VE 445 2019 Spring Lab 3 Generative Adversarial Network

I. Introduction.

In the previous labs, you have implemented and built some discriminant models to solve classification problems. In this lab, you will enjoy implementing the generative model by yourself. Put up by Ian J. Goodfellow in 2014, Generative Adversarial Networks have been wildly used in image generation and semantic segmentation. Through this lab, you will implement one of the most efficient and popular network model and learn the simple idea behind big data and complex network structures.

II. Generative Adversarial Network

Compared with other networks that contain single topological structure, GAN has two individual network structures: discriminant model and generative network. The function of discriminant network $\mathbf{D}$, or discriminator, is to identify whether a given sample is a true sample or a fake one produced by generative model $\mathbf{G}$. While the function for generator $\mathbf{G}$ is to generate samples by random noise.

In the training procedure, the generator should deceive the discriminator as much as possible by generating fake samples. The abstract process of GAN training is shown below:

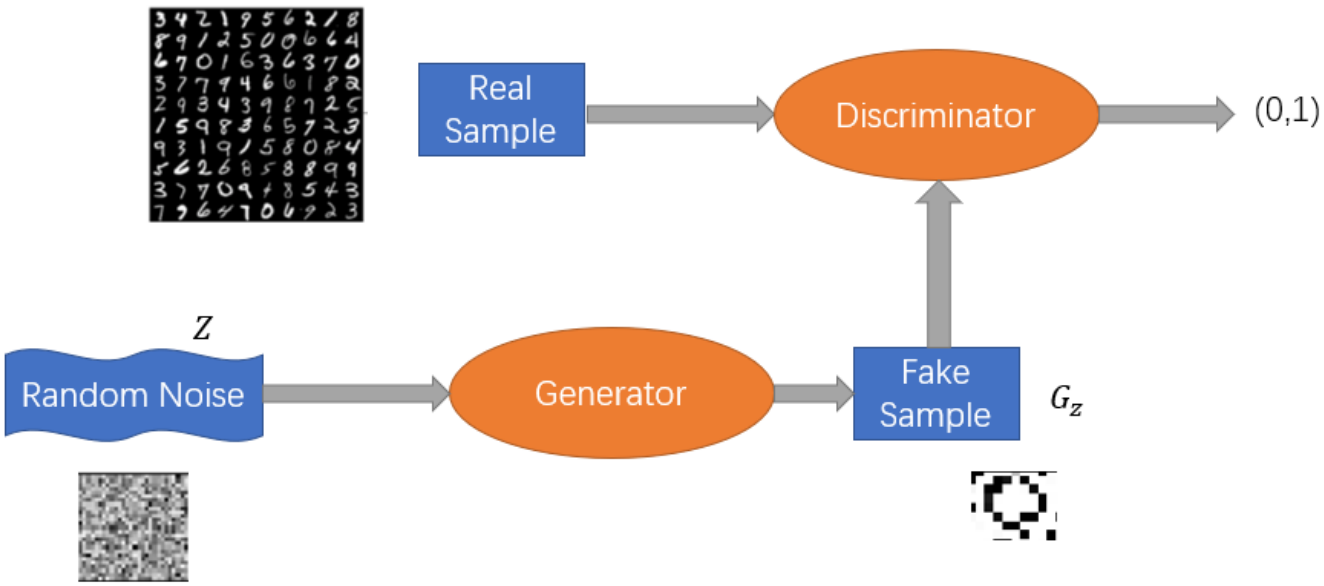

Fig 1. The process of GAN training

The output of the discriminator is a value between (0,1), which represents for the probability whether the given sample is real. The random noise is an random vector served as arbitrary input for generator. The loss function should be calculated individually for both the discriminator and the generator.

$$\min L_D = \min_D -\sum_{i=1}^{m_r} \log D(\mathbf{x}_i) - \sum_{j=1}^{m_f} \log D(G(\mathbf{z}))$$

$$\min L_G = \min_G -\sum_{i=1}^{m_f} \log D(G(\mathbf{z}))$$

For each step of the training, the two loss functions should be optimized individually.

III. Implementation

1. Datasets

   The datasets used in this lab are still the MNIST and CIFAR_10 used in lab 2. The MNIST image generator is the basic task, you could get full marks by only implement it. The CIFAR_10 model is not as easy as the MNIST one and it is optional. You will obtained 2% of the total course grades out of this lab if you implement the CIFAR_10 generator. For CIFAR_10, you could choose only one type of images to train your GAN.

2. Model training

   This lab require you to implement a image generator. For your implementation, tensorflow and keras are both recommended. You should define your generator and discriminator first and train them with the loss functions above in the same epoch. Notice that the stability of GAN models is fragile due to the network structure and parameters, therefore, you could refer to some stabilized model from open sources. The generator model should be saved as the same requirement in lab 2 with name "generator_mnist" and "generator_cifar".

3. Specifics

   The output image of the generator should be sized by 28 x 28 just as the size of data in MNIST. You are supposed to show your generated images by different epochs in your lab report.

IV. Grading and Submission

1. Grading:

   [20%] Coding style and correctness.

   [50%] Identification accuracy with generated image by your

   [30%] Your generated image performance and analysis in your lab report.

   [2% of total grading] Implementation of the CIFAR_10 generation model.

2. Submission:

   The submission should be a zip file only, which contains your saved model, python source code file and your lab report. The assignment is due by May 1st 23:59.