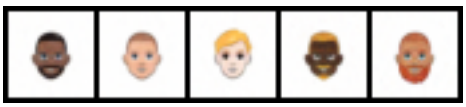
(d) Fake anime 2(From real cartoon 2)

Figure 6: Transfer from cartoon(real cartoon in ground truth) to anime(fake anime generated by GAN)



(a) Real anime 1

(b) Real anime 2



(c) Fake cartoon 1(From real anime 1)

(d) Fake cartoon 2(From real anime 2)

Figure 7: Transfer from anime(real anime in ground truth) to cartoon(fake cartoon generated by GAN)

2.3 Mode collapse discussion

Mode collapse is a term that describes the generator **generates a limited diversity of samples, or even the same sample, regardless of the input.**

The problem of mode collapse is a bit serious in this task as **Figure 7** shows (The appearance of generated images do not differ alot from on to the other, especially in the color of randomly-generated cartoon style).

Reason: The discriminator ends up not really forcing more diversity in the generator, so much as simply pushing the partially collapsed generator to a different part of output space - if it assigns the collapse point a low probability, the generator will simply move its collapsed distribution to focus on a new output point. And finally, in the case where the generator has actually collapsed to a single point, it can't get out.

Geometrically speaking, the gradients for backpropogation, in the multidimensional vector space, do not lies orthogonal to each other anymore, some lies parallel while others form a linear combination of the others.

In this phenomenon, the model gets stuck in certain dimension (dimension deficient), hence in that dimension(i.e. feature), the PDF(probability density function) generated by the model has some peak in certain points, causing the lack of diversity in generated images.

The architecture of CGAN can be understand through: `https://zhuanlan.zhihu.com/p/26995910`

The understanding of mode collapse and how to amend can be referenced in the following.

`https://zhuanlan.zhihu.com/p/36410443` and `https://www.quora.com/What-causes-mode-collapse-in-GANs`