Anime Generation by Generative Adversarial Networks

孫凡耕 羅啟心 許晉嘉 郭子生

National Taiwan University, Department of Electrical Engineering

Introduction

In MLDS Spring 2017 HW3, we dedicate to generate anime images from given input text by the well known text-to-image synthesis method : Conditional Generative Adversarial Network (Conditional GAN). Our work include two stages, in the first stage we train the Conditional GAN with only hair color and eyes color features. Which means, the input text are all in the format "__ hair __ eyes". For example, green hair gray eyes. And a series of five corresponding anime images is then produced. In the second stage, we proceed further by collecting more features from the dataset, and by feeding more complicated input sentence, desired feature is then learned by Conditional GAN. We are then able to synthesis anime character with glasses or no glasses, along with different hair styles. We had also learned the most commonly appeared 200 features from the dataset and generated relevant images, as we display below.

Method

We generate images by using the state-of-the-art text-to-image-synthesis method : Conditional Generative Adversarial Network (Conditional GAN), with GAN structing being DCGAN. This method is first proposed on CVPR, 2015. First, we introduce what is a generative adversarial network (GAN), and further explain how Conditional GAN works, finally to our method.

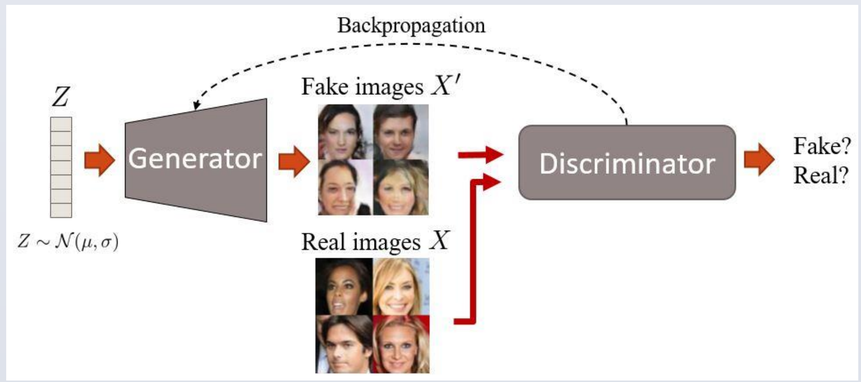
GAN

Originally proposed by Ian Goodfellow — have two networks, a generator and a discriminator. They are both trained at the same time and compete again each other in a minimax game. The generator is trained to fool the discriminator creating realistic images, and the discriminator is trained not to be fooled by the generator.

And this is how GAN works : Input noise z is sampled from a probability distribution (Gaussian distribution for example). Input noise z is then fed into generator G (usually a multi layer perceptron), which produces an image. The synthesized image is now present to the discriminator D for it to output a probability p : the probability that the image comes from the original dataset rather than G .

GAN has training objective function :

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$

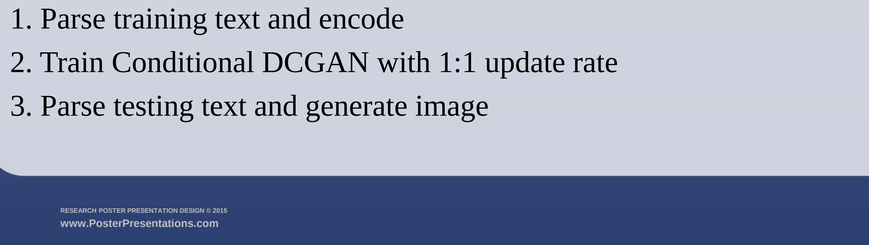


Conditional GAN

For conditional GAN, the main structure are identical but differ in input vector : the random noise vector z is now concatenated by condition vector y, so the objective function now is :

$$\min_G \max_D V(D,G) = \mathop{\mathbb{E}}_{x\sim p_{data}(x)}[\log D(x,y)] + \mathop{\mathbb{E}}_{z\sim p_z(z)}[\log(1 - D(G(z,y),y))]$$

In this figure, we can cast a glance on the structure of Conditional GAN :



DCGAN

DCGAN is a GAN that has certain architectural constraint and performs well on generating images. The input noise z is first projected to a small spatial extent convolutional representation with many feature maps, followed by four fractionally-strided convolutions, producing image G(z). The image G(z) then transform through four convolutional networks and two linear layers of D, 1 dimensional probability is finally produced . (We omit details such as activation functions and how batch norm is added here.)

Our Method

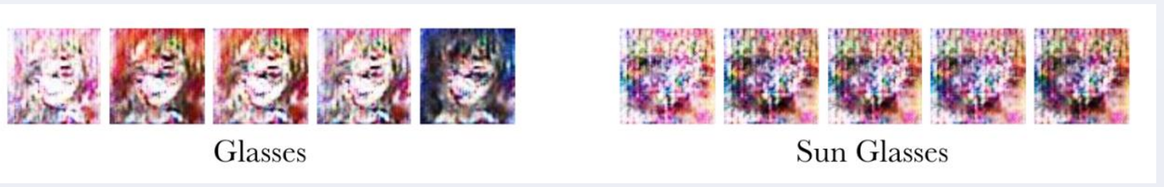
We train text to anime images by

1. Parse training text and encode
2. Train Conditional DCGAN with 1:1 update rate
3. Parse testing text and generate image

Experiments

1. Glasses and Sun Glasses

In the first experiment we manage to generate images with glasses and sun glasses. We first parse the input text and encode by one-hot encoding. As shown in the figure, glasses can be easily recognized in the generated image, while the presence of sun glasses is less evident. We dug through the training data and it turns out that there are 700 images with glasses. On the contrary, there are much less images with sun glasses. Merely 10 of them are with sun glasses in the entire dataset.



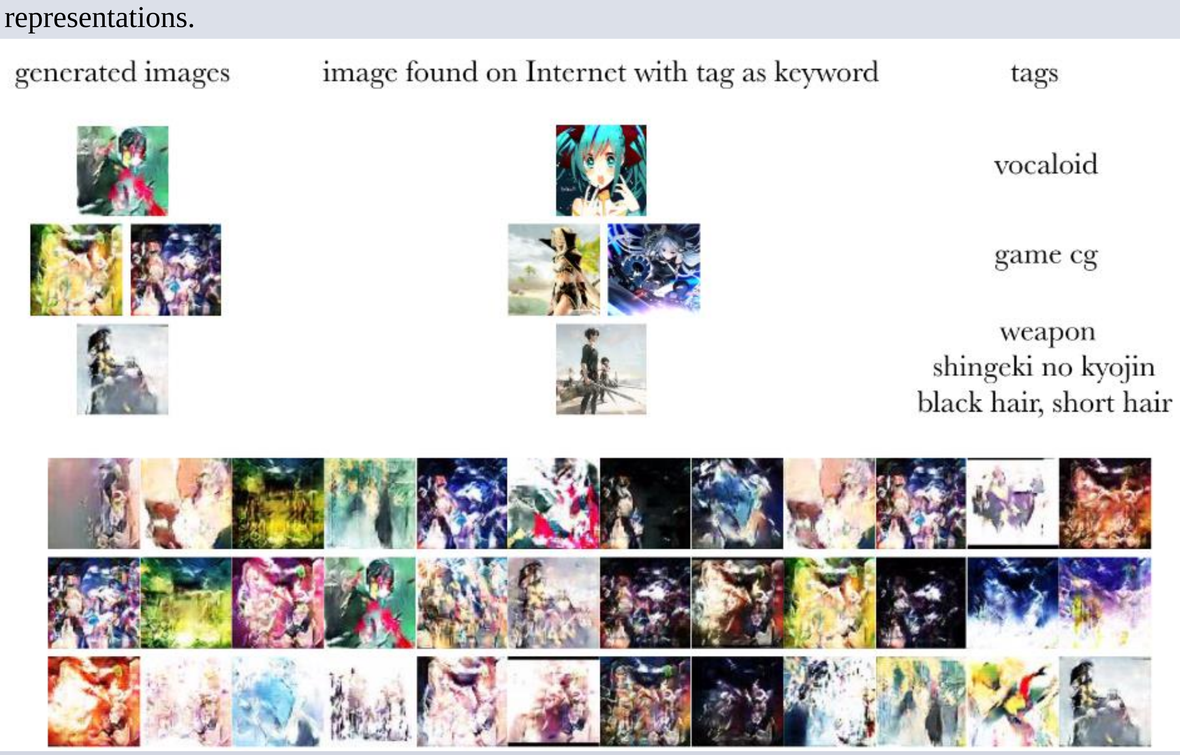
2. Hair Style

After training anime images with glasses, we carry on by training images with various hair styles. The four hair styles we manage to train are all popular hair styles in anime images : "long hair", "short hair", "twin tails" and "pony tails". We train them by one-hot encoding. Although the tags "long hair" and "short hair" ranks first and third in all tags respectively, the generated images still perform weakly on these tag . We believe that such phenomenon emerges because each image is almost occupied by the face of anime characters, little space is left for display of hair styles . No wonder the DCGAN network cannot learn hair style well .

3. Full Images

In the last experiment, we collect the most frequently appeared tags and generate images with these tags . Different from the encoding method in experiment 1 and 2 , we use skip thought vector to encode the input text in this experiment . For example , we have tags A , B , C for one image , and only A and B are in the top 200 tags list . The way we encode the text is by embedding tags A and B into the sentence " The girl has A and B " , and encode by Skip-Thought Vectors . We list some generated images produced by given tags , with the image searched on the Internet by tags aside . We observe that pairs of images roughly match . More images generated by random tags in the top 200 tags list is displayed as well .

* Skip-Thought Vectors is an approach for unsupervised learning of a generic distributed sentence encoder . Using the continuity of text from books, an encoder-decoder model is trained to for the purpose of reconstructing the surrounding sentences of an encoded passage. In such way sentences that share semantic and syntactic properties are thus mapped to similar vector representations.



References

[1] Radford, A., Metz, L., and Chintala, S. **Unsupervised representation learning with deep convolutional generative adversarial networks.** 2016.

[2] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. **Generative adversarial text to image synthesis.** 2016.

[3] R. Kiros, Y. Zhu, R. Salakhutdinov, R. S. Zemel, A. Torralba, R. Urtasun, S. Fidler. **Skip-Thought Vectors.** 2015.

[4] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D, Warde-Farley, S. Ozair, A.Courville, Y. Bengio. **Generative Adversarial Networks.** 2014

Acknowledgement

We would like to thank Prof. Hung Yi Lee for arranging such enriched lectures and courses.