DLCV Hw4

DLCV 2018 SPRING

TAs : ntu.dlcvta@gmail.com

Outline

Goal

Task Description

Dataset

Grading

Rules

Q&A

Goal

- ✓ Ability to handle large-scale human face data with deep neural network
- ✓ Learn and implement well-known image generation models
- ✓ Gain experience of adversarial training
- ✓ Supervised/unsupervised feature disentanglement

Outline

Goal

Task Description

Dataset

Grading

Rules

Q&A

Task Description

- Image Generation
  - Problem 1 - **V**ariational **A**uto**E**ncoder (VAE)
  - Problem 2 - **G**enerative **A**dversarial **N**etwork (GAN)
- Feature Disentanglement
  - Problem 3 - **A**uxiliary **C**lassifier **GAN**s (ACGAN)
  - Bonus - InfoGAN

Image Generation - VAE

Instead of directly encoding and decoding an image...

<u>Encoder</u>

Learn to predict the mean and std of every dimension of an image in the latent space.

<u>Decoder</u>

Reconstruct an image from a vector sampled from the latent space.

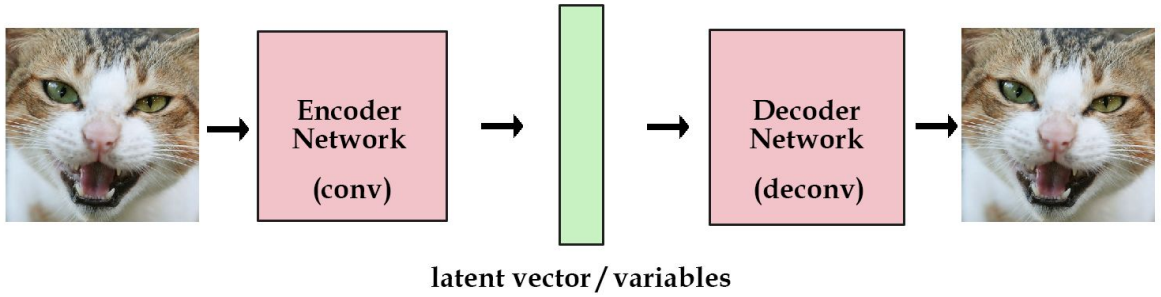
What's the concern of AE?

We have no idea how AE access the latent space.

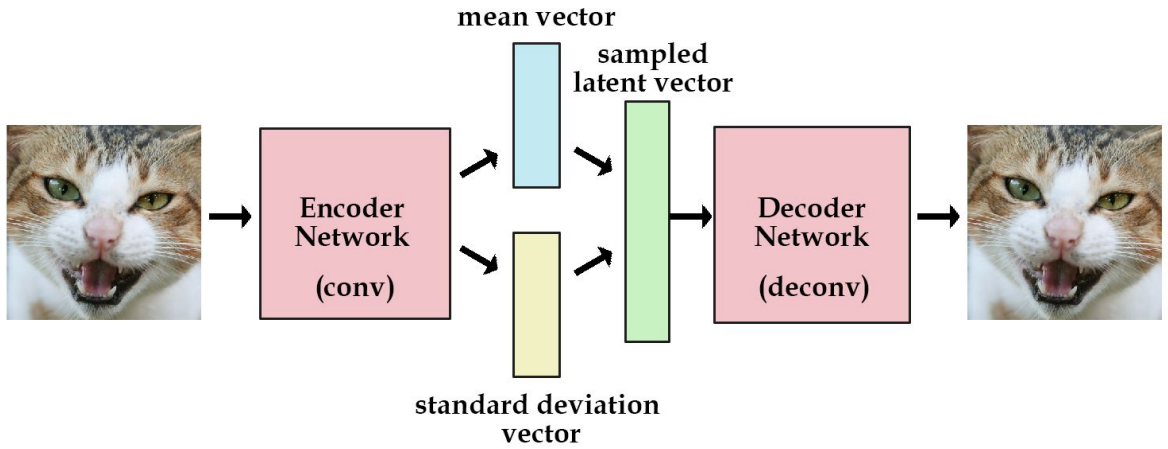
With VAE

One can apply additional restriction to the latent space to make it accessible! (i.e. decoder itself is now a generative model)

AutoEncoder



Variational AutoEncoder



Reference : http://kvfrans.com/variational-autoencoders-explained/

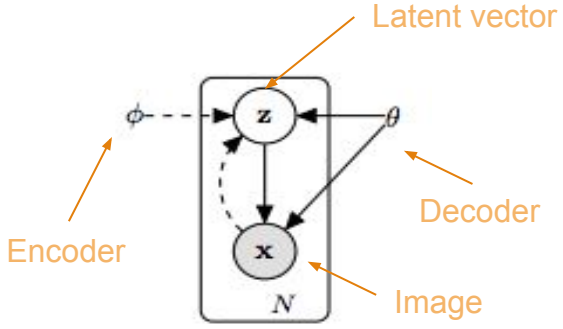
Image Generation - VAE

What's to expect in the latent space?

We'd like to <u>randomly</u> sample from the latent space and get the corresponding image in the pixel space.

Let's force the latent space to be similar to a N-dimension Gaussian!

$$D_{KL}(q_\phi(z|x)||p(z|x)) \qquad \text{where } p(z|x) = N(0,1)$$



Objective function of VAE

$$L_{vae} = L_{reconstruction} + \lambda_{KL} \times L_{KL}$$ where λ_{KL} controls the importance of KL

To be more specific…

$$L_{KL} = -\frac{1}{2}(1 + \log(\sigma^2) - \mu^2 - \sigma^2)$$

In practice, we usually let encoder output mean μ and log variance $logvar$

$$L_{KL} = -\frac{1}{2}(1 + logvar - \mu^2 - e^{logvar})$$

what your code should look like

Image Generation - VAE

The Reparameterization Trick

Random sampling is non-differentiable.

$$z = \mu + \epsilon \times std \quad \text{where} \quad \epsilon \sim N(0,1)$$

The trick allow us to sample latent vector with a differentiable way.

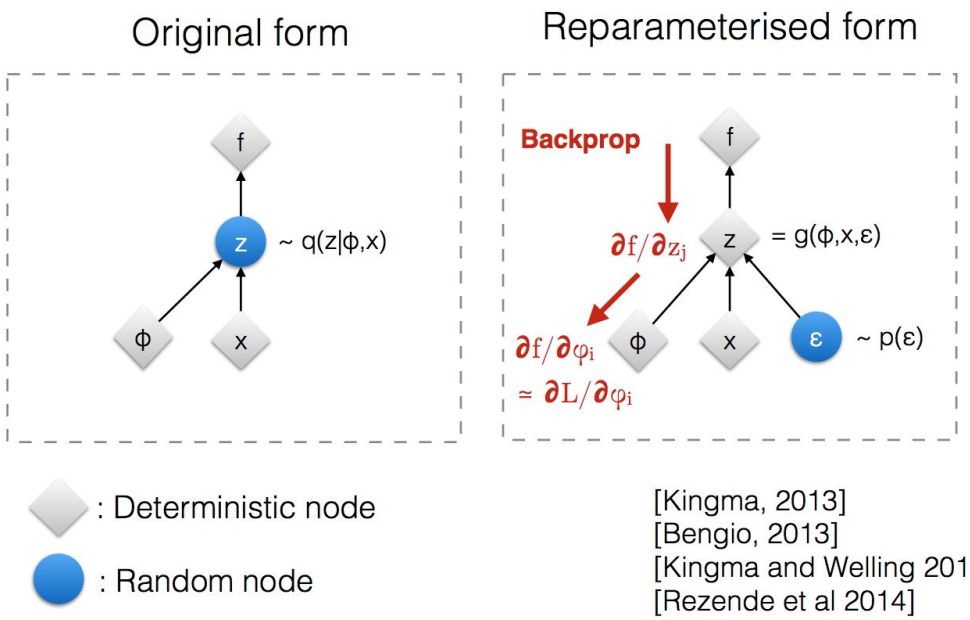
In practice, forwarding from encoder to decoder during training

$$z = \mu + \epsilon \times e^{0.5 \times logvar}$$

For testing,

$$z = \mu$$

what your code should look like



[Kingma, 2013]
[Bengio, 2013]
[Kingma and Welling 2014]
[Rezende et al 2014]

Reference : Kingma's NIPS 2015 workshop slide

Image Generation - VAE Training Tips

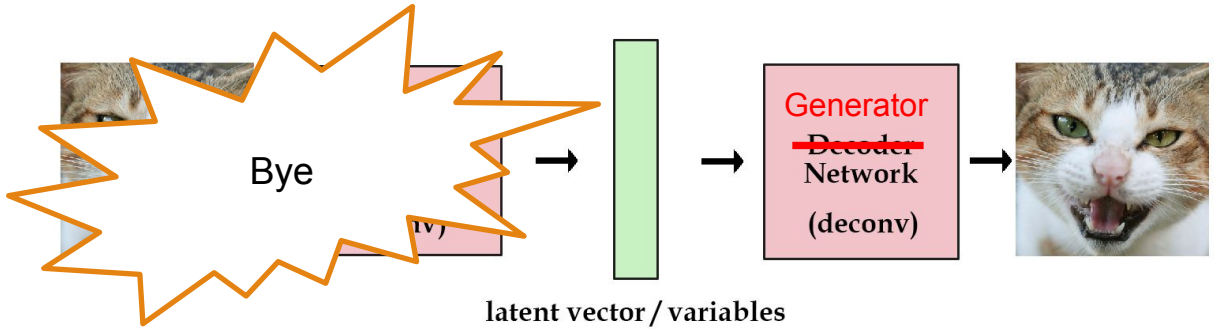
If you still have no idea how to begin, here are some instructions

- Try to implement a normal AutoEncoder and make sure it works first, then modify it to VAE
- Use (pixel-wise averaged) Mean Squared Error as reconstruction loss
- For the choice of latent space dimensional, you may first pick 512 or 1024 temporarily
- λ_{KL} is the weight between "Reconstruction quality" and "Generated image quality"
- You may select λ_{KL} = 1e-5 first and adjust it according to the image quality you received.
- If MSE is not less than 0.05 after first 20k steps , it's probably never going to be. Start over.

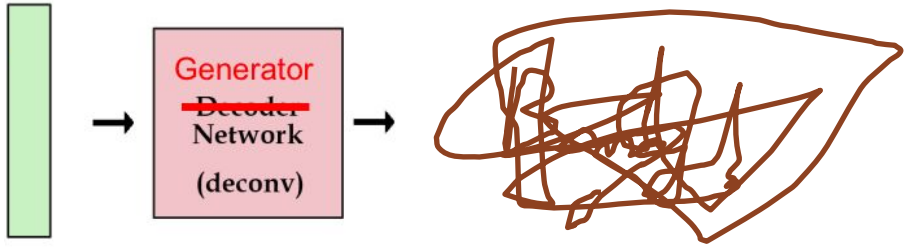
VAE is relatively friendly generative model in terms of training and tuning,
you should finish it ASAP.

Image Generation - GAN

How about generating image without encoder?



A generative network! But, this will likely to happen…



How can we ensure the quality of generated images?

Image Generation - GAN

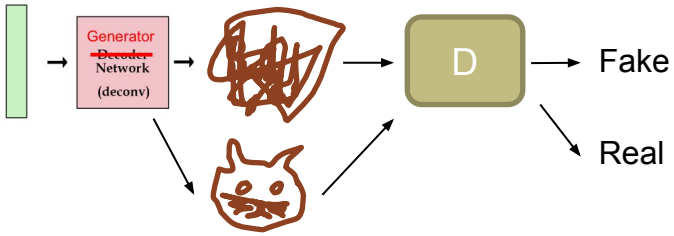
Generative Adversarial Network

Adversarial

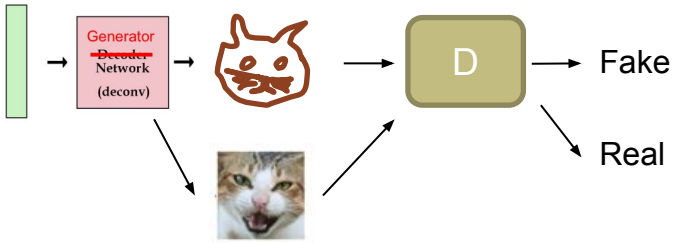
Generator network

Trained to generate image with a random vector

epoch = 1

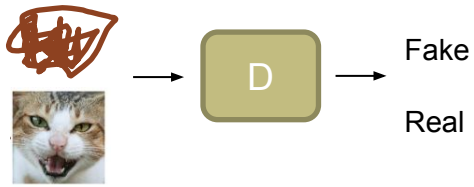


epoch = 2



Discriminator network

Trained to tell if an image is real or generated



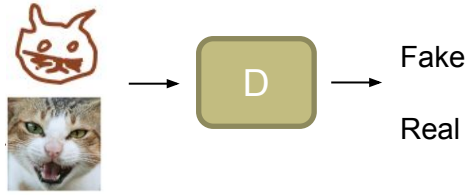
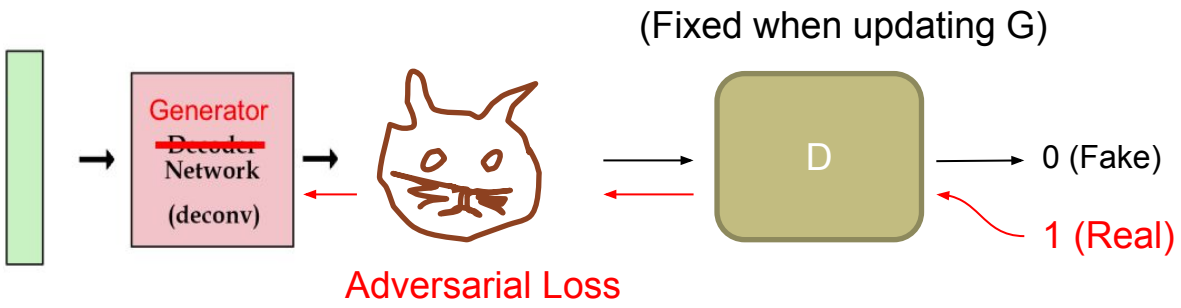


Image Generation - GAN

Adversarial Training

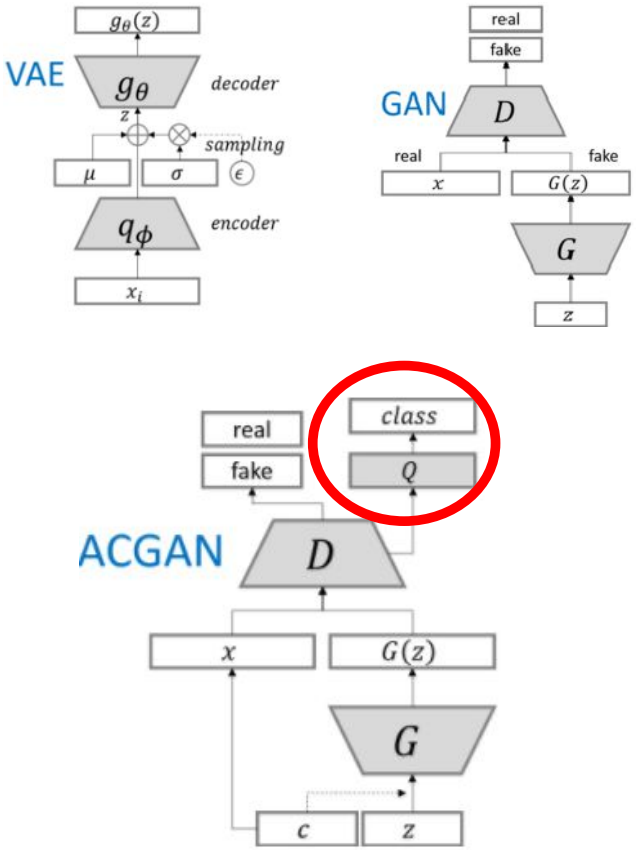
How can the generator fool the discriminator?



Two networks should have similar capacity in order to remain competitive training progress.

Feature Disentanglement- ACGAN

Adding auxiliary classifier to disentangle specific attribute…



<u>Discriminator</u>

- classify if an image is real or not
- classify attribute of image (minimize cross entropy)

$\Longrightarrow$ <u>D and Q should be jointly trained</u>

<u>Generator</u>

- Fool the discriminator on image reality
- Satisfy the discriminator by presenting the inputting attribute

<u>ACGAN is trained with supervised (image, attribute) pair.</u>

What if there is no paired (x, c) during training?

Feature Disentanglement- InfoGAN

Unsupervised feature disentanglement

Discriminator

- classify if an image is real or not
- ~~classify attribute of image (minimize cross entropy)~~

Generator

- Fool the discriminator on image reality
- Preserve the information assigned by c in $G(z)$ and let c be recognize by Q

Different from ACGAN, G and Q should be jointly trained

In the paper, author uses information theory to proofed th reason why InfoGAN works.

(a) Varying c_1 on InfoGAN (Digit type)

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

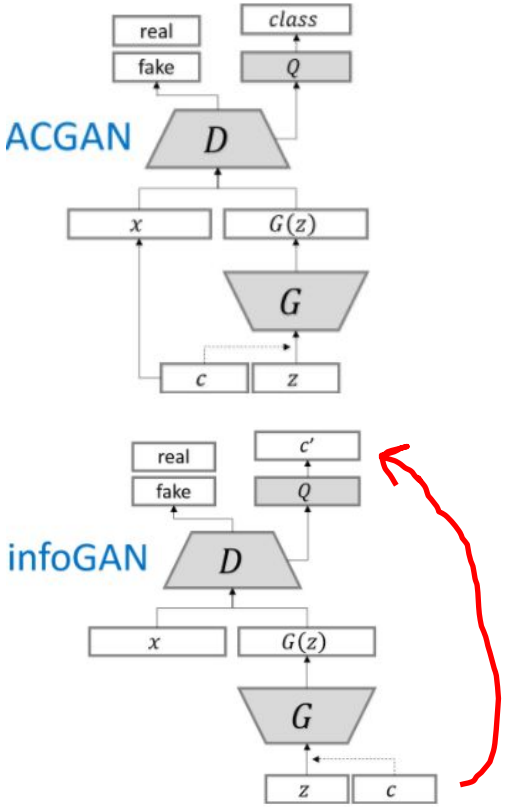


Image Generation – GAN Training Tips

Architecture Guideline proposed by Radford et al. in DCGAN

Architecture guidelines for stable Deep Convolutional GANs

- Replace any pooling layers with strided convolutions (discriminator) and fractional-strided convolutions (generator).
- Use batchnorm in both the generator and the discriminator.
- Remove fully connected hidden layers for deeper architectures.
- Use ReLU activation in generator for all layers except for the output, which uses Tanh.
- Use LeakyReLU activation in the discriminator for all layers.

Well known tips and tricks published on github, recommended to take a look before you start.

TA's experience and hints

- Survey related papers and use similar architecture to them helps.
- Trace the accuracy of discriminator network to see if Gnet and Dnet performance matches.
- Improved GAN algorithm is harder to implement but easier to train (e.g. WGAN, WGAN-GP)
- Use your VAE's decoder/encoder as the generator/discriminator prototype.

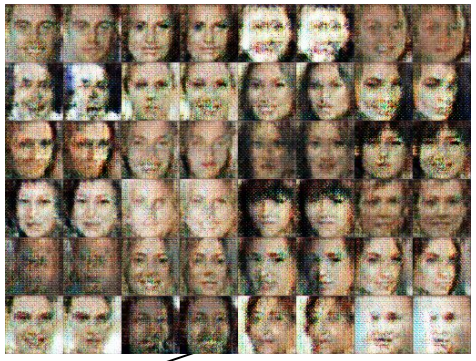
GAN is often difficult to train and tune, start this part early may help a lot.

Image Generation – GAN Training Tips

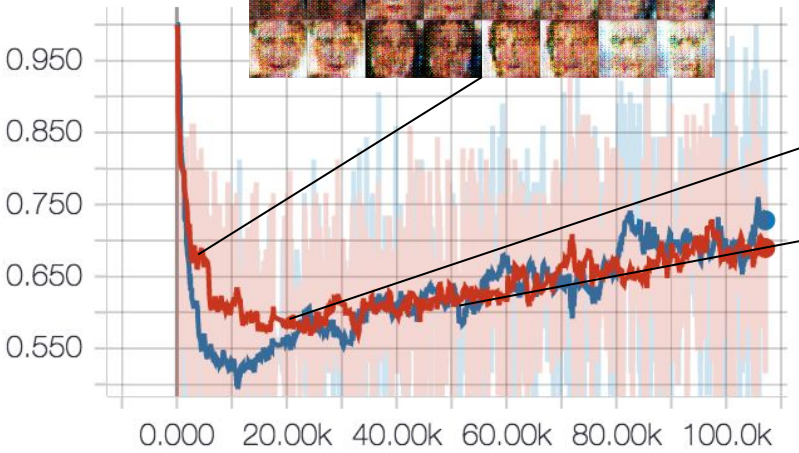
Generated example over training steps

After training 5k steps

After training 20k steps



After training 50k steps



Hints

- Basic outline of face should appear in early stage of training
- Stop training if output does not make sense after 5~15k steps
- Color is NOT at concern in early stage
- More training steps usually brings more image quality

Outline

Goal

Task Description

Dataset

Grading

Rules

Q&A

Dataset - Description

A subset of human face dataset CelebA

- Images are cropped and downscaled to 64x64
- 40000 training samples (about 21% of complete CelebA)
- 13 out for 40 attributes are provided for feature disentanglement experiments

```
Training Attribute Distribution (% of smaple with attribute == 1)
Bangs                   18.4825
Big_Lips                18.3075
Black_Hair              27.9925
Blond_Hair              14.3700
Brown_Hair              27.2625
Heavy_Makeup            38.6300
High_Cheekbones         39.5025
Male                    37.1950
Mouth_Slightly_Open     42.1600
Smiling                 41.1050
Straight_Hair           25.2775
Wavy_Hair               32.3700
Wearing_Lipstick        49.0125
```

Note : Download link will be available on ceiba .

Dataset - Format

```
hw4_data
├── test          # 2621 Files for VAE testing
├── test.csv      # Testing file attributes
├── train         # 40000 images for training
└── train.csv     # Training file attributes
```

train/ - 40000 training images (00000.png ~ 39999.png)
test/ - 2621 testing images for VAE testing (40000.png ~ 42620.png)
Testing set is not allowed to use during training in problem 1.

train.csv/test.csv - Header in first row, <img_filename>, <attribut_1>, <attribut_2>,…, <attribut_13> afterward

```
image_name,Bangs,Big_Lips,Black_Hair,Blond_Hair,Brown_Hair,Heavy_Ma
keup,High_Cheekbones,Male,Mouth_Slightly_Open,Smiling,Straight_Hair
,Wavy_Hair,Wearing_Lipstick
00000.png,0.0,0.0,0.0,0.0,1.0,0.0,1.0,0.0,1.0,1.0,0.0,0.0,0.0
00001.png,0.0,0.0,0.0,0.0,0.0,1.0,1.0,0.0,0.0,0.0,0.0,1.0,1.0
00002.png,0.0,0.0,1.0,0.0,0.0,0.0,0.0,0.0,1.0,1.0,0.0,0.0,0.0
00003.png,0.0,0.0,1.0,0.0,0.0,0.0,1.0,1.0,1.0,1.0,1.0,0.0,0.0
...
```

Outline

Goal

Task Description

Dataset

Grading

Rules

Q&A

Grading – Overview

Problem 1. VAE (6%)

Problem 2. GAN (5%)

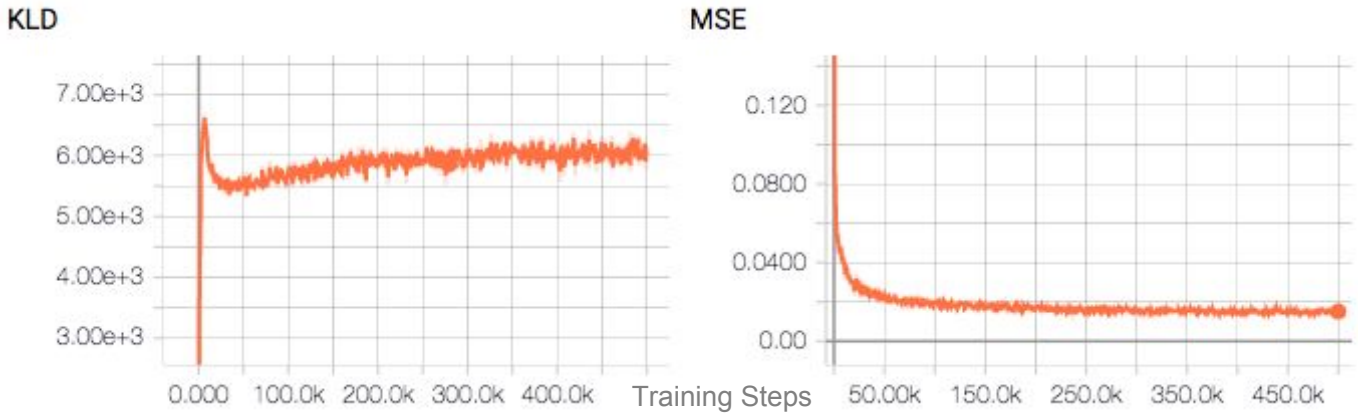
Problem 3. ACGAN (4%)

Bonus InfoGAN (up to 5%)

Grading – Problem 1

Problem 1. VAE (6%)

1. Describe the architecture & implementation details of your model (1%)
2. Plot the learning curve (reconstruction loss & KL divergence) of your model [fig1_2.jpg] (1%)



Example for fig1_2.jpg

Note : Please specify the meaning of axis. If the curve is a mess, you may smoothen it

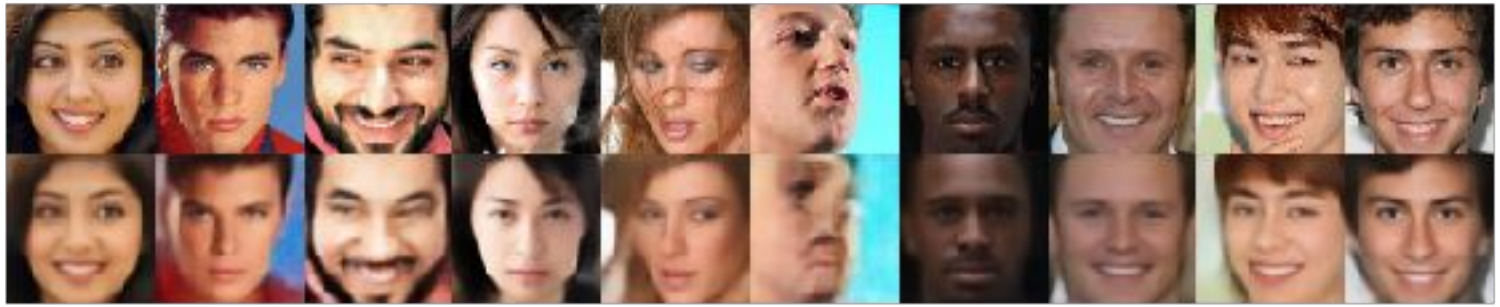
Grading – Problem 1

Problem 1. VAE (6%)

3. Plot 10 testing images and their reconstructed result of your model [fig1_3.jpg] and report your testing MSE of the entire test set(1%)

Input

Reconstruct



Example for fig1_3.jpg

Grading – Problem 1

Problem 1. VAE (6%)

4. Plot 32 random generated images of your model [fig1_4.jpg] (1%)

Example for fig1_4.jpg

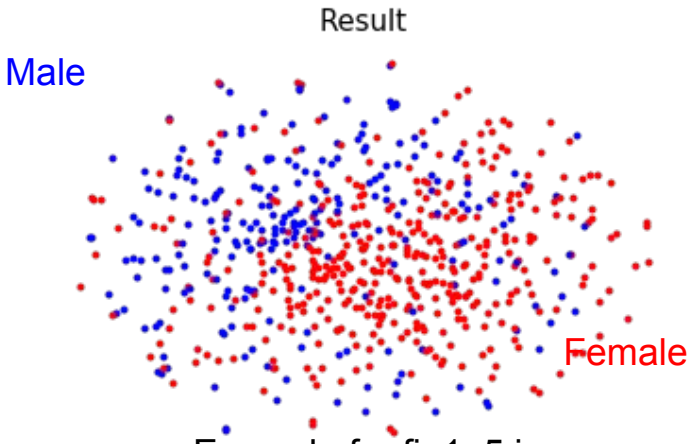
Note : You don't need to manually select the random vectors that provides great result, just make sure the figure is reproducible.

Grading – Problem 1

Problem 1. VAE (6%)

5. Visualize the latent space by mapping test images to 2D space (with tSNE) and color them with respect to an attribute of your choice [fig1_5.jpg] (1%)

6. Discuss what you've observed and learned from implementing VAE. (1%)



Example for fig1_5.jpg

Note : The result may be different case by case depending on your model and tSNE parameters.

Grading – Problem 2

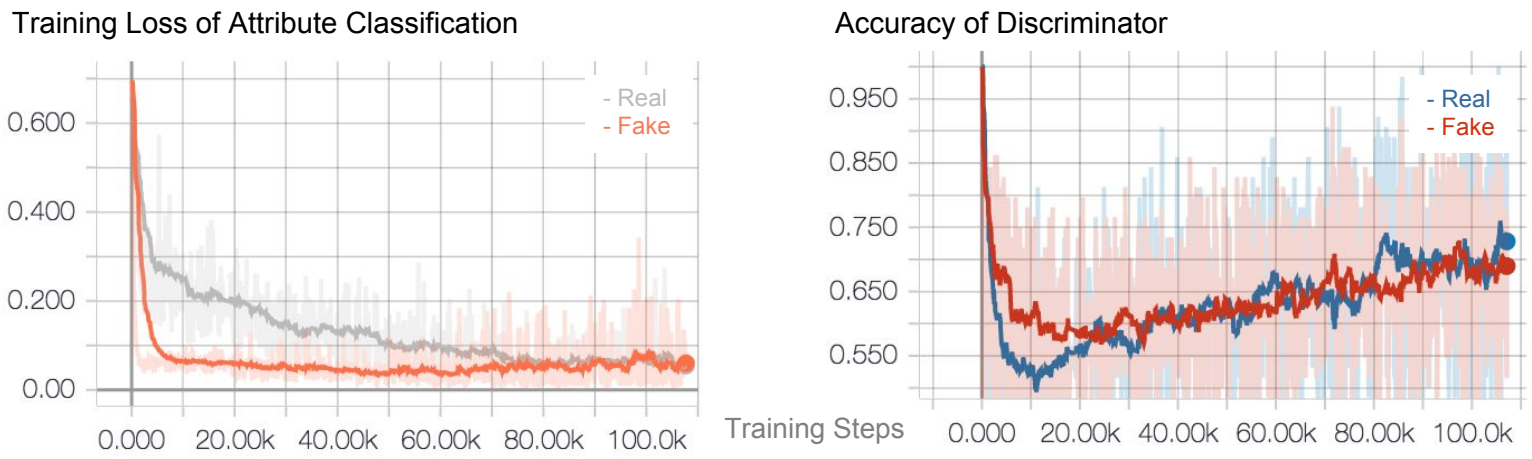
Problem 2. GAN(5%)

1. Describe the architecture & implementation details of your model (1%)
2. Plot the learning curve (in the way you prefer) of your model and briefly explain what you think it represents [fig2_2] (1%)
3. Plot 32 random generated images of your model [fig2_3] (1%)
4. Discuss what you've observed and learned from implementing GAN. (1%)
5. Compare the difference between image generated by VAE and GAN, discuss what you've observed (1%)

Grading – Problem 3

Problem 3. ACGAN(4%)

1. Describe the architecture & implementation details of your model (1%)
2. Plot the learning curve (in the way you prefer) of your model and briefly explain what you think it represents [fig3_2] (1%)



Example for fig3_2.jpg

Grading – Problem 3

Problem 3. ACGAN(4%)

3. Plot 10 pair of random generated images of your model, each pair generated from the same random vector input but with different attribute. This is to demonstrate your model's ability to disentangle feature of interest. [fig3_3] (2%)

Not Smiling

Smiling

Example for fig3_3.jpg

Grading – Bonus

Bonus. InfoGAN (up to 5%, TA preserves the right of grading)

Implement InfoGAN and report what you've found.

You will be graded by how you discuss your implementation and how well the experiment result are. Probably 0 if there's nothing impressive, so do not try to work on this bonus problem without making efforts.

Grading – Notes

- Each and every figure required should be 100% reproducible with submission code
- You will get zero point for a figure if it is not reproducible by your code
- Remember to save training log in order to plot learning curve
- All figures should be clear enough to see details (e.g. axis ticks)
- Scoring may be relative, depending on the quality of your report comparing to other's
- You don't need to use all test images in problem 1-5, but tSNE should be execute with fixed random state within the total 10 minutes execution time.
- For problem 2 and 3, ask for permission first if you plan to implement model different from GAN/WGAN/WGAN-GP (principle : input random vector, output image)

Outline

Goal

Task Description

Dataset

Grading

Rules

Q&A

Rules – Submission format

Please submit your homework on Github (not ceiba) before deadline

DLCV2018SPRING/hw4 on your GitHub repository should include the following files:

hw4_YourStudentID.pdf	# Your report.
hw4.sh	# Script for reproducing every required fig in your report
your model files	
other files (e.g., source code)	

Do NOT upload dataset.

If your model is too big to upload, involve automatic-download commands in hw4.sh

Rules – Execution

TA will execute the following command to test your code

```
sh hw4.sh <absolute path of hw4_data/> <output_path/>
```

i.e. in your shell script $1 will be the path of *hw4_data/* folder (with structure identical to dataset format in page 16.)

and $2 will be the path of where all your figures should be saved to.

After executing your script, there should be *fig1_2.jpg* *fig2_2.jpg* *fig1_3.jpg* *fig2_3.jpg* *fig1_4.jpg* *fig3_2.jpg* *fig1_5.jpg* *fig3_3.jpg* under the specified output path

Again, these figures should be identical to corresponding figures in your report

hw4.sh execution time limit is 10 minutes (excluding model download)

Rules – Allowed packages

- Python : 3.6
- Tensorflow : 1.6
- Keras : 2.1.5 (please upgrade.)
- Pytorch : 0.3.1
- h5py : 2.7.1
- Numpy : 1.14.2
- Pandas : 0.22.0
- Matplotlib, Scikit-image, Pillow, Scipy, Python standard Lib
- E-mail or ask TA first if you want to import other packages

Rules – Policy

Deadline : Report & code should be upload before 5/18 23:59:59 (GMT+8)

Late policy : Deducted 30% each day excluding using 3 free late day quota this semester

Academic Ethics : Discussion between classmates is encouraged, however, please do NOT copy (or let someone copy your) homework. As we've handled a number of cases in HWs 1 & 2, plagiarism or cheating will result in F for this class. TA will check the similarity of every submission.

Rules Violation : Violation of any format/execution specification will get zero points on the corresponding problem. Please follow homework spec carefully. If anything is unclear, ask.

External Dataset: Using external dataset is forbidden for this homework.

Rules – Reminder

- Training and tuning GAN is extremely time (and mind…) consuming, start your homework right after today's launch!
- Remember to fix the random seed in your code such that all results are reproducible.
- There is no standard answer for most of the problems in this homework. Just report your result and observation honestly is often good enough.
- This homework is relatively difficult and heavy loaded, don't hesitate to post your question under hw4 post in FB group or come to TA hour.
- Do NOT directly message any TA.
- Test your hw4.sh with the exact command TA will use before submission

Q&A

TAs : ntu.dlcvta@gmail.com