A SEMINAR REPORT ON

UNSUPERVISED ARTISTIC IMAGE GENERATION USING GENERATIVE ADVERSARIAL NETWORKS

SUBMITTED BY

Param Bhavsar
Roll No.: 3213
Class: TE-2

Under the guidance of
Prof.P. S. Vidap

DEPARTMENT OF COMPUTER ENGINEERING
Pune Institute of Computer Technology
Dhankawadi, Pune
Maharashtra 411043

DEPARTMENT OF COMPUTER ENGINEERING
Pune Institute of Computer Technology
Dhankawadi, Pune
Maharashtra 411043

CERTIFICATE

This is to certify that Mr.Param Bhavsar, Roll No.3213 a student of T.E. (Computer Engineering Department) Batch 2017-2018, has satisfactorily completed a seminar report on "UNSUPERVISED ARTISTIC IMAGE GENERATION USING GENERATIVE ADVERSARIAL NETWORKS" under the guidance of Prof.P. S. Vidap towards the partial fulfillment of the third year Computer Engineering Semester II of Pune University.

Prof. P. S. Vidap
Internal Guide
PICT, Pune

Dr. R. B. Ingle
Head
Department of Computer Engineering
PICT, Pune

Place: PICT, Pune
Date: / /2018

ACKNOWLEDGEMENT

We sincerely thank our Seminar Coordinator Prof. A. A. Jewalikar and Head of Department Dr. R. B. Ingle for their support.

We also sincerely convey our gratitude to our guide Prof. P.S. Vidap, Department of Computer Engineering for his constant support, providing all the help, motivation and encouragement from beginning till end to make this seminar a grand success.

Above all we would like to thank our parents for their wonderful support and blessings, without which we would not have been able to accomplish our goal.

Contents

List of Figures

1 ABSTRACT

Unsupervised learning is the type of machine learning algorithm in which learning dataset is done without using the data labels. The Unsupervised learning problem is difficult to tackle in machine learning. Artistic Generative Adversarial Networks (GANs) are used to generate the image from high feature training data by competition between the Discriminative network and Generative network. Artistic GAN learns the data distribution of random data input to generate artistic images. It tries to achieve the capability obtained by Supervised learning algorithms. The gap between the success of Convolutional Neural Network (CNN) for Supervised and Unsupervised learning can be narrowed down.
Keywords: CNN, KL Divergence, GAN, Image Generation, deep neural network,generative algorithm, unsupervised, Natural image generation, high dimension feature learning.

2 INTRODUCTION

2.1 History

Unsupervised learning is an approach to design a machine learning system that has become very popular over last several years. It can be described as the general problem of extracting value from the unlabelled data. The process of Unsupervised Learning is kind of similar to how human learns and acquire knowledge through the experience. Even though machine work in a dark, it somehow manages to extract features and patterns from the probability distribution of data.

A generative model is a powerful way of learning any kind of data distribution using unsupervised learning and achieved tremendous success in last few years. All generative models aim at learning true data distribution of training set so as to generate new data points with some variations. But it is not always possible to learn the exact distribution of given data so it tries to model a distribution which is as similar as possible to true data distribution.

2.2 Literature Review

Most commonly used and efficient generative model approach is Generative Adversarial Networks (GANs)[1]. GANs consist of two models Generative and Discriminative. The training of these two models is occurring simultaneously. A generative model that captures the data distribution and the discriminative model establish the probability that the sample comes from the sample data rather than the generative model. The training process for the generative model is to maximize the probability of discriminative model making mistake. This framework corresponds to the minmax two-player game. In this minmax game, the generative model is pitted against a discriminative model that learn to determine the where the data is from model distribution or data distribution. In GANs the generative models can be thought of an analogous to a term counterfeiter, trying to produce fake currency and use it without detection, while the discriminative model is analogous to the police try to detect the counterfeit currency. Competition in this game helps both models to improve their method until counterfeits are indistinguishable.

Recently, GANs have shows significant result in synthetically generates the nature images using the MNIST, CIFAR-10 and CUB-200 datasets[2]. We could notice that all there datasets have common characteristics such as most of the foreground and the background of images are easily distinguishable[7], each image contain one object per image and the object has finally a structured shape such as number, bird, car etc.

2.3 Motivation and Challenges

This work like to investigate if the model can produce artwork images which do not hold above discussed characteristic. The artwork is very hard to learn as it is difficult to distinguish between the foreground and background of the image[9]. Also, the artwork does not follow the natural structure of object it might be different based on the taste of the artist. This makes the learning of the feature and generates the artwork images difficult. The artwork generation helps not only to generate the artistic images but also help to learn the high feature images. There is some approach to the generation of artwork based on the Conditional GAN [12] but this work try to generate the artistic image without any condition or supper vision.

2.4 Applications

The artistic image generation can be applied to the not only generate the artistic images but also applied to other application such as generating different clothing style, different car design, 3D model generation and any field of fashion and design industry where art is included and artistic nature required to generate new design. This can also be used in the field of style transfer. Remove the distortion and generate the distortion free image. Distortion can be snow, rain, low light in case of self driving car so that the car knew the surrounding environment and help the car to navigate using distortion free generated image.

In this work of unsupervised artistic images generation study of the appropriate architecture for the better quality of image generation than that of Deep Convolutional Generative Adversarial Network [2] which is state of the art for the unsupervised image generation. The new architecture consists of clipping the weight between -0.01 to 0.01 of the discriminative model while training so that the discriminative does not learn fast and try to always reject the sample from the generative model. Also, introduce random noise in the input of discriminative model so that the learning of discriminative model on the more robust dataset.

3 SURVEY OF MATHEMATICAL MODELS

3.1 Kullback-Leibler Divergence

Generative models estimate real data probability distribution $p_{data}(x)$ with real samples, they can be thought as maximizing the likelihood of real samples with parameter θ. With an assumption of independent and identically distributed m training samples x_i where i ϵ $(1, 2, 3, ..., m)$, generative model can formulated as $argmax_\theta \prod_{i=0}^{m} p_\theta(x^i)$ where $p_\theta(x)$ stands for a probability distribution of generated samples with parameter θ, We need to find an optimal parameter θ^* for maximizing likelihood. Therefore argmax is used instead of max. Intuitively, such a generating process can be thought of minimizing a certain type of distance between $p_{data}(x)$ and $p_\theta(x)$. Importantly, it can be shown that maximizing the log likelihood is equivalent to minimizing Kullback-Leibler Divergence (KLD) between $p_{data}(x)$ and $p_\theta(x)$ as the number of samples m increases. The relationship is derived as follows:

$$\theta^* = argmax_\theta \lim_{m\to\infty} \frac{1}{m} \sum_{i=0}^{m} \log p_\theta(x^i) \tag{1}$$

$$\theta^* = argmax_\theta \int p_{data}(x) \log p_\theta(x) dx \tag{2}$$

$$\theta^* = argmax_\theta \int -p_{data}(x) \log p_\theta(x) + p_{data}(x) \log p_{data}(x) dx \tag{3}$$

$$\theta^* = argmax_\theta \int p_{data}(x) \log \frac{p_{data}(x)}{p_\theta(x)} \tag{4}$$

$$\theta^* = argmax_\theta \; KLD(p_{data} || p_\theta) \tag{5}$$

Equation (2) established by the central limit theorem in that the m increases, variance of the expectation distribution decreases. Equation (3) can be introduced because $p_{data}(x)$ does not depend on θ and equation (5) follow from the definition of KLD. Intuitively, maximizing number of real training data because the minimum of KDL between two distributions can be interpreted as approximating $p_{data}(x)$ with large number of real training data because the minimum of KLD is achieved when $p_{data}(x) = p_\theta(x)$. However, we need the model distribution $p_\theta(x)$ to explicitly calculate KDL and more importantly, KDL is not defined when x is outside the support of $p_\theta(x)$.

3.2 Generative Adversarial Networks

GAN, which is composed of two components, the generator G and discriminator D. G produces fake samples from the latent variable z whereas D takes both fake samples and real samples and decides where input is real or fake. D produces higher probability as it determine its input is more likely to be real. G and D oppose each other to achieve their individual goals, so the adversarial term is coined. When the adversarial situation is formulated as the object function, GAN solve the minmax expression with parameterized networks G and D. $p_{data}(x)$ and $p_z(z)$ in equation (6) denote a real data probability distribution defined in the data space χ and a probability distribution of latent space z define on the latent space $\mathbb{Z}$. It should be noted that G maps the latent variable z from $\mathbb{Z}$ into the element of χ, whereas D takes an input x and distinguishes where x comes from real samples or from G. The objective function is thus represented as follows:

$$min_G max_D V(G, D) = \mathbb{E}_{x \sim p_{data}}[log D(x)] + \mathbb{E}_{z \sim p(z)}[log(1 - D(G(z)))] \quad (6)$$

Where V(G, D) is a binary cross entropy function is commonly used in binary classification problems

As D wants to classify real or fake samples, V(D, G) is a natural choice for an objective function. From D's perspective, if a sample comes from G, D will minimize its output thus, the log(1 - D(G(z))) term is naturally derived in equation (6). Meanwhile, G wants to deceive D so it tries to maximize D's output when a fake sample is presented to D. Consequently, D tries to minimize V(G, D), and which is where the maximum relationship in equation (6) comes from.

Theoretically, assuming that models of G and D both have sufficient capacity, the Nash equilibrium for equation (6) can be achieved through following procedure. First, D is trained to obtain an optimal discriminator for a fixed G, and then G tries to fool D to enable D(G(z)) produce a high probability. By iteratively optimizing such minmax problem, D cannot discriminate whether its is real or fake anymore because $p_{data}(x) = p_\theta(x)$ has been achieved, and D(x) produces a probability of $\frac{1}{2}$ for all real and fake samples. Particularly, solving the equation (1) is equivalent to minimizing the Kullback-Leibler divergence between $p_{data}(x)$ and $p_\theta(x)$.

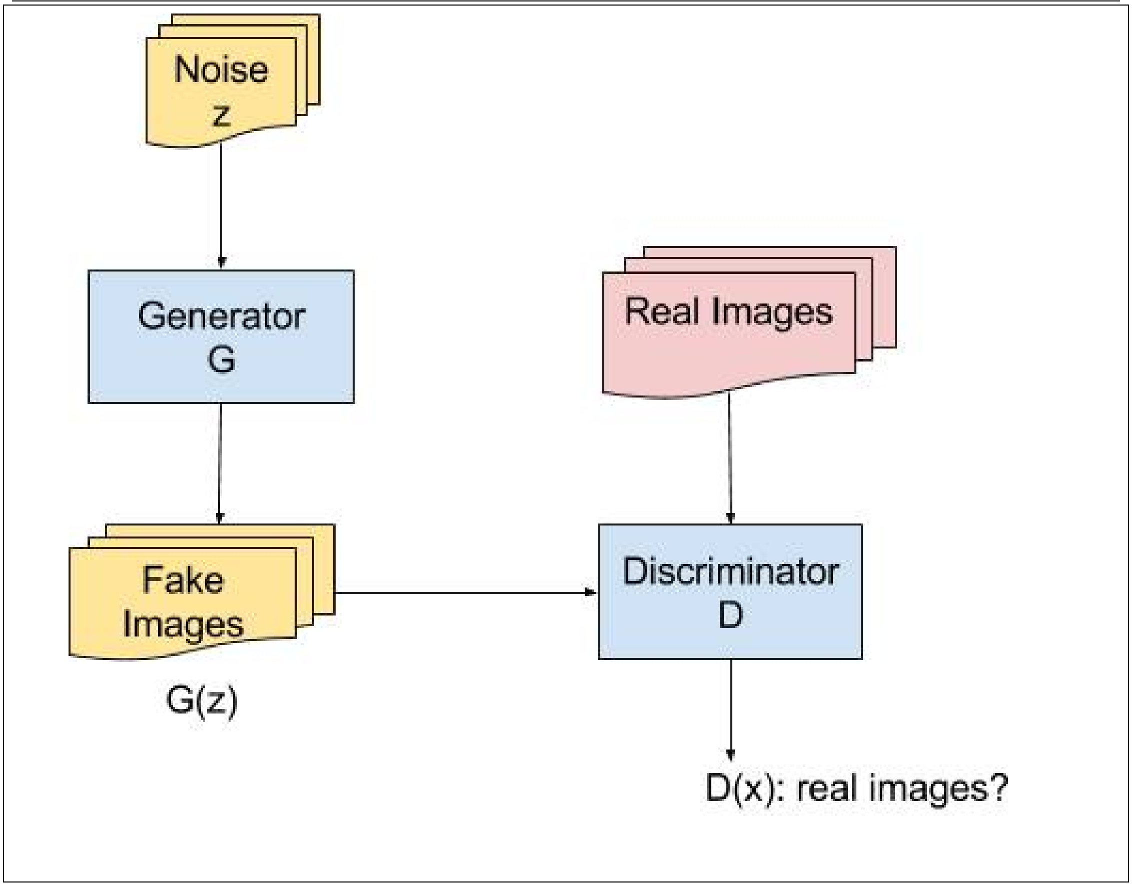


Figure 1: General Adversarial Networks Architecture

4 DESIGN AND ANALYSIS OF SYSTEM

4.1 Architecture

The architecture of Artistic GAN consist of two networks one is discriminator and other is the generator. The discriminator network is a CNN which has fully connected layers at its end.The input to the discriminator is an image, while the output is the probability that the image is real or fake. Where 0 represents the probability that the input image is fake and the image is from the generator. While 1 represents the probability that the input image is real and from the data distribution. In case of generator networks, the network consists of a fully connected layer and after transpose of CNN. An input to this network is the random noise vector this network try to generate the image base on the generator train from discriminator network. The output of the generator network is the generated image which is similar to the images on which it is trained.

4.2 Components Of Architecture

4.2.1 Convolutional Layer

Main function of it is to remember the input. Filter (Kernel) shape represent the array of numbers or weights which used to remember the features. Filter size represent the spatial dimensions of filters and its depth is same as that of depth of input. At higher level the filter are used to identify things such as curve, color, line etc. and output of filter with input is higher it represent the thing is present else not.Strides represent how much window is shifted in each of the dimensions. Output of Convolutional layer is feature map representing the feature contain in the input.

4.2.2 Batch Normalization

It stabilizes the learning by normalizing the input to the each unit to have zero mean and unit variance. This helps deals with training problems that arise due to poor initialization and flow gradient in deeper models. This proves that it is critical to get deep generators to begin learning, prevent the generator from collapsing all the samples to the same point which is common failure mode observed in GANs. Directly applying batch normalization to all layers however, result in sample oscillation and model instability. This was avoided by not applying batch normalization to the generator output layer and discriminator input layer.

4.2.3 ReLU Activation

It is used in generator with the expansion of the output layer which uses the Tanh function. We observed that using a bounded activation allowed the model to learn more quickly to saturate and cover the color space of training distribution. Within the discriminator previous work of DCGAN found that leaky rectifier activation to work well, especially the higher resolution modeling.

4.2.4 Discriminator Weight Clipping

Discriminator weight clipping is most important while training because if we does not clip the weights then it may cause the discriminator learn faster than that of generator and faster learning of discriminator cause the discriminator always reject the sample generated by generator and result in the GAN does not learn anything.

4.3 Factors Must Be Consider While Design

1. Generator and Discriminator network should be deep enough or have multiple layers so that it can learn the maximum number of feature from the datase.

2. Replace pooling layer with strided convolution.

3. Use batch normalization in the both Generator and Discriminator network except at input of Disriminator and output of Generator.

4. Use ReLU activation for all layers of generator except last layer which uses the tanh.

5. Use Leaky ReLU for all layers of discriminator.

4.4 Algorithm

Algorithm 1 Pseudocode for training of Artistic GAN, where k is number of step to apply to discriminator and also a hyperparameter.

for *num of training iteration* **do**

for *k steps* **do**

- Sample minibatch of m noise samples $(z^1, ..., z^m)$ from noise prior $p_g(z)$
- Sample minibatch of m examples $(x^1, ..., x^m)$from data generating distribution $p_{data}(x)$
- Add noise to the input of discriminator so that the train model should be robust
- Clip weight to be update of discriminator between -0.01 to 0.01
- The discriminator updated by ascending its stochastic gradient:

$$\nabla_\theta \frac{1}{m} \sum_{i=1}^{m} [\log D(x^i) + log(1 - D(G(z^i)))]$$

end

- Sample minibatch of m noise samples $(z^1, ..., z^m)$ from noise prior $p_g(z)$
- The generator updated by descending its stochastic gradient:

$$\nabla_\theta \frac{1}{m} \sum_{i=1}^{m} log(1 - D(G(z^i)))$$

end

The gradient-based updates can use any available and suitable standard gradient-based learning rule.

5 DISCUSSION ON IMPLEMENTATION RESULTS

5.1 Dataset

In this work we used the publicly available WikiArt dataset for our experiment. WikiArt is the largest publicly available dataset that contains around 80,000 artwork in terms of style class, artist, genres, media. WikiArt system is based on the principle same as that of wikipedia such that any can freely add, remove and edit the contents by participating. However for this experiment we take only one class of images that is Landscape.

5.2 Experimental Setting

Implementation of Artistic GAN was done on the WikiArt dataset here we target our scope of image generation specific to the landscape images generation. In this experiment we have 14978 images we clean all images during preprocessing to the jpeg format and the channels are restricted to 3 which is RGB. We also convert the RGBA images to RGB and all images are resize to 128×128pixel. Also from the many standard experiment conducted on the DCGAN which is related to our experiment it found that learning rate 2×10^{-4} is best suitable. We also use alpha=0.2 for our Leaky ReLU, Minibatch of 64 images, Discriminator wright are clipped between -0.01 to 0.01. Also convoluitonal layer with kernel size means the filter size is 5 and size of strides is 2. All layer has batch normalization except the input of discriminator and output of generator.

5.3 Result

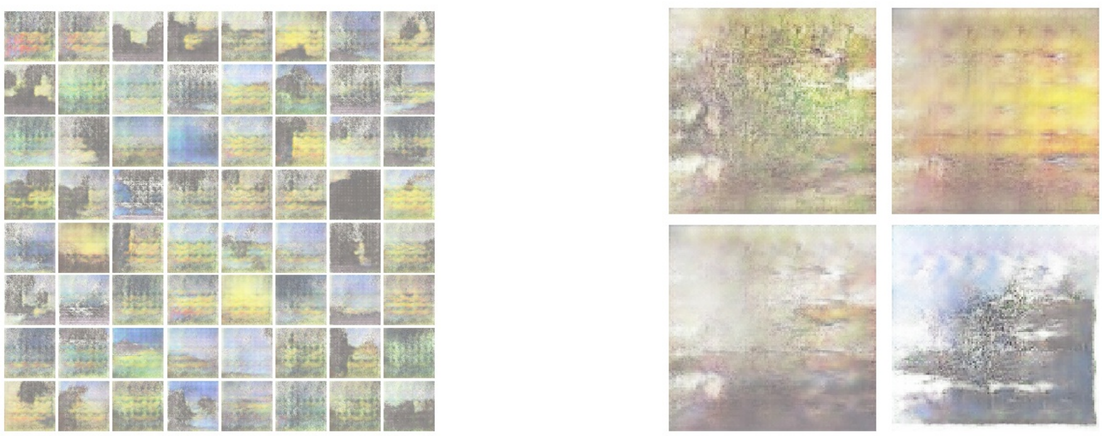
We conducted the experiment on the Artistic GAN architecture which we discuss above we found that the images generated by the this are good and due to clipping of discriminator weights it reduces the fast learning of discriminator than that of the generator which protect the generator from mapping multiple input to single output. The probability of mode collapse problem is drastically reduce. The Quality of images generated by and the mapping of distribution from random unifrom vector to the output image is learn very good. The feature learn discriminator and from it by generator is high and helps to generate the high quality image as the output of generator.

Same dataset and the experimental setup is used for the DCGAN architecture and after some time it try produce the same type of images with less quality as output of generator. In this case of DCGAN mode collapse occurred and maps the multiple inputs to the same output.The mapping between input random vector to the output generated images is weaker. The feature learn by the generator from the discriminator is less and unsatisfied. The result produced by the generator is poor and high probability that is cause mode collapse.

(a) from generated images, it represents that the generator learns the good and avoid the repetition.

(b) In this figure, we can see that the ArtistGAN learn the different feature of nature. We can see the mountain, river and trees etc.

Figure 2: Images Generated by Artistic GAN on landscape Images



(a) from generated images, it represents that the generator learns not as per our requirement and the most of the generated images are same it confirms that the mode collapse occurs. And multiple inputs have the same output.

(b) In this figure, we can see that the DCGAN learn the different feature of nature but still it need to improve

Figure 3: Images Generated by DCGAN on landscape Images

6 CONCLUSION AND FUTURE ENHANCEMENT

6.1 Future Enhansment

The result of Artistic GAN is good but we found the slightly bur image which can be reduced by applying the multiple stack of GAN such as Laplacian Pyramid. Apply same concept multiple real life domain to reduce the time required to design the new designs. Furthermore, we are also interested in jointly learn these model which generate artwork in combine with other models. 3D modeling such as 3D object reconstruction from single depth view. Unsupervised anomaly detection using adversarial network for any kind anomaly detection dataset.

6.2 Conclusion

Design process in any domain is the hardest part and require time to produce the design. Artistic GAN can do this in much less time for the domain specific work. In this experiment the Artistic GAN to synthesized much challenging and complex images. A natural extension this experiment can be used deeper network so the to learn more detail features. This experiment on Landscape images generation by Artistic GAN is done successfully and generate images are high quality and similar to the train images.

References

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. WardeFarley,S. Ozair, A. Courville, and Y. Bengio, " Generative adversarial nets," In NIPS, pages 2672–2680, 2014.

[2] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," In ICLR, 2016.

[3] Martin Arjovsky, Soumith Chintala, and Léon Bottou, "Wasserstein gan," arXiv preprint arXiv:1701.07875, 2017.

[4] Sanjeev Arora, Rong Ge, Yingyu Liang, Tengyu Ma, and Yi Zhang, "Generalization and equilibrium in generative adversarial nets (gans)," arXiv preprint arXiv:1703.00573, 2017.

[5] J. T. Springenberg, "Unsupervised and semi-supervised learning with categorical generative adversarial networks," arXiv preprint arXiv:1511.06390, 2015.

[6] L. Theis, A. v. d. Oord, and M. Bethge,"A note on the evaluation of generative models,"arXiv preprint arXiv:1511.01844, 2015.

[7] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell,"Adversarial feature learning," arXiv preprint arXiv:1605.09782, 2016.

[8] Chelsea Finn, Paul Christiano, Pieter Abbeel, and Sergey Levine," A connection between generative adversarial networks, inverse reinforcement learning, and energy-based models," arXiv preprint arXiv:1611.03852, 2016.

[9] Ian Goodfellow,"Nips 2016 tutorial: Generative adversarial networks," arXiv preprint arXiv:1701.00160, 2016.

[10] Xun Huang, Yixuan Li, Omid Poursaeed, John Hopcroft, and Serge Belongie," Stacked generative adversarial networks," arXiv preprint arXiv:1612.04357, 2016.

[11] Daniel Jiwoong Im, Chris Dongjoo Kim, Hui Jiang, and Roland Memisevic," Generating images with recurrent adversarial networks," arXiv preprint arXiv:1602.05110, 2016.

[12] M. Mirza and S. Osindero," Conditional generative adversarial nets,"arXiv preprint arXiv:1411.1784, 2014.

[13] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling, "Semi-supervised learning with deep generative models," in NIPS, 2014, pp. 3581–3589.

[14] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in ICLR, 2008, pp. 1096–1103.

[15] T. D. Kulkarni, W. F. Whitney, P. Kohli, and J. Tenenbaum, "Deep convolutional inverse graphics network," in NIPS, 2015, pp. 2530–2538.