Homework
Coupled Generative Adversarial Networks

Vo Hoang Anh and Yong-Guk Kim

vohoanganh89@gmail and ykim@sejong.ac.kr

1 Introduction

Generative Adversarial Nets (GANs) [4] have been applied to various domains such as computer vision [5, 1], natural language processing [8, 9], time series synthesis [3, 7], etc. GANs belongs to generative model in machine learning. The architecture of GANs includes two components, one of which is a discriminator (D) distinguishing between generated images and real images while the other one is a generator (G) creating images to fool the discriminator. GANs is able to generate desired samples, high resolution images, and diversity images. However, GANs must against the two most significant that they are hard to train and they are difficult to evaluate.

In this report, we implemented Coupled Generative Adversarial Networks (CoGAN), proposed by Liu at al [6]. CoGAN learns a joint distribution of multi-domain images instead of one domain as GANs. The source code is available in `https://github.com/VoHoangAnh/Mnist_cogan`.

2 Coupled Generative Adversarial Networks

The main contribution of CoGAN is not reply on training samples drawn from the join distribution, $p_{X1,X1}$, which it requires training samples drawn from the marginal distributions, p_{X_1} and p_{X_2} separately. The objective function $V(D_1, D_2, G_1, G_2)$ of CoGAN is mentioned as follows

$$\begin{aligned} &\max_{G_1,G_2} \min_{D_1,D_2} V(D_1, D_2, G_1, G_2) = E_{\boldsymbol{x_1}\sim p_{\boldsymbol{x_1}}}[-log D_1(\boldsymbol{x_1})] + E_{\boldsymbol{z}\sim p_{\boldsymbol{z}}}[-log(1 - D_1(G_1(\boldsymbol{z})))] \\ &+E_{\boldsymbol{x_2}\sim p_{\boldsymbol{x_2}}}[-log D_2(\boldsymbol{x_2})] + E_{\boldsymbol{z}\sim p_{\boldsymbol{z}}}[-log(1 - D_2(G_2(\boldsymbol{z})))] \end{aligned} \tag{1}$$

Subject to:

$$\boldsymbol{\theta}_{G_1^{(i)}} = \boldsymbol{\theta}_{G_2^{(i)}} \text{ for } i = 1, 2, ..., k$$
$$\boldsymbol{\theta}_{D_1^{(n_1-j)}} = \boldsymbol{\theta}_{D_2^{(n_2-j)}} \text{ for } j = 0, 2, ..., l - 1$$

where n_1 and n_2 are the numbers of layers, k and l denote the number of weight-shared layers in generative model and discriminative model, respectively. CoGAN training is relied on existence of shared high-level representations in the domains. The overview of CoGAN is presented in Figure 1.

In this report, we utilized CoGAN models that were mentioned in [6] with two variant versions, which we called CoGAN-1 and CoGAN-2. The details of CoGAN-1 includes generative and discriminative models are presented in Table 1 and 2. Similarly, we show the generative and discriminative models for CoGAN-2 in Table 3 and 4, respectively. The main difference between CoGAN-1 and

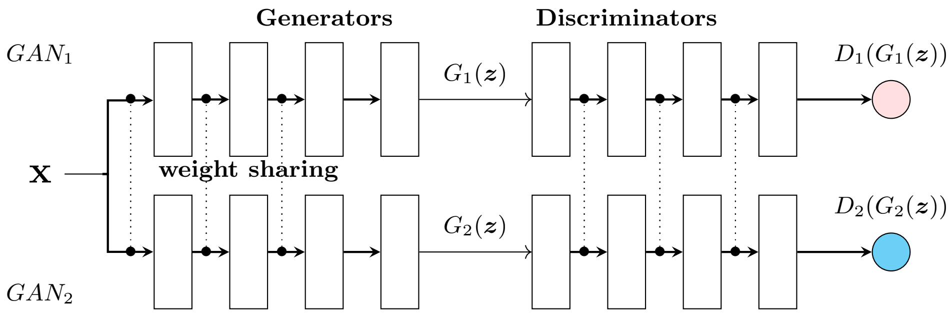


Figure 1: The overview framework of CoGAN in [6]. The weights of the first few layers of the generative models are shared together, while the weights of the last few layers of the discriminative models are also shared to extract high-level features.

CoGAN-2 is the generative models in CoGAN-1 utilizing convolutional transposed 2D layer while fully connected layers are utilized in CoGAN-2.

Generative models			
Layer	Domain 1	Domain 2	Shared
1	Conv2DTransposed(1024, K4×4, S1) BatchNorm2D PReLU	Conv2DTransposed(1024, K4×4, S1) BatchNorm2D PReLU	Yes
2	Conv2DTransposed(512, K3×3, S2) BatchNorm2D PReLU	Conv2DTransposed(512, K3×3, S2) BatchNorm2D PReLU	Yes
3	Conv2DTransposed(256, K3×3, S2) BatchNorm2D PReLU	Conv2DTransposed(256, K3×3, S2) BatchNorm2D PReLU	Yes
4	Conv2DTransposed(128, K3×3, S2) BatchNorm2D PReLU	Conv2DTransposed(128, K3×3, S2) BatchNorm2D PReLU	Yes
5	Conv2DTransposed(1, K6×6, S1) Tanh	Conv2DTransposed(1, K6×6, S1) Tanh	No

Table 1: Generative models architecture of COGAN-1

3 Experiments

3.1 MNIST dataset

MNIST dataset consists of 70,000 handwritten digits images with 10 classes from 0 to 9. The dataset can be downloaded at this website[1]. In this report, we created three datasets from MNIST corresponding 90 degree rotation images, negative images, and edge images, presented in Figure

[1] http://yann.lecun.com/exdb/mnist/

Discriminative models			
Layer	Domain 1	Domain 2	Shared
1	Conv2D(20, K4×4, S1) Maxpool(2)	Conv2D(20, K4×4, S1) Maxpool(2)	No
2	Conv2D(50, K5×5, S1) Maxpool(2)	Conv2D(50, K5×5, S1) Maxpool(2)	Yes
3	Fully Connected Layer(500) PReLU	Fully Connected Layer(500) PReLU	Yes
4	Fully Connected Layer(1) Sigmoid	Fully Connected Layer(1) Sigmoid	Yes

Table 2: Discriminative models architecture of COGAN-1

Generative models			
Layer	Domain 1	Domain 2	Shared
1	Fully Connected Layer(1024) BatchNorm2D PReLU	Fully Connected Layer(1024) BatchNorm2D PReLU	Yes
2	Fully Connected Layer(1024) BatchNorm2D PReLU	Fully Connected Layer(1024) BatchNorm2D PReLU	Yes
3	Fully Connected Layer(1024) BatchNorm2D PReLU	Fully Connected Layer(1024) BatchNorm2D PReLU	Yes
4	Fully Connected Layer(1024) BatchNorm2D PReLU	Fully Connected Layer(1024) BatchNorm2D PReLU	Yes
5	Fully Connected Layer(1) Tanh	Fully Connected Layer(1) Tanh	No

Table 3: Generative models architecture of COGAN-2

Discriminative models			
Layer	Domain 1	Domain 2	Shared
1	Conv2D(20, K4×4, S1) Maxpool(2)	Conv2D(20, K4×4, S1) Maxpool(2)	No
2	Conv2D(50, K5×5, S1) Maxpool(2)	Conv2D(50, K5×5, S1) Maxpool(2)	Yes
3	FC(500) PReLU	FC(500) PReLU	Yes
4	FC(1) Sigmoid	FC(1) Sigmoid	Yes

Table 4: Discriminative models architecture of COGAN-2

2. Specifically, we created the variant datasets based on the simple image processing techniques as

follows

- MNIST dataset for rotation image with $\theta = 90^0$:

$$M = \begin{pmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{pmatrix} \tag{2}$$

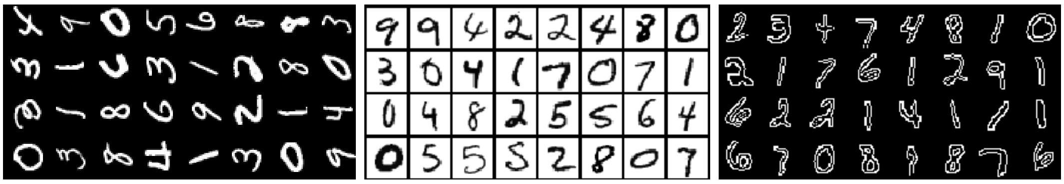
$$\hat{f}(x,y) = M * f(x,y) \tag{3}$$

- MNIST dataset for negative image:

$$\hat{f}(x,y) = 255 - f(x,y) \tag{4}$$

- MNIST dataset for edge image: we utilized Canny method [2] to create the corresponding edge images, which the low threshold λ_1 is 100 and the high threshold λ_2 is 200.

where M is the transformation matrix, $f(x,y)$ denotes the original image and $\hat{f}(x,y)$ denotes the output image from Eq (2) to Eq (4).



(a) 90 degree rotation images. (b) negative images (c) edge images

Figure 2: MNIST dataset with 90 degree rotation image, negative image, and edge image.

3.2 Experiment Setting

In the experiment, we utilized the configuration setting as [6] both generative models and discriminative models, presented Table 5.

Parameter	Value
learning rate	0.0002
optimizer	Adam
β_1	0.5
β_2	0.999
batch size	32

Table 5: The parameters setting for training process

3.3 Experiment results

We performed two main experiments using CoGAN [6] on MNIST dataset and it's perturbation datasets such as 90 degree rotation, edge detection, and negative images.

The first experiment presents the comparison of the generated images between CoGAN-1 and

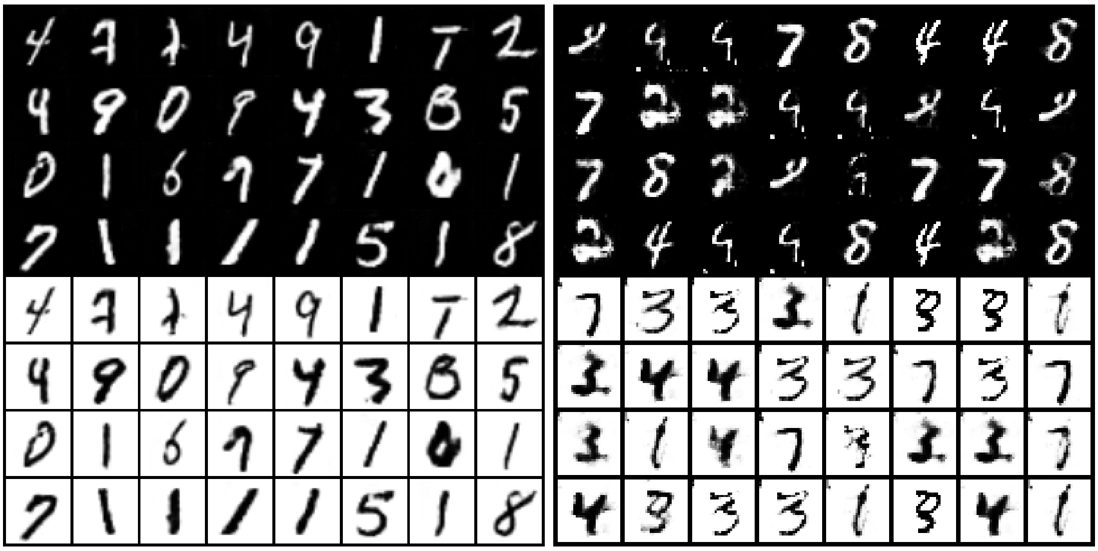
(a) Generated images using CoGAN-1

(b) Generated images using CoGAN-2

Figure 3: The comparison of generated images (the orginal MNIST and 90 degree rotation images) between CoGAN-1 and CoGAN-2

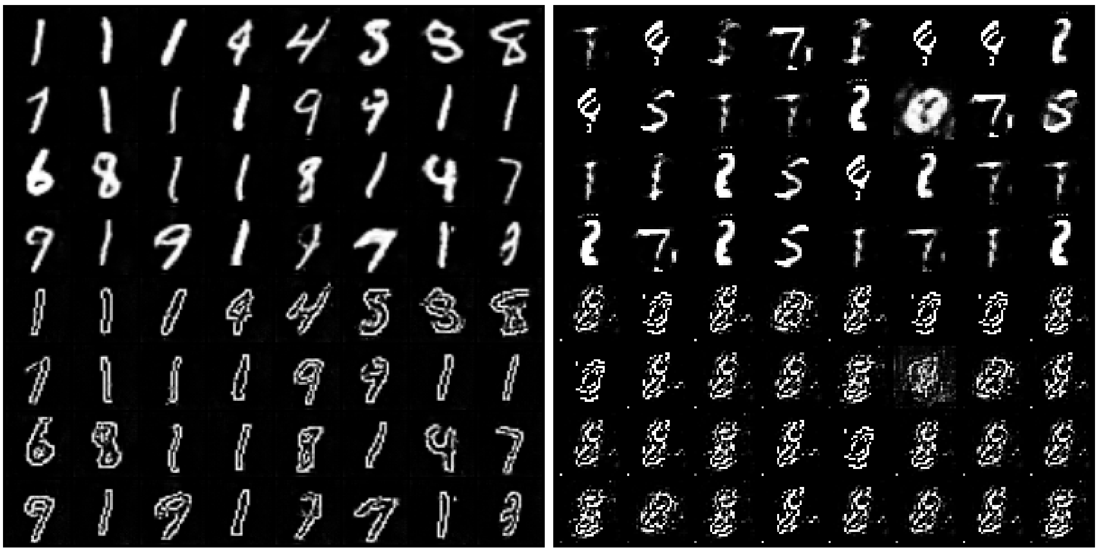
CoGAN-2. Figure 3 and 4 indicate that the generated images using CoGAN-1 and CoGAN-2 are the meaning, however, some generated images by CoGAN-1 look better than CoGAN-2. In cases of the edge images, we are easy to recognize that CoGAN-1 is better CoGAN-2 because the generated images using CoGAN-2 have been occurred mode collapse issue, when generative model creates one or a small subset of different outcomes.

The second experiment evaluates the influence of the data distribution to CoGAN training process. Therefore, we performed three variant cases: rotation and edge images, rotation and negative image, and edge and negative images rather than the original MNIST and it's perturbation as [6]. Figure 6 shows the generated images using CoGAN-1 in all cases. It also indicates that the combination of rotation and edge images is not better than the rest of cases.



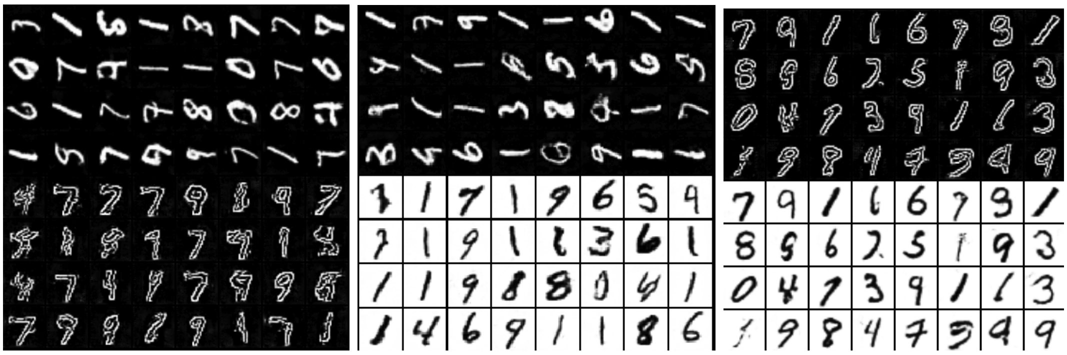
(a) Generated images using CoGAN-1 (b) Generated images using CoGAN-2

Figure 4: The comparison of generated images (the orginal MNIST and negative images) between CoGAN-1 and CoGAN-2



(a) Generated images using CoGAN-1 (b) Generated images using CoGAN-2

Figure 5: The comparison of generated images (the orginal MNIST and edge images) between CoGAN-1 and CoGAN-2.



(a) Generation of rotated and edge images
(b) Generation of rotated and negative images
(c) Generation of edge and negative images

Figure 6: The comparison of other cases: a) rotation and edge images b) rotation and negative images c) edge and negative images using CoGAN-1

4 Conclusion

In this report, we implemented CoGAN [6] with two different versions to generate hand digits using MNIST dataset and it's perturbation such as 90 degree rotation, edge detection, and negative images. The first one is called CoGAN-1 with convolutional transposed 2D in the generative models and the second one is named CoGAN-2 with fully connected layers instead of convolutional transposed 2D.

The experimential results indicated that the generated images using CoGAN-1 look the meaningful images and realistic digit images than CoGAN-2. In addition, the data distribution between the first domain and the second domain has a small impact to the CoGAN training process.

References

[1] Yang C, Lu X, Lin Z, Shechtman E, Wang O, and Li H. High-resolution image inpainting using multi-scale neural patch synthesis. In *Advances in Neural Information Processing Systems*, pages 6721–6729, 2017.

[2] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986.

[3] Cristóbal Esteban, Stephanie L. Hyland, and Gunnar Rätsch. Real-valued (medical) time series generation with recurrent conditional gans. *CoRR*, abs/1706.02633, 2017.

[4] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[5] Ma L, Jia X, Sun Q, Schiele B, Tuytelaars T, and Van Gool L. Pose guided person image generation. In *Advances in Neural Information Processing Systems*, pages 406–416, 2017.

[6] Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[7] Jinsung Yoon, Daniel Jarrett, and Mihaela van der Schaar. Time-series generative adversarial networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[8] Dai Z, Yang Z, Yang F, Cohen W, W, and Salakhutdinov R, R. Good semi-supervised learning that requires a bad gan. In *Advances in Neural Information Processing Systems*, pages 6510–6520, 2017.

[9] Yang Z, Hu J, Salakhutdinov R, and Cohen W. Semi-supervised qa with generative domain-adaptive nets. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1040–1050, 2017.