IMPROVING EFFICIENCY OF LOSS FUNCTION IN CYCLE-CONSISTENT ADVERSARIAL NETWORKS

[1]Jarda Aamir A, [2]Priyang Bhatt

[1]Department of Computer Engineering, G.H Patel College of Engineering and Technology, Gujarat, India.,
[2]Assistant Professor, Department of Computer Engineering, G.H Patel College of Engineering and Technology, Gujarat, India.

Abstract: **Unpaired image to image translation is a most interesting and challenging topic because of graphic problem and loss function. Through this paper, we aim to understand what exactly Cycle GAN are, what the existing applications of such models are and how we improve the loss function. Scenario is we give input is an image and the output is different version of this input image that is changed according to our guideline. The basic goal of this translation is to learn mapping between input image and output image using training set. Initial approach for translate an image from X to Y in the absence of paired data. Model contain two functions G: X →Y and F: Y → X where X is source domain and Y is target domain. The first things we should know about this technique is that it uses Generative adversarial network. In this we have two neural network battling each other. Generator tries to create realistic image and discriminator which tries to learn difference between real and fake image. A cycle consistency loss function is introduced to the optimization problem that means if we convert a zebra image to a horse image and then back to a zebra image, we should get the very same input image back.**

Index Terms **– Generative adversarial network, CycleGAN, L1 loss, Logistic regression, Adam Optimizer**

I. INTRODUCTION

First of all we need to know about Generative adversarial network (GAN). A generative adversarial network (GAN) is a type of model in neural network that offer a lots of potential in the world of machine learning. In GAN there are two neural network: first is generative network and second is discriminative network. So main concept behind this project is generative adversarial network. GAN is about creating stuff and this is hard to compare other deep leaning field. Main focus of GAN is to generate data from scratch. Example of GAN is, generate a zebra from a horse. As we see early GAN composes of two network the generator and the discriminator. GAN are deep neural network architectures which two network counter one against the other.

GANs were introduce in a paper by Ian Goodfellow and other researcher in 2014. Yann LeCun called this concept "the most interesting idea in the last 10 years in ML." GANs' potential is very huge, because they can learn to mimic any data. So use of GAN we create worlds similar to our own in any domain: image, anime, news anchor, speech.

Brief Working of GAN Model,
As we know these algorithms belong to the field of unsupervised learning [2]
Generative Adversarial Networks are composed of two models:

- First model is called a Generator and it target to generate new data similar to real one. Generator can create data and discriminator is check whether the data is real or fake.
- And second model is called a Discriminator. This model's goal is to recognize if an input data is real or fake – belongs to the original dataset- or if it's fake generated by generator. So discriminator is like a police which tries to detect work is real or fake.

Many GAN Models suffer the huge problem, for example

- Sometime model can't combine Non-convergence.
- The generator collapses that's why produce limited verities of sample data
- Unbalance between discriminator and generator causing over fitting data
- Highly sensitive to the hyper parameter selections.

II. CYCLEGAN

The Cycle-GAN architecture was proposed in the paper, Unpaired Image-to-Image Translation Cycle-Consistent Adversarial Networks. Jan-Yan Zhu and his colleagues (2017) [1] suggested: As PIX2PIX can produce truly amazing result but challenge is in training data. So convert one image to other image is bit difficult as well as time consuming, infeasible or even sometime impossible based on which image we were trying to translate(one to one match between two images). This is where CycleGAN comes in. The key idea behind CycleGAN is that they allow you to point the model at two unpaired collection of the images. For Example, one collection of images, Group A would be Zebra while other group is B would be Horse. So Cycle-GAN model can learn to translate unpaired image.

So the main part of this paper is here Cycle-consistency loss like if our input image is A from domain X is transformed into a target image or output image B from domain Y via Generator G, then image B from domain Y is translated back to domain X via Generator F. So that time the difference between these two images is called as the Cycle-Consistency loss.

This approach requires creating two pairs of generators and discriminators: one for A2B (source to output conversion) and another for B2A (output to source conversion). Cyclic loss and loss function are very important part in this whole scenario. We use L1 loss function for both generator and Logistic loss for Discriminator.

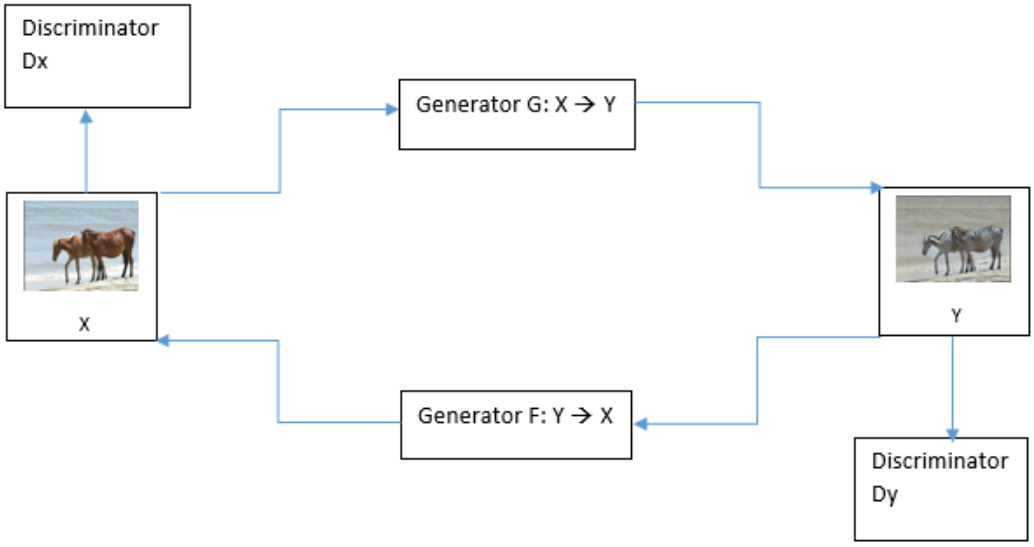

Figure 1 Two mapping Function G: X → Y and F: Y→ X and two Discriminator Dx and Dy

III. Related Work

Jun-Yan Zhu, Taesung Park, Phillip Isola and Alexei A. Efros [1] presented Cycle GAN. They contain two generator (G, F) and two discriminator (Dx, Dy). In this paper author is also find efficient way to achieve loss function. They used Generative adversarial model to perform operation like image translation, Style transfer. PIX2PIX can produce truly amazing result but challenge is in training data convert one image to other image is bit difficult as well as time consuming. Some problem are caused by distribution characteristic of the training datasets. (Exp. Person riding a horse).

Ian Goodfellow [2] were introduce GAN in 2014. GAN contain two neural network first is generative network and second is discriminative network. GANs include a generator and a discriminator, both prepared under the antagonistic learning thought. To train the network, the loss function is defined as

$$\text{Min G Max D } Ex \in \chi [\log D(x)] + Ez \in Z [\log (1\text{-}D (G (z))] \quad (1)$$

Mirza and Osindero [3] presented Conditional GAN, basically in CGAN the generator G and discriminator D are molded on some additional data c. So this data could be text, class label or word. CGANs also provide the information of data to be generated, while original GANs do not provide such type of information. It's very popular for image fusion and image editing.

Ledig et al. introduce Super-Resolution Generative Adversarial Network (SRGAN) [4] this GAN takes a low resolution image as input and generate image with 4* resolution. In this GAN generator use deep convolutional network with resnet block. There are two loss adversarial loss and a feature loss. Feature loss is computed between ground truth and generated image. Working same as a DCGAN.

Radford et al proposed deep convolutional Generative adversarial network (DCGAN) [5]. In this paper author used convolutional neural network for both generator and discriminator. Author used LeakyReLu activation function instead of ReLu function in discriminator for all layer (because of negative value) and generator used ReLu function for all other layer. Also removing fully connected hidden layer. DCGAN is more stable than other GAN and its produce higher quality image that's why DCGAN widely used in many application.

Huanget al. introduce Face Conditional Generative Adversarial Network (FCGAN) [6], FCGAN is focused on facial image super resolution. Within the architecture both generator and discriminator utilize an encoder-decoder alongside skip relations. FCGAN working is same as SRGAN both are used for image super-resolution.

Yongqi Zhang presented XOGAN [7]: One-to-Many Unsupervised Image-to-Image Translation. In this paper author exhibit aims at learning the relationship between samples from two image domains without supervised information. Author introduce an additional variable to control the variations in one-to-many mapping, XO-structure, called the XOGAN. So we cannot only learn to translate between the two domains, but we can also handle the translated image with additional variations.

IV. Dataset

- Dataset is Collected from various website like Imagenet, CMP Facades dataset, Cityscapes training set, Google Maps, Flicker, Wikiart.
- Orange to Apple (Orange -> Apple) and Apple to Orange: Collected from ImageNet
- Horse to Zebra (Horse -> Zebra) and Zebra to Horse: Collected from ImageNet
- Landscape photo -> Monet painting style, landscape photo -> Van Gogh painting style, landscape photo -> Ukiyo-e painting style: Collected from Flicker
- Monet to Photo: Collected From Flicker
- Cityscapes Photo to Label and Cityscapes Label to Photo: Collected from Cityscapes dataset
- Map to Aerial photo and Aerial photo to Map: Collected from Google Maps
- IPhone photos of flowers -> DSLR photos of flowers: Collected from Flicker

V. PROPOSED METHOD

As we know two networks are called "Generator" and "Discriminator". The Generator's goal is to produce data that is indistinguishable from the training dataset and try to fool discriminator. The Discriminator's goal is to correctly determine whether a particular example is real (i.e., coming from the training dataset) or fake (i.e., created by the Generator). In GAN there are two network [8] but in Cycle-GAN there are four network 2 Generator and 2 discriminator. Model includes two generator G : X → Y and F : Y → X and we have two discriminator Dx and Dy, Where Dx is try to distinguish between input image (X) and translate image in the same manner Dy work exactly same as Dx. We have two type of loss, adversarial loss and cycle consistency losses.

Overview of CNN

Basically CNN is mostly used for image classification or recognition. CNN makes function more efficient and reduce the parameter. A CNN accept pixel value as input in network. The hidden layer for Feature extraction and last fully connected layer recognize the image fall in which category? There are 4 layer in CNN. Convolution layer, ReLu layer (Activation Function), pooling layer and fully connected layer.

The Convolution layer uses a filter matrix over the image or shift filter matrix over image and obtain Convolved feature map. Next layer is ReLu layer. It convert all negative pixel to zero and perform operation but in CycleGAN we used LeakyReLu because of negative value. Output is called Rectified Feature map. Next layer is pooling layer. Pooling layer is reduces the dimensionality of rectified Feature map, means we have to choose max pixel from the image. Then pooled feature map converted into long Continuous LV. This whole process is called Flattering. Flattering Connected with fully connected layer to classify the image.

Building the Generator

Building Generator is not actually that much easy because we have encoding part, transformation and decoding part. First step is extracting the feature from an image. In encoding part we used three convolution layer for extracting maximum feature. In first layer size of filter window is 7 and stride is 1 and other two layer size of windows is 3 and stride is 2. Every convolution layer prompts extraction of continuously more elevated amount of highlights. We can also add some other layer like LeakyReLu or normalization.[9]

Now second part is transformation. You can see that these layer collecting nearby feature of an image and we used 9 resent block for achieving good result. Resent block is nothing but the store previous layer input for later layer so our result or output don't differ from the input image, that's why resent network are too good for this type scenario. Now third part is decoding. Decoding step is exact adverse of Encoding

Building the Discriminator

Now we need to create discriminator for image prediction. What discriminator do? Discriminator would take an image as input an constantly try to predict if it is a real image or generator image, discriminator work is simply like Convolution neural network to classify the image. First we will extract maximum feature from the input image or converted image and after deciding these feature fit for which category.

Adversarial Loss

We have two discriminator and two generator so what we have to do we have to design loss for all network for better results. Here are some steps about loss function. [1]

Discriminator confirm all the original image of the relating classification.
Discriminator reject image which generator are used to fool them.
Generators force the discriminators to approve all the produced pictures, to trap them.

We apply this loss for both generator and its discriminator for example G: X→Y and Dy as well as F: Y→X and discriminator Dx.

$$LGAN (G, Dy, X, Y) = Ey \sim pdata(y) [logDy(y) + Ex\sim pdata(x) [log (1- Dy (G (x))] \quad (2)$$

Base on this equation G tries to generate image which is very similar to Y while discriminator (Dy) is try to figure out between real image and translated image. Instead of minimize G we have to maximize (G) objective for better result. We introduce same for the other Generator and discriminator.

Cycle-Consistency Loss

If we convert X image into Y using mapping function G and get back to X using mapping function F, difference between this two image original image and Cyclic image should be very limited.

Horse image (X) → Zebra Image (G(x)) → Horse Image f (G(x)) ≈ Original image (x)
Zebra image (Y) → Horse Image (G(y)) → Zebra Image f (G(x)) ≈ Original image (y)

Figure 2 Image to image translation - Horse to Zebra and get back to horse

We call this forward Cycle-Consistency and Backward Cycle-Consistency. For improving efficiency for loss function we used L1 loss for both generator network and logistic loss for Discriminator. Discriminator work is just characterize picture is genuine or counterfeit.

$$Lcyc\ (G, F) = Ex\sim pdata(x)\ [\|\ F\ (G(x))-x\ \|\ 1] + Ey\sim pdata(y)\ [\|\ G\ (F(y))-y\ \|\ 1] \quad (3)$$

Total loss for both Generator and Discriminator

We have to add all loss, mapping function G and F and both discriminator Dx and Dy as well as Cyclic-loss with multiplying λ on equation. Cyclic-loss are very important for our case that's why they are multiplied by constant lambda.

$$L\ (G, F, Dx, Dy) = LGAN\ (G, Dy, X, Y) + LGAN\ (F, Dx, Y, X) + \lambda\ Lcyc\ (G, F) \quad (4)$$

Where λ controls the general significance of the two objective. In addition, we contain history of last 100 images to train the discriminator why we doing all this stuff because both generator and discriminator can overfit themselves (example: Putin on horse) that lead to mode collapse. Using of this discriminator will not be forward in defeating the generator. It needs to beat the last 100 generators to make an optimum solution

L1 Loss

Least absolute deviations (L1) is a Classic loss functions, that determine which operation or role should be minimized while learning from a dataset. So basically this machine learning function are used to minimize the error. In our case L1 loss function minimize the difference between the true value and the predicted value.

$$L1 = \sum_{i=1}^{n} |y\ true - y\ predicted| \quad (5)$$

Adam Optimizer

Optimization Algorithms are obviously used to update bias and weight after each iteration. Basically they can divide into two categories:

1. Constant Learning Rate Algorithms
2. Adaptive Learning Algorithms

In Constant learning rate algorithms learning rate do not change after each iteration best example of this algorithms is Stochastic Gradient. In Adaptive Learning Algorithms learning rate change after iteration example: Adagrad and Adam.

Working with optimization functions:

1. Stochastic Gradient Decent
2. Adagrad
3. Adam

We used stochastic gradient but cannot achieve best result after we moving to Adam and RMSProp. Adam is a very popular algorithm in the field of deep learning because it achieve good result fast. Most of deep learning research used this optimization for perfect result.

Adam works well in practice, is faster, and outperforms other techniques. Stochastic Gradient Decent was much faster than the other algorithms but the results produced were far from optimum.

Just like in the original GAN implementation, we'll create individual optimizers which can only update certain parts of the network. We'll do the same thing here, except now we actually have 3 networks to optimize, and so we'll need 3 optimizers:

- G_xy and G_yx variables will be optimized as the generator,
- While Dx and Dy, should update two different discriminators.

The learning rate was set to 0.0002 for the first half of training, and then linearly reduced to zero over the remaining iterations.

VI. Result

First we perform operation like horse to zebra, orange to apple, Monet to Photo using method which we define earlier in our study then we compare our approach against recent method of image to image translation and other paper like CoGAN[10], BiGAN/ALI[11], pix2pix[12], Feature loss + GAN [13], pixel Loss + GAN[12], SimGAN[13]

CoGAN [10] in this method there are two generator one is for domain X and another one is for Domain Y. Coupled generative adversarial network (CoGAN) for learning a joint distribution of multi-domain images.

BiGAN/ALI [11] learn a generator G: X → Y this method also perform inverse mapping function like F: Y → X. They were initially intended for mapping an idle vector z to an image x, we executed a similar goal for mapping a source image x to an objective image y. [1]

Feature loss + GAN [13] we also compare with FL + GAN in this paper L1 loss is estimate over deep image, and the rest of the functionality is same as other GAN operation

SimGAN [13] Like Jun-Yan Zhu paper method Shrivastava et al. also uses an adversarial loss to train and perform operation like (X to Y).

Loss	Per-pixel acc.	Per-class acc.	ClassIOU
CoGAN	0.40	0.10	0.06
BiGAN/ALI	0.19	0.06	0.02
SimGAN	0.20	0.10	0.04
CycleGAN	0.52	0.17	0.11
Improved CycleGAN(ours)	0.54	0.20	0.14

Table 1: FCN-scores for different methods, evaluated on Cityscapes labels → photo.

Loss	Per-pixel acc.	Per-class acc.	ClassIOU
CoGAN	0.45	0.11	0.08
BiGAN/ALI	0.41	0.13	0.07
SimGAN	0.47	0.11	0.07
CycleGAN	0.58	0.22	0.16
Improved CycleGAN(ours)	0.60	0.24	0.19

Table 2: Classification performance of photo→labels for different methods on cityscapes.

Loss	Per-pixel acc.	Per-class acc.	ClassIOU
Cycle Alone	0.22	0.07	0.02
GAN Alone	0.51	0.11	0.08
GAN + Forward Cycle	0.55	0.18	0.12
GAN + Backward Cycle	0.39	0.14	0.06
CycleGAN	0.52	0.17	0.11
Improved CycleGAN(ours)	0.56	0.18	0.12

Table 3: Ablation study: FCN-scores for different variants of our method, evaluated on Cityscapes photo→labels

Loss	Per-pixel acc.	Per-class acc.	ClassIOU
Cycle Alone	0.10	0.05	0.02
GAN Alone	0.53	0.11	0.07
GAN + Forward Cycle	0.49	0.11	0.07
GAN + Backward Cycle	0.01	0.06	0.01
CycleGAN	0.58	0.22	0.16
Improved CycleGAN(ours)	0.60	0.25	0.18

Table 4: Ablation study: Classification performance of photo→labels for different losses, evaluated on Cityscapes.

Input A Input A - > B Input B - > A

Input A Input A - > B Input B - > A

Input A Input A - > B Input B - > A

Input A Input A - > B Input B - > A

Input A Input A - > B Input B - > A

Input A Input A - > B Input B - > A

Input A Input A - > B Input B - > A

VII. Application

- Resurrecting Ancient Cities: Convert ancient maps of Babylon (Iraq), Jerusalem and London into modern Google Maps and satellite views. Also convert satellite view.
- Animal adjustment: Translate black bears to pandas.
- Portrait to Doll face: Game of throne character looks like doll.
- Face ↔ Ramen: performed a magical and hilarious Face ↔ Ramen translation with CycleGAN.
- Colorizing legacy photographs: turn black & white photos into colour versions.
- Converting Fortnite into PUBG: using CycleGAN we also convert Fortnite graphic into PUBG graphic and same PUBG graphic into Fortnite graphic, this two are most trendy Battle Royale games with millions of users. Using this CycleGAN we can enjoy the visuals of each other.
- Converting Monet into Thomas Kinkade: Using CycleGAN convert Monet photo into different style
- Colorizing legacy photographs: Mario Klingemann trained CycleGAN to convert black and white photo into colour versions.
- The Electronic Curator: Eran Hadas and Eyal Gruss used CycleGAN to convert human faces into vegetable portraits. Eran and Eyal built a System which allow users to interact with the system with own faces.

VIII. Conclusion

We all know Cycle-GAN are one of the newest and interesting methods. Cycle-GAN is an open field for research sill lots of work can be done in this field. From literature survey we understand lots of things as well as we observe some limitation like training time, Cycle loss, sometime

it's very complex to control the synchronization of the two networks that time training process not stable, etc. Then we see gradient-based optimization algorithm can be used to train the GAN. Accuracy of model can be improved by proper optimization function and loss function. We used Adam optimization function and L1 loss function to achieve good result.

REFERENCES

[1] Jun-Yan Zhu, Taesung Park, Phillip Isola and Alexei A. Efros," Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Network", arXiv:1703.10593v5 [cs.CV] arXiv – 2018

[2] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio," Generative Adversarial Networks", arXiv:1406.2661v1- 2014

[3] Mehdi Mirza, Simon Osindero," Conditional Generative Adversarial Nets", arXiv: 1411.1784 [cs.LG] – 2014

[4] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, Wenzhe Shi," Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network", arXiv:1609.04802 [cs.CV] – 2016

[5] Alec Radford, Luke Metz, Soumith Chintala," Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks", arXiv: 1511.06434 [cs.LG] – 2016

[6] Huang Bin, Chen Weihai, Wu Xingming, Lin Chun-Liang," High-Quality Face Image SR Using Conditional Generative Adversarial Networks", arXiv: 1707.00737 [cs.CV] – 2017

[7] Yongqi Zhang,"XOGAN: One-to-Many Unsupervised Image-to-Image Translation", arXiv: 1805.07277v1 [cs.CV] 18 May 2018

[8] Basic of GAN, "https://skymind.ai/wiki/generative-adversarial-network-gan"

[9] Hardik Bansal on GAN,"https://hardikbansal.github.io/CycleGANBlog/"

[10] Ming-Yu Liu, Oncel Tuzel," Coupled Generative Adversarial Networks" arXiv: 1606.07536 [cs.CV] – 2016

[11] Jeff Donahue, Philipp Krähenbühl, Trevor Darrell," Adversarial Feature Learning", arXiv: 1605.09782 [cs.LG] – 2017

[12] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, Alexei A. Efros," Image-to-Image Translation with Conditional Adversarial Networks", arXiv:1611.07004 [cs.CV] – 2018

[13] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Josh Susskind, Wenda Wang, Russ Webb," Learning from Simulated and Unsupervised Images through Adversarial Training", arXiv:1612.07828 [cs.CV] – 2017