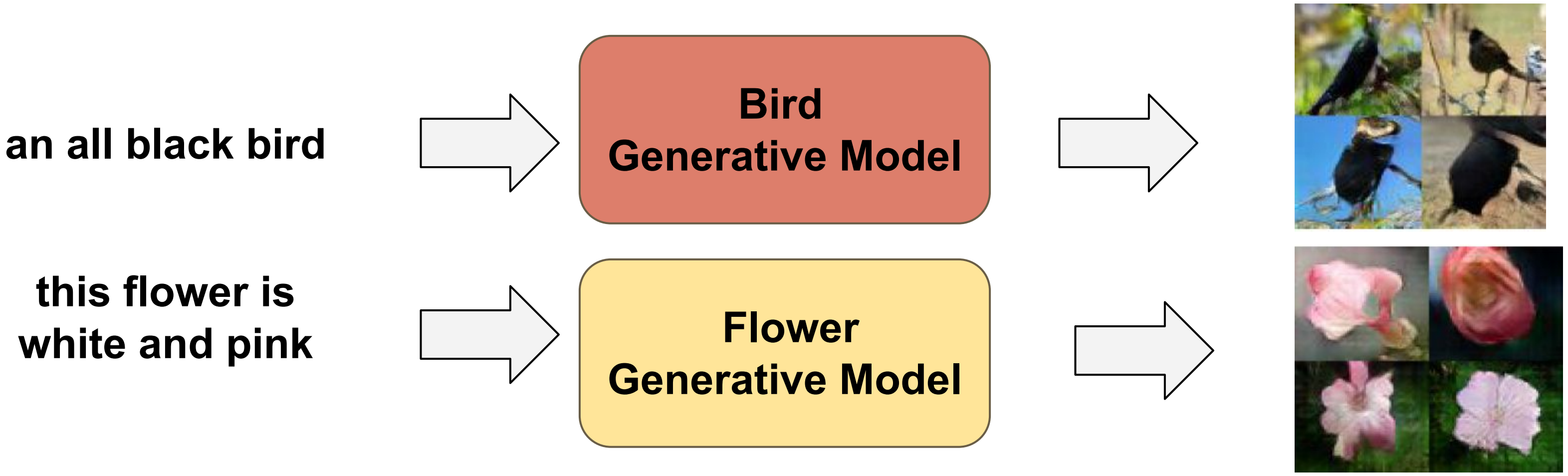
MLDS 2017 Spring
HW3 - Generative Adversarial Networks

mldsntu2017@gmail.com

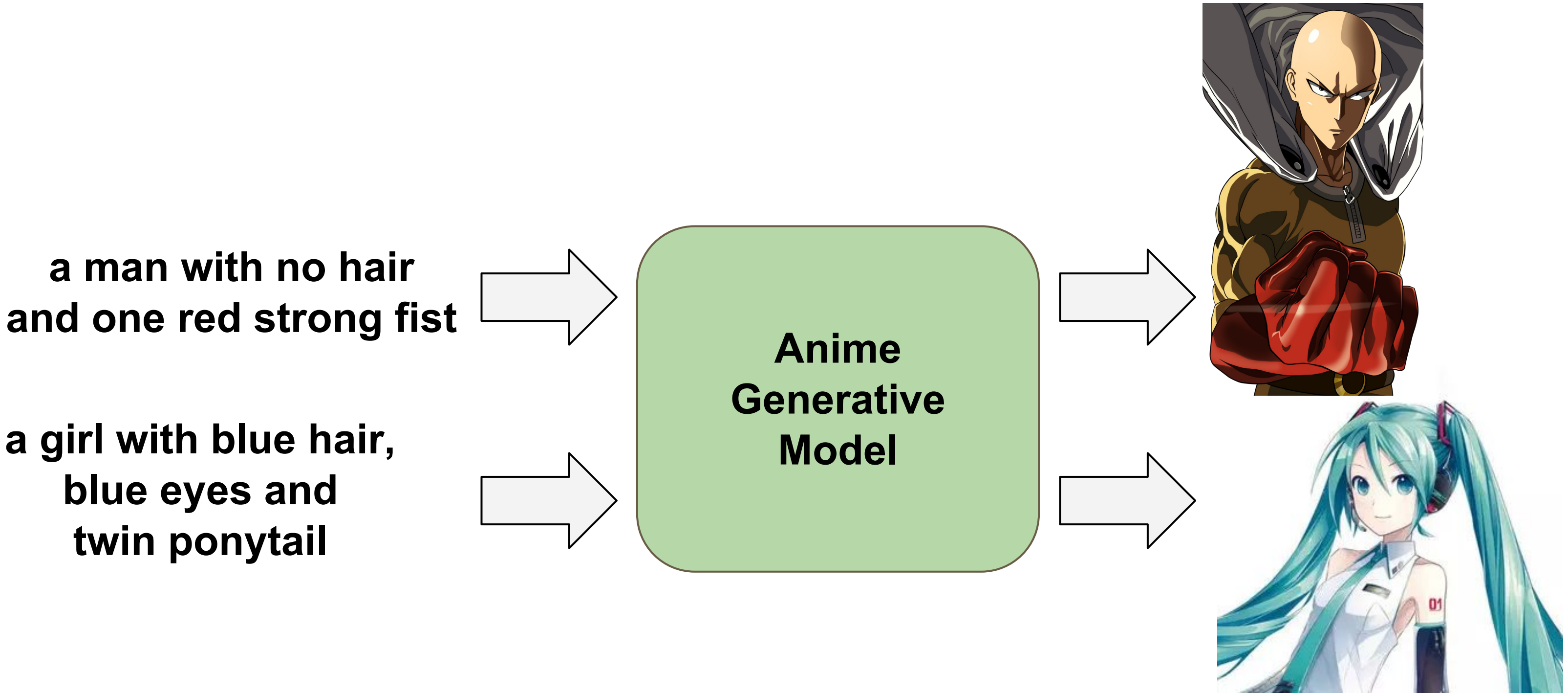
Outline

- Task Introduction
  - Text2image generation
  - Dataset collection
- Model
  - Conditional GAN
  - Tips for training
    - Discriminator loss function
    - Objective function
- Submission and grading

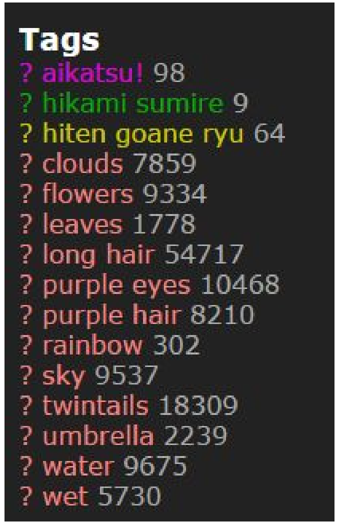
Task Introduction - text2image generation



Task Introduction - text2image generation

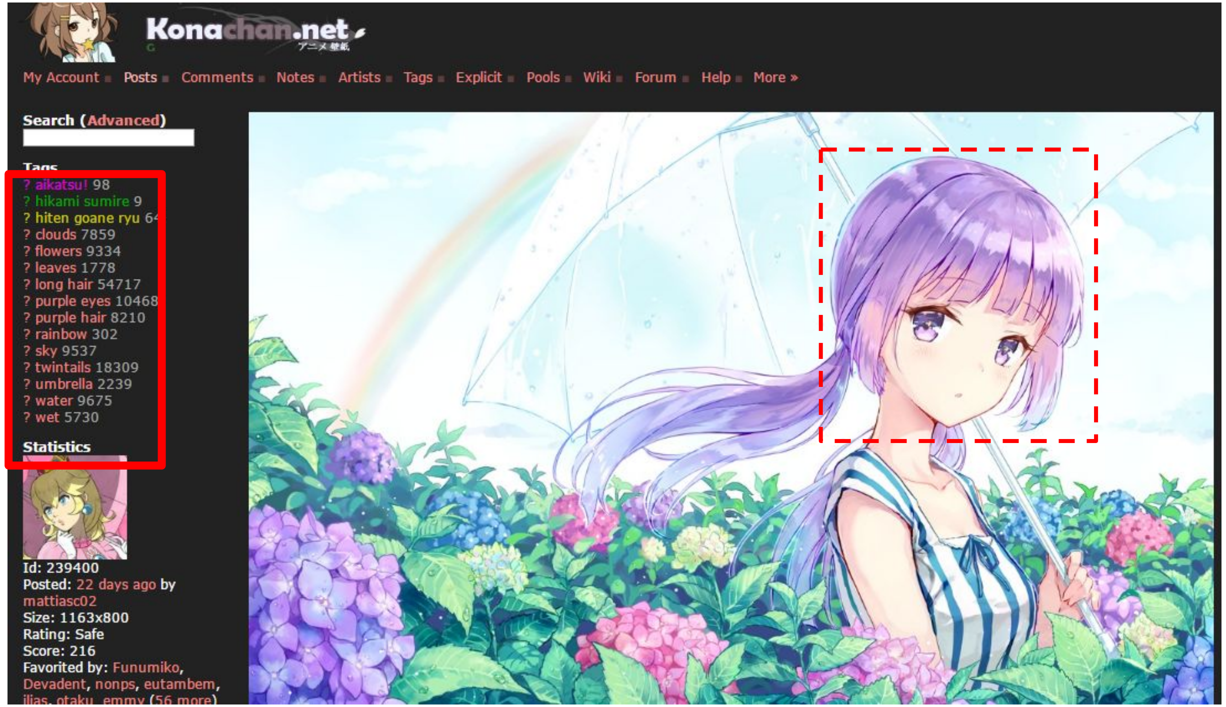


Data Collection



Tags
? aikatsu! 98
? hikami sumire 9
? hiten goane ryu 64
? clouds 7859
? flowers 9334
? leaves 1778
? long hair 54717
? purple eyes 10468
? purple hair 8210
? rainbow 302
? sky 9537
? twintails 18309
? umbrella 2239
? water 9675
? wet 5730

Not all tags are useful



Tags
? aikatsu! 98
? hikami sumire 9
? hiten goane ryu 64
? clouds 7859
? flowers 9334
? leaves 1778
? long hair 54717
? purple eyes 10468
? purple hair 8210
? rainbow 302
? sky 9537
? twintails 18309
? umbrella 2239
? water 9675
? wet 5730

Statistics

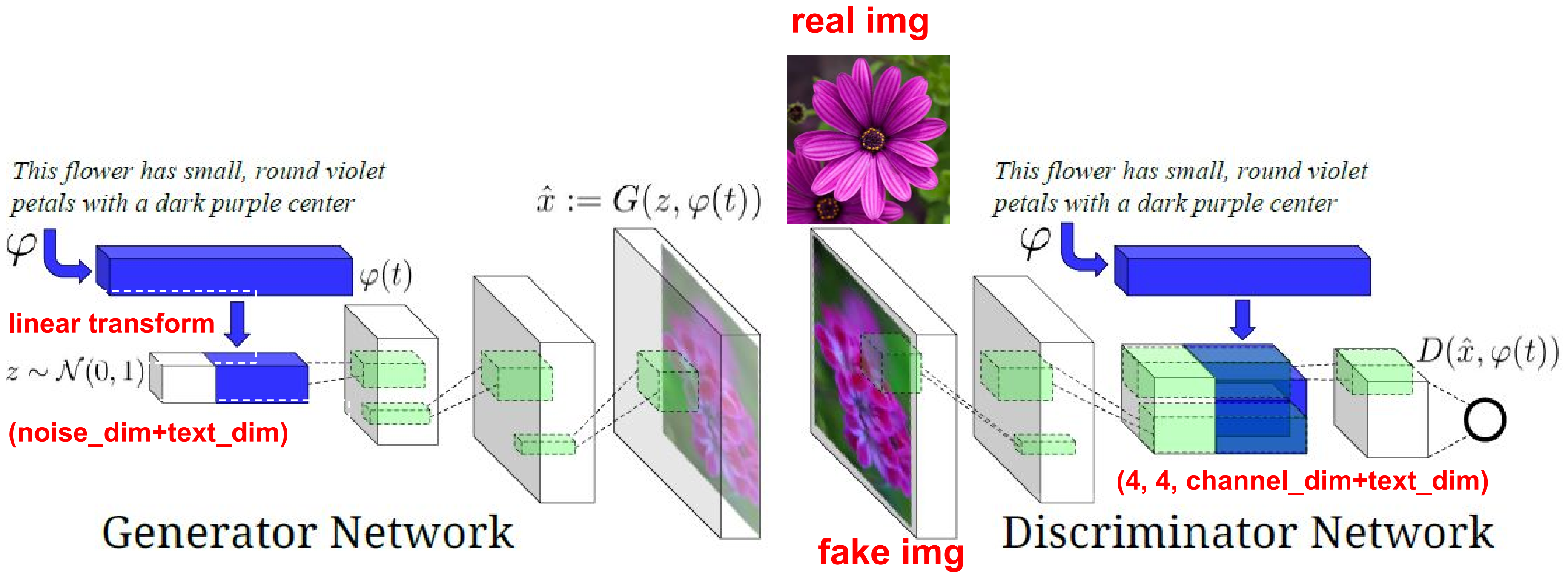
Id: 239400
Posted: 22 days ago by mattiasc02
Size: 1163x800
Rating: Safe
Score: 216
Favorited by: Funumiko, Devadent, nonps, eutambem, ilias, otaku_emmy (56 more)

http://konachan.net/post/show/239400/aikatsu-clouds-flowers-hikami_sumire-hiten_goane_r

感謝樊恩宇助教蒐集data

Model and training tips

Conditional GAN for text2image generation

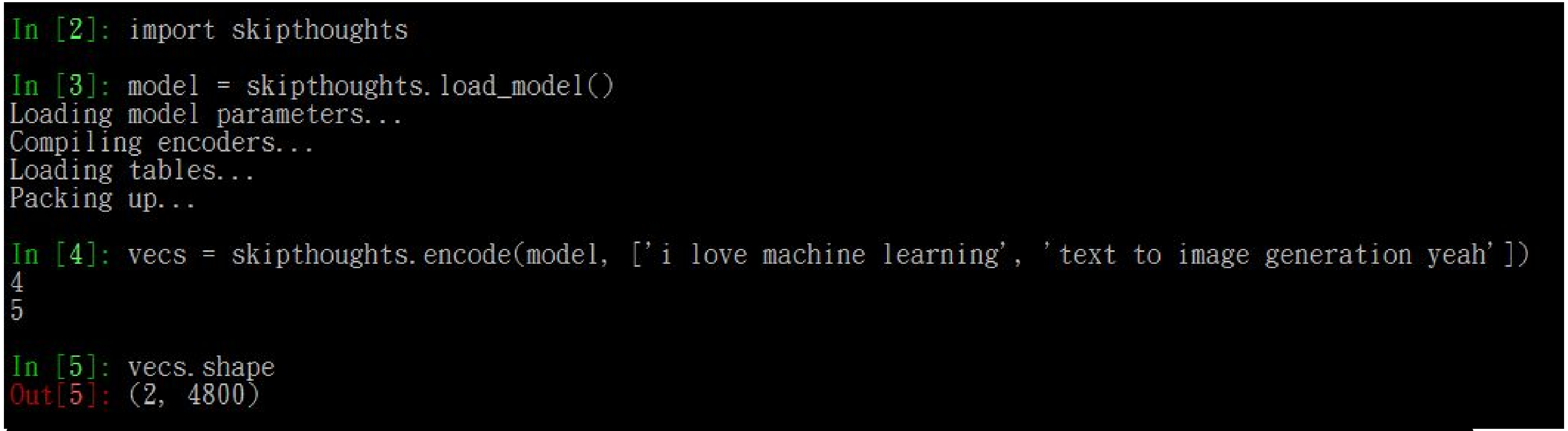

real img
This flower has small, round violet petals with a dark purple center
φ
$\varphi(t)$
linear transform
$z \sim \mathcal{N}(0,1)$
(noise_dim+text_dim)
$\hat{x} := G(z, \varphi(t))$
Generator Network
fake img
This flower has small, round violet petals with a dark purple center
φ
$D(\hat{x}, \varphi(t))$
(4, 4, channel_dim+text_dim)
Discriminator Network

Paper: https://arxiv.org/pdf/1605.05396.pdf

Details for training

- Updates between Generator and Discriminator
  - 1 : 1 or 2 : 1
- ADAM with lr = 0.0002, momentum = 0.5
- gaussian or uniform noise dim = 100
- batch size = 64
- epoch = 600
  - Practically, epoch = 300 is enough

Text feature process tool - Skip-thought vector

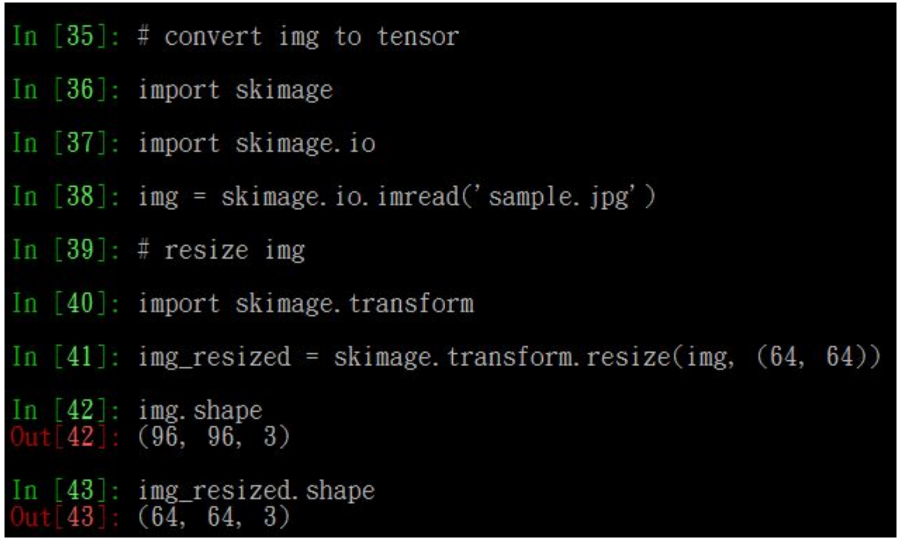


```
In [2]: import skipthoughts

In [3]: model = skipthoughts.load_model()
Loading model parameters...
Compiling encoders...
Loading tables...
Packing up...

In [4]: vecs = skipthoughts.encode(model, ['i love machine learning', 'text to image generation yeah'])
4
5

In [5]: vecs.shape
Out[5]: (2, 4800)
```

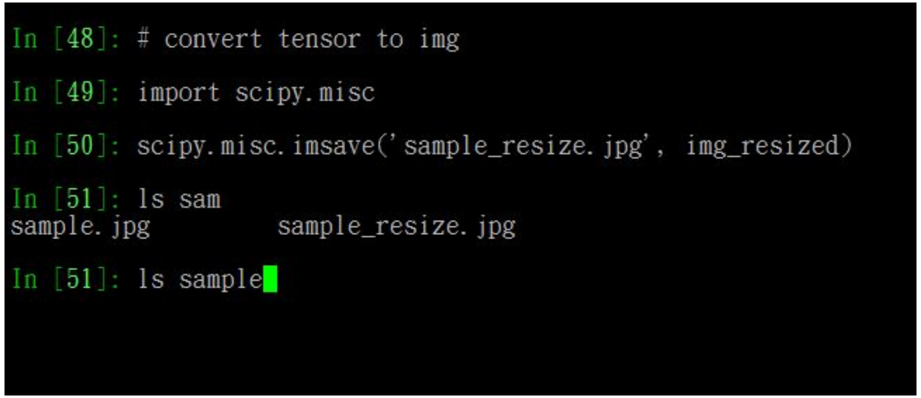
No matter which tool you use to process text input, please make sure you include that pre-trained model in your repository to let us run your code successfully.

skip-thought source code: https://github.com/ryankiros/skip-thoughts

Image process tool - skimage and scipy.misc



```
In [35]: # convert img to tensor

In [36]: import skimage

In [37]: import skimage.io

In [38]: img = skimage.io.imread('sample.jpg')

In [39]: # resize img

In [40]: import skimage.transform

In [41]: img_resized = skimage.transform.resize(img, (64, 64))

In [42]: img.shape
Out[42]: (96, 96, 3)

In [43]: img_resized.shape
Out[43]: (64, 64, 3)
```



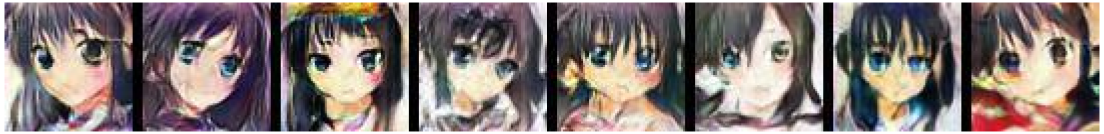
```
In [48]: # convert tensor to img

In [49]: import scipy.misc

In [50]: scipy.misc.imsave('sample_resize.jpg', img_resized)

In [51]: ls sam
sample.jpg          sample_resize.jpg

In [51]: ls sample
```

Install:

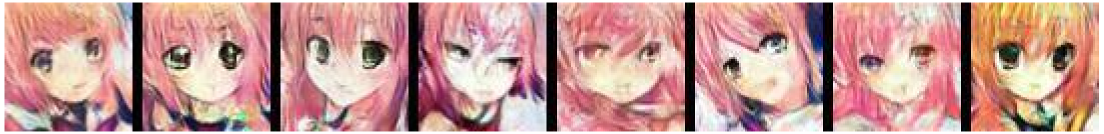
- sudo apt-get install python-skimage
- sudo pip install --user numpy scipy

Little Demo

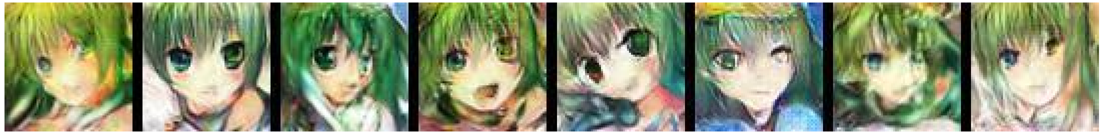
input text: black hair blue eyes



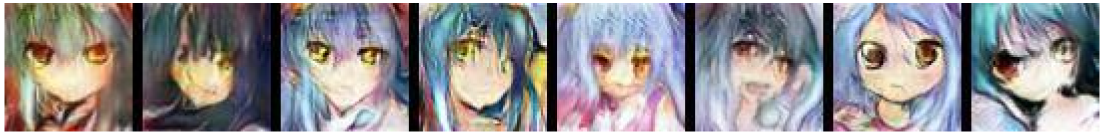
input text: pink hair green eyes



input text: green hair green eyes



input text: blue hair red eyes



Tips for training

- Discriminator output:
  - (real img, right text): 1
  - (fake img, right text): 0
  - (real img, wrong text): 0
  - (wrong img, right text): 0
- Different objective function
  - Wasserstein GAN (WGAN)
  - Least Squares GAN (LSGAN)
  - Boundary-Seeking GAN (BGAN)

Wasserstein GAN

- Earth-Mover (Wasserstain-1) distance

$$W(P_r, P_g) = \inf_{\gamma \in \Pi(P_r, P_g)} E_{(x,y)\sim\gamma}[\|x-y\|]$$
$$= \sup_{\|f\|_L \leq 1} \{E_{x\sim P_r}[f(x)] - E_{x\sim P_\theta}[f(x)]\}$$

EM distance is Continuous
JS-divergence is not

- The optimization problem is thus :

$$\max_{w \in W} \{E_{x\sim P_r}[f_w(x)] - E_{x\sim P_\theta}[f_w(x)]\}$$
$$= \max_{w \in W} \{E_{x\sim P_r}[f_w(x)] - E_{z\sim P_{z(z)}}[f_w(g_\theta(z))]\}$$

Reference : Arjovsky, Martin, Soumith Chintala, and Léon Bottou. "Wasserstein gan." *arXiv preprint arXiv:1701.07875* (2017).

Wasserstein GAN

- For Discriminator

The output of D is thus not probability anymore !
The D loss turn to be a measure of distance

$$\max_{D}\{E_{x\sim P_r}[D(x)]-E_{z\sim P_{z(z)}}[D(G(z))]\}$$

- For Generator

$$\max_{G}\{E_{z\sim P_{z(z)}}[D(G(z))]\}$$

Wasserstein GAN

Implementation Notes :

- Do not apply sigmoid at the output of D
- Clip the weight of D
- Use RMSProp instead of Adam
- Train more iteration of D (the paper use 5)

Least Squares GAN

- For Discriminator

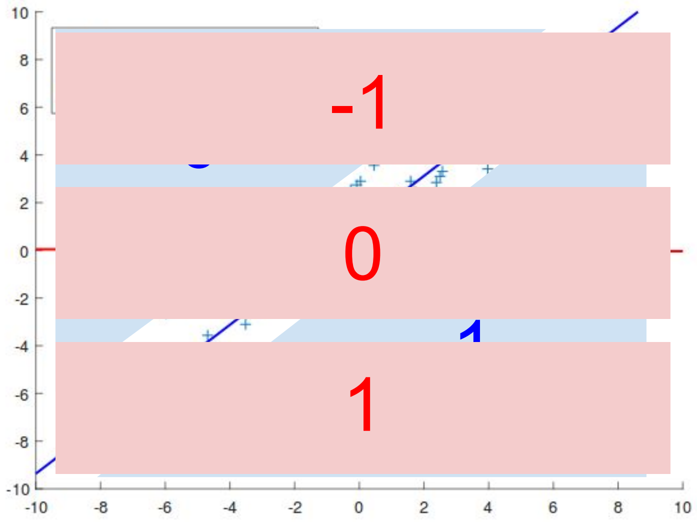
$$\min_G \left\{ \frac{1}{2} E_{x\sim p_{data}(x)}[(D(x)-\overset{1}{b})^2] + \frac{1}{2} E_{z\sim p_z(z)}[(D(G(z))-\overset{-1}{a})^2] \right\}$$

- For Generator

$$\min_G \frac{1}{2} E_{z\sim p_z(z)}[(D(G(z))-\overset{0}{c})^2]$$

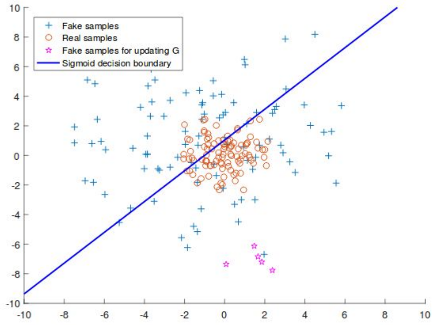
Reference: Mao, Xudong, et al. "Least squares generative adversarial networks." *arXiv preprint ArXiv:1611.04076* (2016).

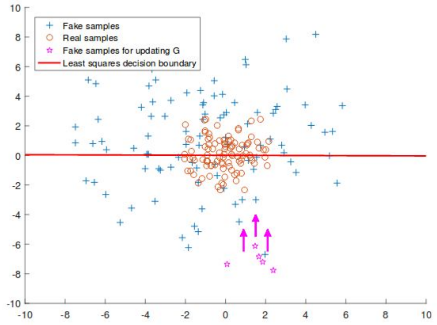
Least Squares GAN

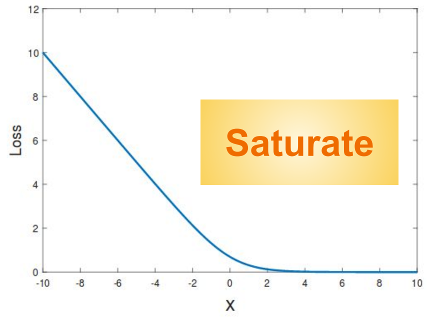


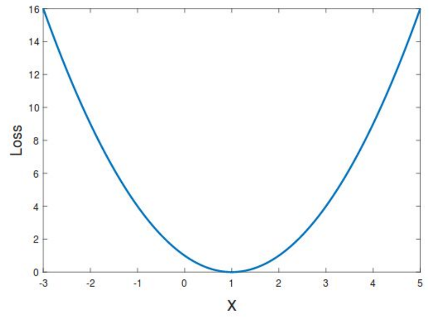
ideal situation
D can't distinguish real and fake samples

Drag to decision boundary









Least Squares GAN

- For Discriminator

$$\min_G \left\{ \frac{1}{2} E_{x \sim p_{data}(x)}[(D(x) - \overset{1}{b})^2] + \frac{1}{2} E_{z \sim p_z(z)}[(D(G(z)) - \overset{0}{a})^2] \right\}$$

- For Generator

$$\min_G \frac{1}{2} E_{z \sim p_z(z)}[(D(G(z)) - \overset{1}{c})^2]$$

Least Squares GAN

Implementation Notes :

- Do not apply sigmoid at the output of D

Boundary-Seeking GAN

Optimum Discriminator :

$$D^*(x) = \frac{P_{data}(x)}{P_{data}(x) + P_g(x)}$$

$$\Downarrow$$

$$P_{data}(x) = P_g(x)\frac{D^*(x)}{1 - D^*(x)}$$

$$\Downarrow$$

For imperfect discriminator :

$$\widetilde{P}(x) = \frac{1}{Z}P_g(x)\frac{D(x)}{1 - D(x)}$$

Z : normalize term

Reference : Hjelm, R. Devon, et al. "Boundary-Seeking Generative Adversarial Networks." *arXiv preprint arXiv:1702.08431* (2017).

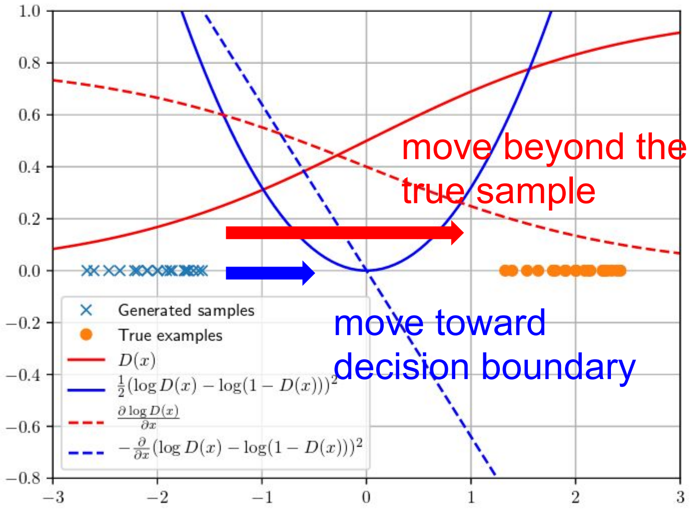
Boundary-Seeking GAN

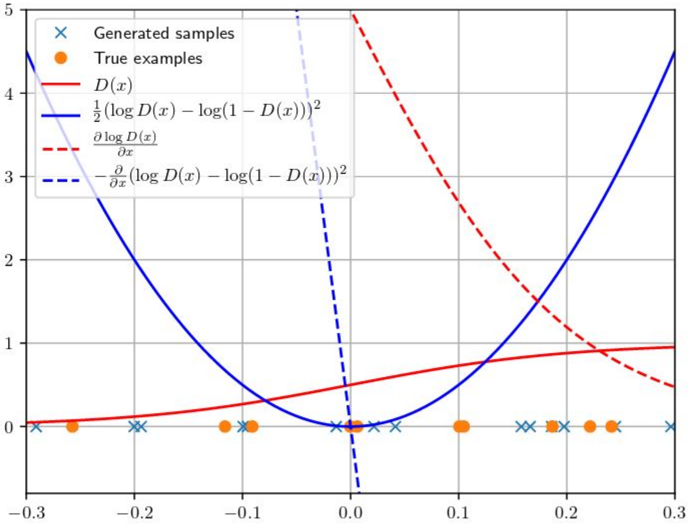
- For Generator

$$\min_G E_{z\sim P_z(z)}[(log(D(G(z)) - log(1 - D(G(z)))^2]$$

- Only change the loss function of G

Boundary-Seeking GAN

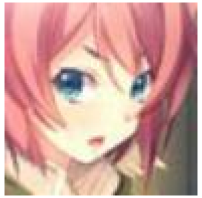


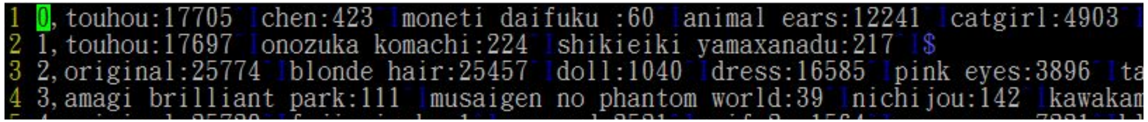


Submission and Grading

Homework 3 package

- Anime Dataset:
  - training data: 33.4k (image, tags) pair
  - faces/, tags.csv, sample_testing.txt
- training tags file format
  - img_id <comma> tag1 <colon> #_post <tab> tag2 <colon> …

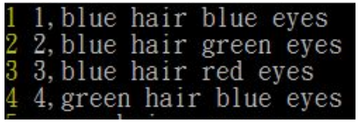




```
0, touhou:17705	chen:423	moneti daifuku :60	animal ears:12241	catgirl:4903
1, touhou:17697	onozuka komachi:224	shikieiki yamaxanadu:217	$
2, original:25774	blonde hair:25457	doll:1040	dress:16585	pink eyes:3896	ta
3, amagi brilliant park:111	musaigen no phantom world:39	nichijou:142	kawakam
```

tags.csv

- testing text file format
  - testing_text_id <comma> testing_text



```
1, blue hair blue eyes
2, blue hair green eyes
3, blue hair red eyes
4, green hair blue eyes
```

sample testing_text.txt

- testing text only includes **'color hair'** and **'color eyes'**, only alphabetic char involved.
- Data download link:
  - https://drive.google.com/open?id=0BwJmB7alR-AvMHEtczZZN0EtdzQ
  - If you want to do something cool beyond generating faces, mail us. We will give you original images

Submission

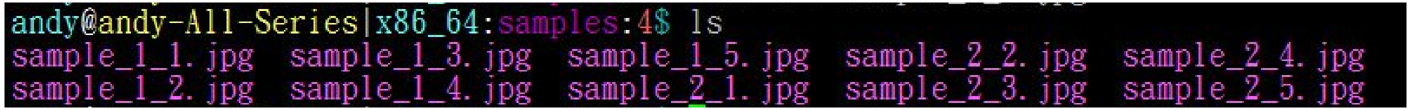
- Only **Python** with **Tensorflow r1.0** for GAN
    - You are allowed to use any tool to process text
    - You are allowed to use any data you collect
- Deadline: **5/25(Thu.) 23:59:59 (UTC+8)**
- **MLDS2017/hw3** should contain following files
    - **run.sh train.py, (pre-)trained_model, generate.py, samples/, report.pdf**
    - If some files are too big, upload to your cloud and download them when running your run.sh
- TAs will run your run.sh to generate images given a text
    - bash run.sh [testing_text.txt]
    - run.sh must output in **10 minutes**.

Submission on Github

- Only one branch **master** is needed
- **master** stores the model by **using GAN structure**
- Remember to put your **pre-trained models or download scripts** so that we can run your code successfully

Output Format Requirement

- The generated images should be in Directory samples/
  - make sure it’s empty before we run your code
- Each generated image must be 64 x 64 in size
  - please resize all training images to 64 x 64 before training the model
- For each input text, you must generate 5 images
- Generated img should be named as **“sample_(testing_text_id)_(sample_id).jpg”**
- Example:



```
andy@andy-All-Series|x86_64:samples:4$ ls
sample_1_1.jpg  sample_1_3.jpg  sample_1_5.jpg  sample_2_2.jpg  sample_2_4.jpg
sample_1_2.jpg  sample_1_4.jpg  sample_2_1.jpg  sample_2_3.jpg  sample_2_5.jpg
```

組別互評

- We will put your generated images in the grading platform
- Link will be sent to your mail after HW deadline
- Answer **2** scores for each image
  - How the image fits the text
  - How the image looks real
- Scores should be integer from 1 to 5
  - 1 to 5 corresspondіng to (super bad, bad, average, good, super good)
- You may score your results, so be fair when your are scoring :)

組別互評

- Separate scores with a comma (score for matching text, score for reality)
- Example:

3 → Gray hair green eyes

4, 5

Ok ✓ press ENTER

What report should cover?

- Environment (1%)
  - Ex. OS, CPU, GPU, Memory, libraries you used and version, etc.
- Model description(3%)
  - Must include model strucuture, objective function for G and D
- How do you improve your performance (5%)
- Experiment settings and observation (5%)
- Team division (1%)
- No more than 5 pages
- Please written in Chinese (unless you don’t know how to type Chinese)

Grading Policy (20%)

- Wrong output format will not be graded
- Report (10%)
- Code (5%)
  - You will be scored only if you use GAN and output results in 10 minutes
- ~~Baseline~~
- 組別互評 (5%)
- Bonus
  - 組別互評 top 3 (5%, 3%, 2%)
  - Generate images beyond faces and show results in report (10%)

Other Policy

- Late policy: 30% off per day late afterwards.
  [Delay form will be announced afterwards]
- No plagiarism is allowed.