Language Technologies Institute

Carnegie Mellon University

Generative Adversarial Nets for Image Generation

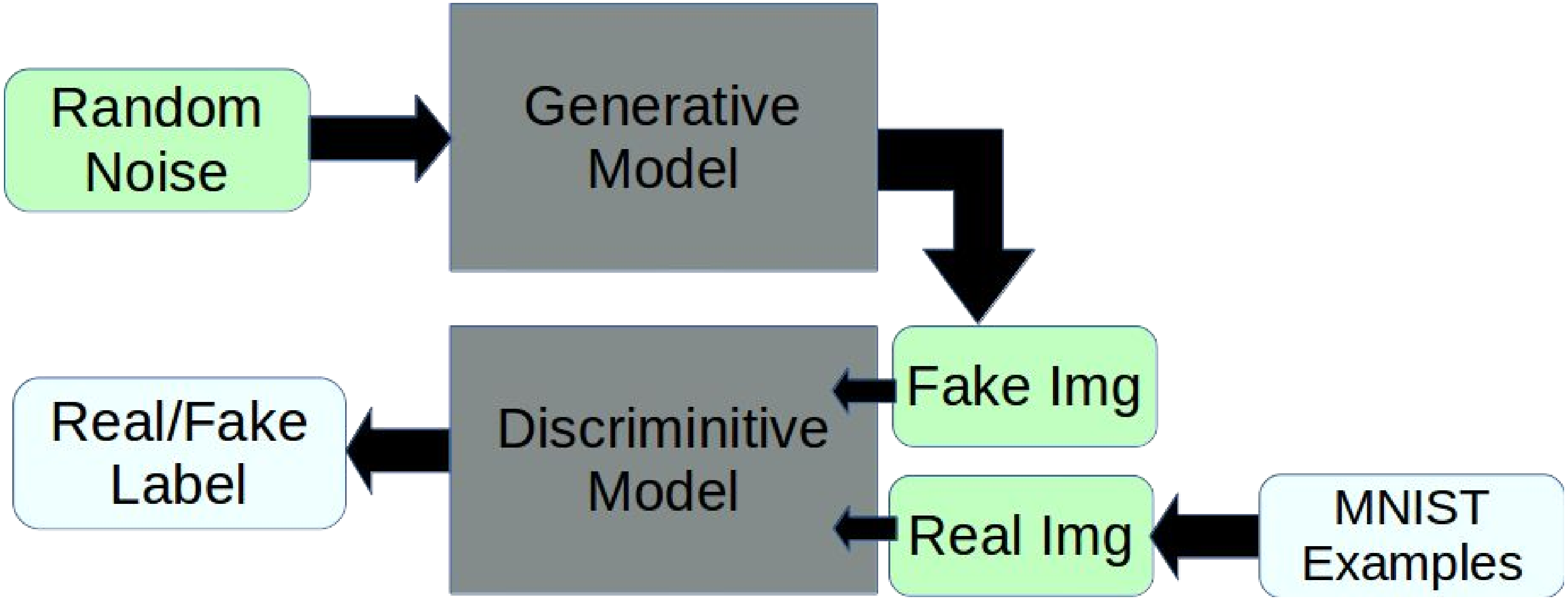
Chirag, Hongliang, Lyndon

Introduction

- Generative Adversarial Nets
  - Discriminative Model $D(\boldsymbol{x}; \theta_d)$
    - Learn to determine whether a sample is from the data or the generative model
  - Generative Model $G(\boldsymbol{z}; \theta_g)$
    - Produce fake samples
    - $\boldsymbol{z}$: Noise variable
  - Prior of the noise variable $p_z(\boldsymbol{z})$

Introduction

- Generative Adversarial Nets



Introduction

- Generative Adversarial Nets
  - Two-player minmax game

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

x came from the data

Generated samples $G(z)$ is not from the data

Introduction

- Generative Adversarial Nets
  - Two-player minmax game

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

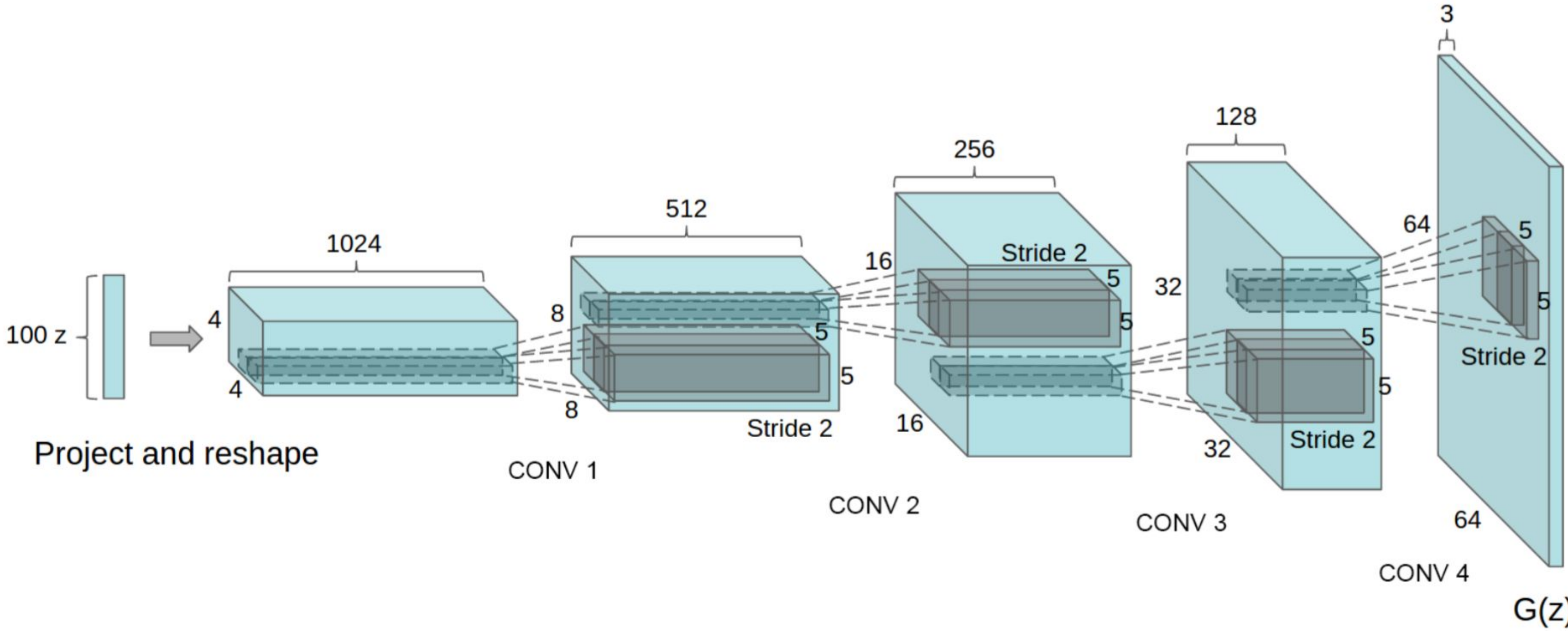
x came from the data

Generated samples $G(z)$ is not from the data

  - D Optimization
    - Maximize the probability of assign the correct labels
  - G Optimization
    - Minimize $\log(1\text{-}D(G(z)))$

Implementation

- Discriminator: ConvNets
- Generator: DeconvNets



Implementation

- Optimization: Minibatch Gradient Descent

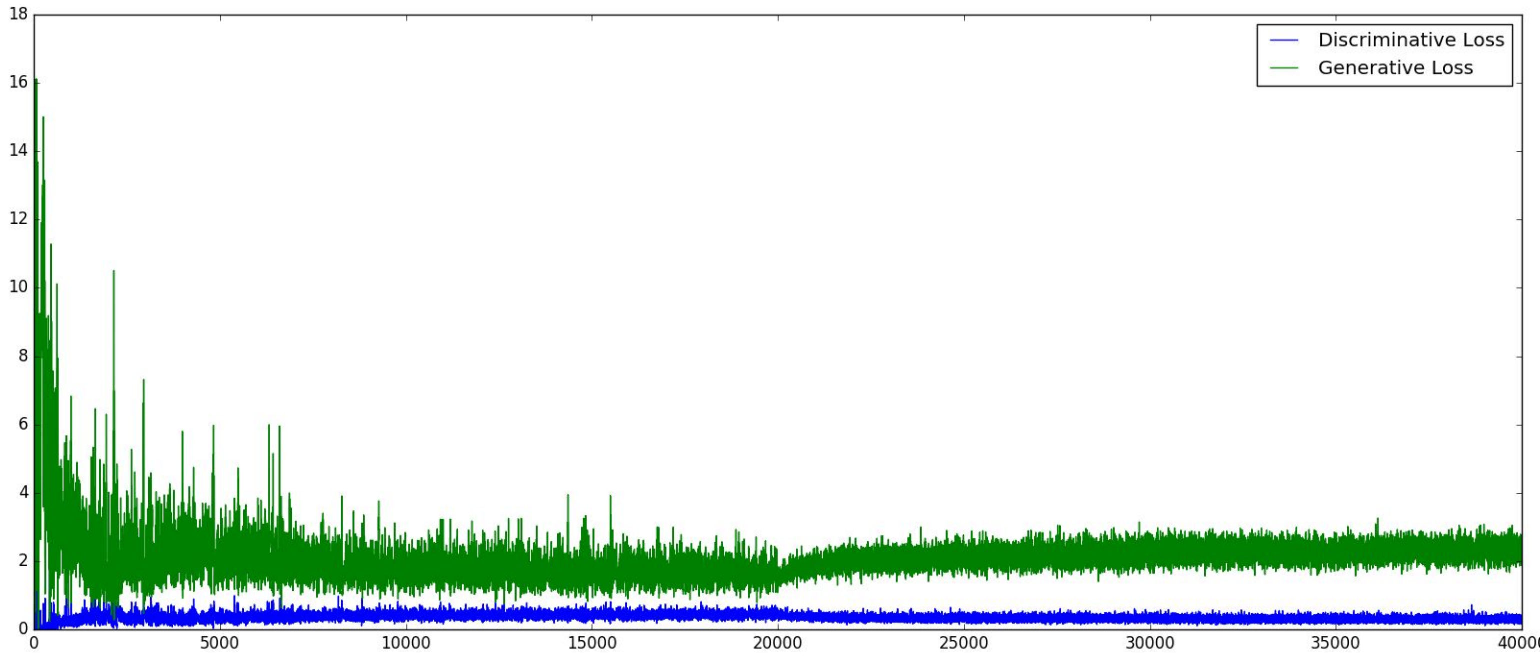
1. *Generate images using* $G(z)$
2. *Batch update of weights in* D *given* $G(z)$, $\boldsymbol{x}$, *and labels*
3. *Batch update of weights in* G *to minimize* $-\log(D(G(z)))$
4. Go to 1, Repeat …

Experiments

- Dataset
  - MNIST
    - 70,000 digits
  - CelebA
    - 202,599 face images
- Preprocess
  - Resize the images in CelebA dataset to 28 * 28
- Optimizer
  - Adam
  - Decrease the learning rate after predefined number of iterations

Experiments

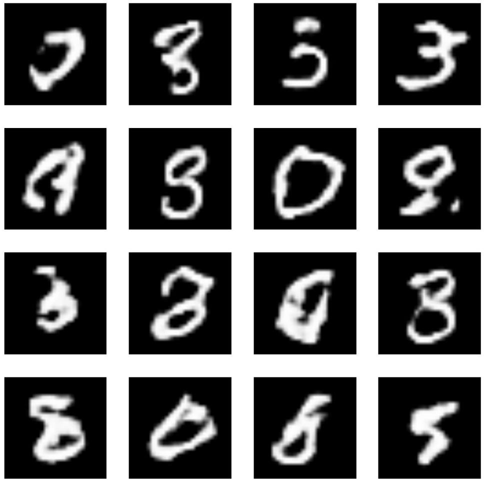
- Discriminator loss vs. generator loss (on CelebA dataset)



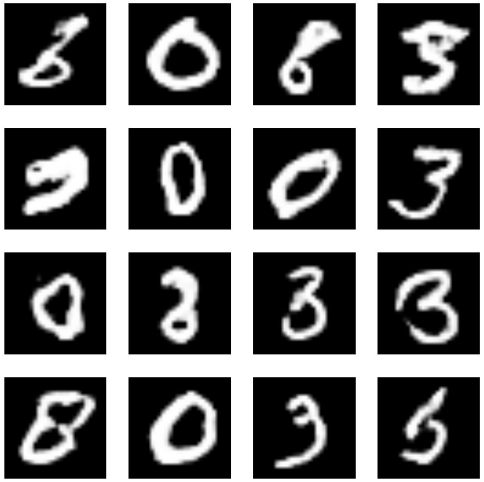
Experiments

- Visualization of MNIST

After 500 iterations



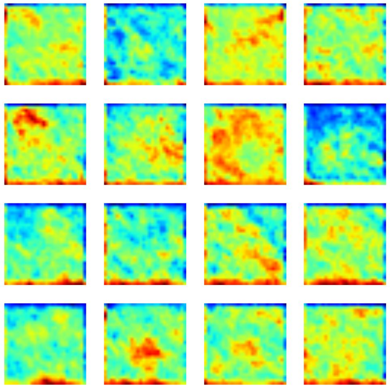
After 3000 iterations



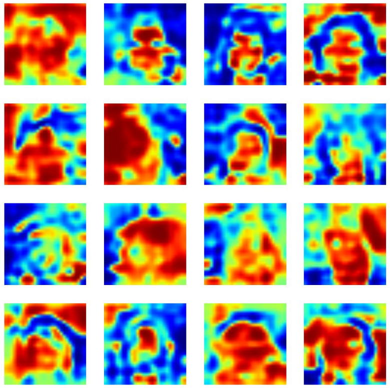
After 10000 iterations

Experiments

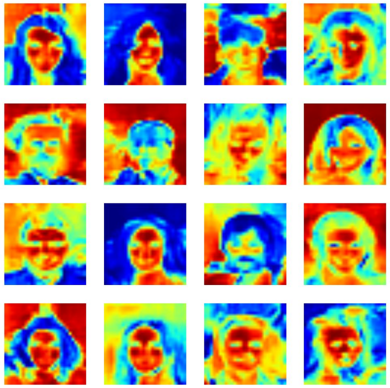
- Visualization of CelebA



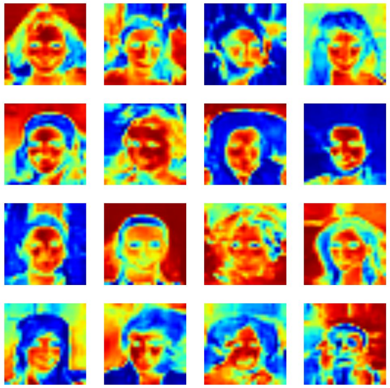
After 500 iterations



After 2000 iterations



After 20000 iterations



After 40000 iterations

InfoGAN

Motivation

Learning interpretable and meaningful representations

InfoGAN

Motivation

Learning interpretable and meaningful representations

Main idea

Maximising the mutual information between a subset of the generators noise variables and observed data. These subset of variables are called latent codes.

InfoGAN

Mutual Information - Definition

The mutual information between random variables X and Y, I(X:Y) measures the "amount of information" learned about the variable X from the knowledge of variable Y

InfoGAN

Mutual Information - Definition

The mutual information between random variables X and Y, I(X:Y) measures the “amount of information” learned about the variable X from the knowledge of variable Y

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$$

InfoGAN

Mutual Information - Interpretation

I(X;Y) represents the reduction of uncertainty in X when Y is observed.

InfoGAN

Mutual Information - Interpretation

$I(X;Y)$ represents the reduction of uncertainty in X when Y is observed.

If X and Y are independent, $I(X;Y) = 0$.

If X and Y are related by a deterministic, invertible function, maximal mutual information is attained.

InfoGAN

Formulating loss

Given some $x \sim P_G(x)$, we'd like $P_G(c \mid x)$ to have a small entropy.

InfoGAN

Formulating loss

Given some $x \sim P_G(x)$, we'd like $P_G(c \mid x)$ to have a small entropy.

Information regularized minimax game:

$$\min_G \max_D V_I(D, G) = V(D, G) - \lambda I(c; G(z, c))$$

Experiment Setup

Dataset: MNIST

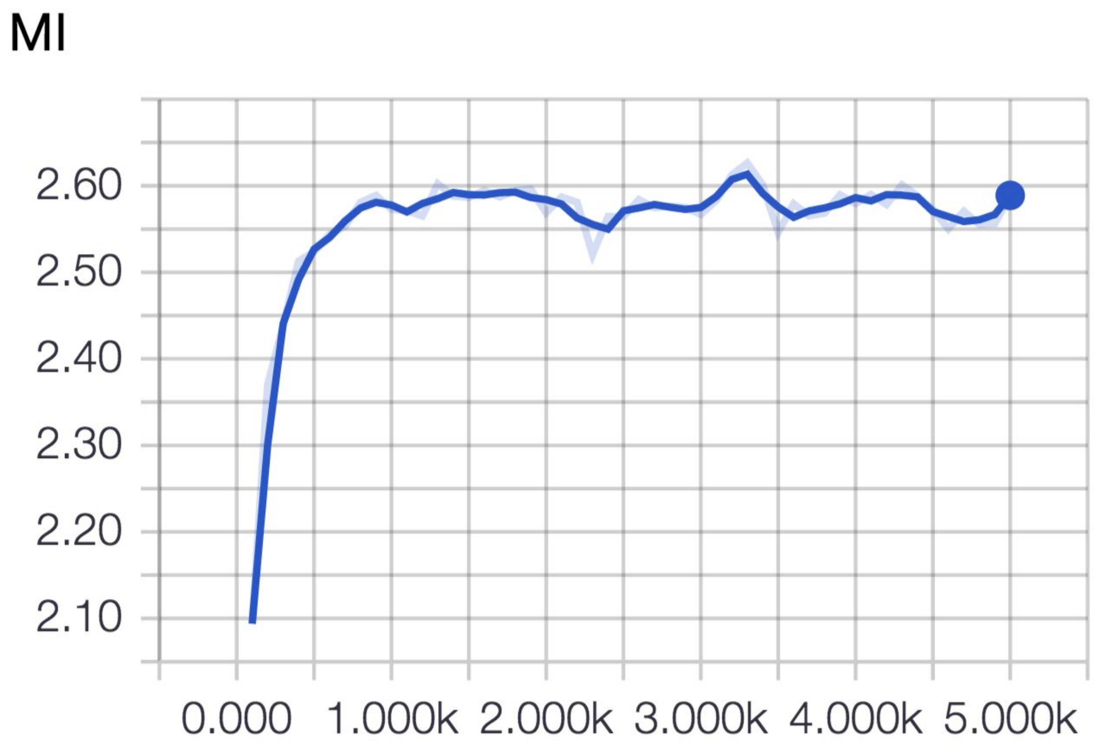
Latend codes setup:

One categorical code, $c1 \sim Cat(K = 10, p = 0.1)$, and two continuous codes: $c2$, $c3 \sim Unif(-1, 1)$.

Lamda: 1

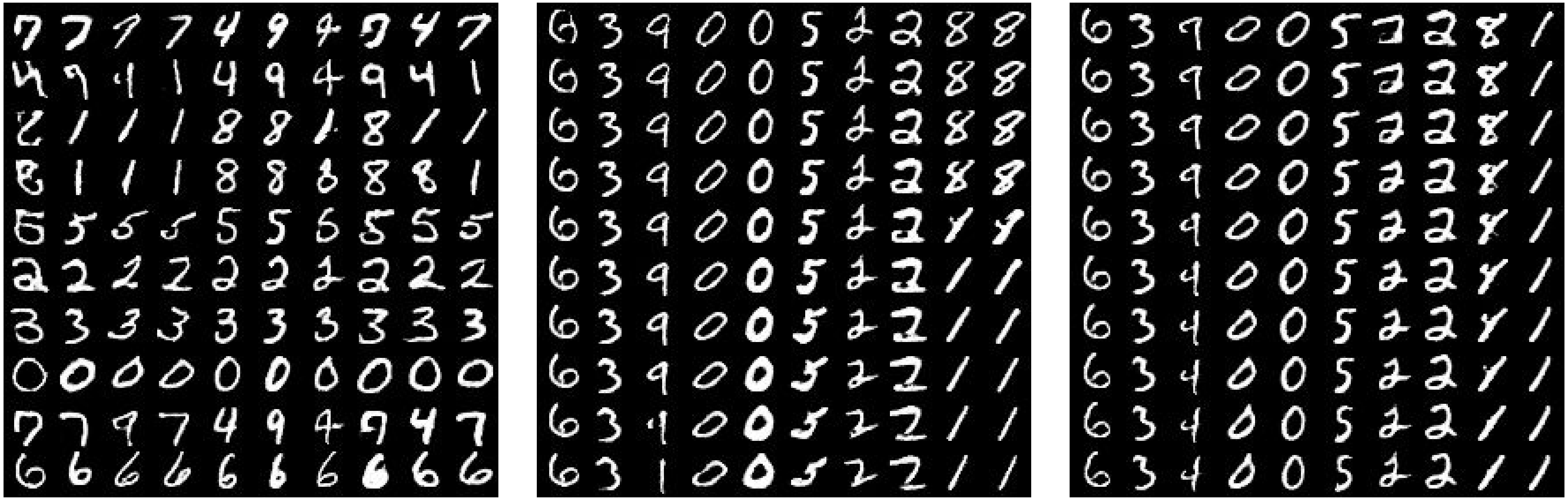
Results

Mutual Information during the training:



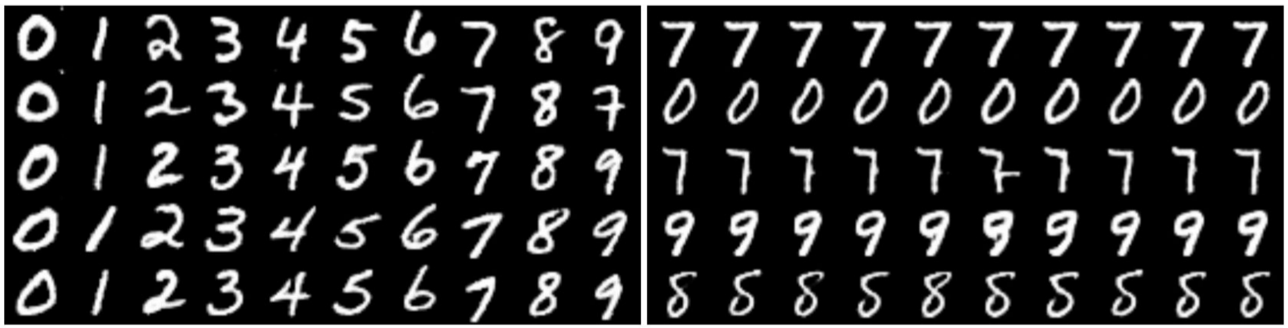
Results

Latent code manipulation results:

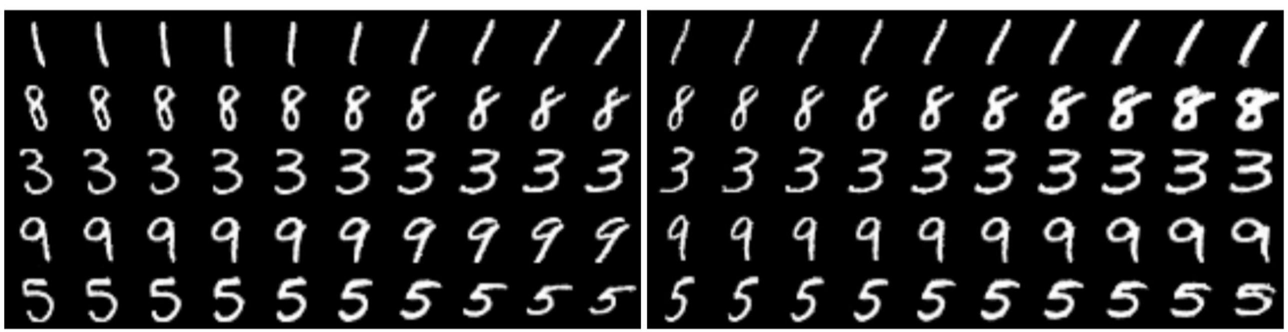


Results

Interpretation:



(a) Varying c_1 on InfoGAN (Digit type) (b) Varying c_1 on regular GAN (No clear meaning)



(c) Varying c_2 from -2 to 2 on InfoGAN (Rotation) (d) Varying c_3 from -2 to 2 on InfoGAN (Width)

Repository

InfoGAN : https://github.com/chiragraman/InfoGAN

Thank you!

Questions?