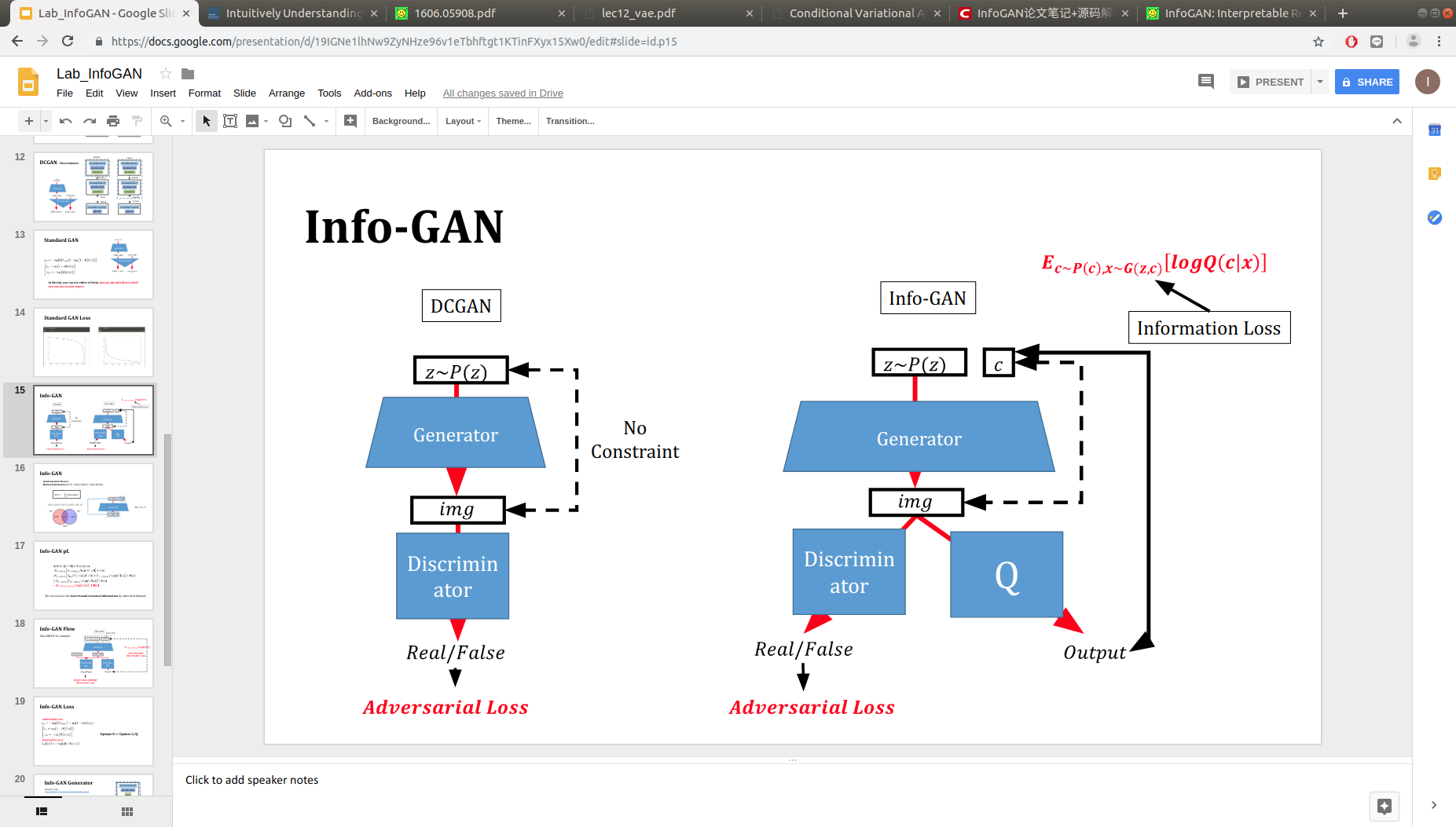
Lab4: InfoGAN

Yu-Hsiang Kao

# Introduction

Generative Adversarial Networks (GANs) are the biggest breakthrough in ML field. Unlike the other generative networks (e.g. VAE) that try to find local minimums via optimization, GANs set up a game between generators and discriminators and try to find Nash equilibrium between them.

However, due to the unconstrained nature of GAN, the input noise vector often fails to match with semantical interpretations. InfoGAN tries to learn meaningful interpretations by maximizing the mutual information between a fixed subset of the GAN’s noise variable and the observations.



# Experiment Setups

## InfoGAN Implementation

The InfoGAN used in the experiment was trained using the following hyperparameters.

|  |  |
| --- | --- |
| Batch size | 64 |
| Learning rate for the discriminator | 2e-4 |
| Learning rate for the generator and Q | 1e-3 |
| c\_size (size of meaningful codes) | 10 |
| Total epochs | 80 |
| Optimizer | Adam |

### Adversarial loss

The adversarial loss is a binary cross-entropy

Ld = -log(D(Ireal)) – log(1-D(G(z)))

which encourages the discriminator to output 1 when fed to a real sample and output 0 when fed to a generated sample.

### Maximizing mutual information

In information theory, mutual information between X and Y , I(X; Y ), measures the “amount of information” learned from knowledge of random variable Y about the other random variable X. The mutual information can be expressed as the difference of two entropy terms: I(X; Y ) = H(X) − H(X|Y ) = H(Y ) − H(Y |X) (2) This definition has an intuitive interpretation: I(X; Y ) is the reduction of uncertainty in X when Y is observed. If X and Y are independent, then I(X; Y ) = 0, because knowing one variable reveals nothing about the other; by contrast, if X and Y are related by a deterministic, invertible function, then maximal mutual information is attained. This interpretation makes it easy to formulate a cost: given any x ∼ PG(x), we want PG(c|x) to have a small entropy. In other words, the information in the latent code c should not be lost in the generation process. Similar mutual information inspired objectives have been considered before in the context of clustering [26–28]. Therefore, we propose to solve the following information-regularized minimax game: min G max D VI (D, G) = V (D, G) − λI(c; G(z, c))

## Choice of generator loss

In this experiment, there are two versions of generator loss to choose from

a) Lg = log(1-D(G(z)))

b) Lg = -log(D(G(z)))

At first glance, a) and b) are both valid as they both penalize generator when being rejected by the discriminator. But if the discriminator becomes to good and reject all the generated samples and D(G(z)) = 0, the two versions of loss become

a) Lg = log(1-0) = log1 = 0

b) Lg = log(0) = -Inf

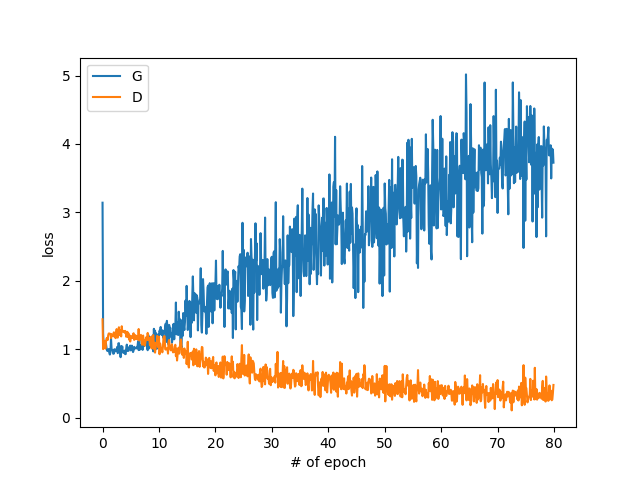
We can see that the a) version of loss becomes 0 and there is nothing for the generator to learn. On the other hand, the b) version is well-behaved and is chosen for this experiment.

# Results

### Disentanglement results



### Loss curve during training phase



# Discussion

15

MNIST

* 1. Introduction (5%)
* 2. Experiment setups: (20%)
  + A. How you implement InfoGAN
    - i. Adversarial loss
    - ii. Maximizing mutual information
  + B. Which loss function of generator you used? What’s different?
* 3. Results (30%)
  + A. Results of your samples (shown as in the expected results section)
  + B. Training loss curves
* 4. Discussion (15%)
* 5. Demo (20%)
  + A. Given a label, you have to generate corresponding images
* (*optional*) Bonus: **Facescrub-5** (10%)
  + Results of your samples (5%)
  + Demo (5%)