Available online freely at www.isisn.org

Bioscience Research

Print ISSN: 1811-9506 Online ISSN: 2218-3973

Journal by Innovative Scientific Information & Services Network

RESEARCH ARTICLE BIOSCIENCE RESEARCH, 2020 17(2): 1356-1363. **OPEN ACCESS**

Image generation and classification via generative adversarial networks

Shirin Mirabedini[1*], Shadi Hejareh Dastgerdi[2], Mohammadreza Kangavari[2] and Mahdi AhmadiPanah[3]

[1]Department of Computer Engineering, Payame Noor University, PO Box 19395-3697 Tehran, I.R of **Iran**
[2]Department of Computer Engineering Iran University Science and Technology (IUST) Tehran, **Iran**
[3]Department of Management, Payame Noor University, PO Box 19395-3697 Tehran, I.R of **Iran**

***Correspondence:** sh.mirabedini@pnu.ac.ir **Received 06-05-2020, Revised: 13-06-2020, Accepted: 15-06-2020 e-Published: 17-06-2020**

In recent years, bio image processing via supervised learning with convolutional neural networks (CNNs) has seen huge adoption in computer vision applications. Comparatively, unsupervised learning with CNNs has received less attention. Generative adversarial networks (GANs) are an emerging technique for both semi-supervised and unsupervised learning that has made a dramatic change in the computer vision field. GANs provide a way to learn deep representations without extensively annotated training data. They achieve this through deriving back propagation signals through a competitive process involving a pair of networks designed by game theory. GANs may be used in a variety of applications, including bio image generation, semantic image editing, style transfer, bio image super-resolution and also classification. Image classification is an issue that utilizes image processing, pattern recognition, and classification methods. We present two applications of GANs: classification, and the generation of images that humans find visually realistic. We focus on MGAN on the MNIST dataset and we apply this architecture in the form of distributed as our novelty. The proposed method experiment on the cluster with multiple machines to improve time-consuming with data distribution and removing the interaction between nodes during the training. Using the voting algorithm to increase the accuracy of classification. We prove the improvement by using T-Test. We conclude the proposed method decreases the training time and improves the accuracy by voting.

Keywords: Game theory, Generative adversarial networks, Bio image Generation, Bio Image Classification, Deep neural networks, Deep learning.

INTRODUCTION

A field of active research has been learning reusable feature representations from large unlabeled datasets. In the sense of computer vision, the virtually unlimited amount of unlabeled images and videos can be leveraged to learn good intermediate representations, which can then be used on a number of supervised learning tasks such as image classification. Supervised learning in recent years supervised learning with convolutional neural networks (CNNs) has seen massive adoption in applications for computer vision in recent years. By contrast, less attention has been paid to unsupervised learning with CNNs. GANs help as a bridge in the gap between CNN's supervised learning and unsupervised learning. There is also a wide range of research on image generation models based on neural networks. Earlier groups of probabilistic image models include restricted Boltzmann machines (e.g., (Hinton et al., 2006)) and their deep variants (Salakhutdinov et al., 2009), auto-encoders (Hinton et al., 2006, Salakhutdinov et al., 2009, Vincent et al., 2008) and more recently stochastic neural networks (Bengio et al., 2014, Gregor et al., 2015, Kingma et al 2104) and deterministic networks (Dosovitskiy et al., 2015), respectively. Goodfellow et al., suggested the Generative Adversarial Network (GAN) (Goodfellow et al., 2014), train a generative network in a mini-max objective together with a second biased adversarial network.

There are two subnetworks in the GAN system, a generator and a discriminator. A set of images (training images) is accessible to the discriminator. The discriminator attempts to differentiate between the generator-generated "real" images (from the training set) and "fake" images.

Without seeing these images, the generator attempts to produce images identical to the training set. The generator begins by generating random images and the discriminator receives a signal as to whether the discriminator considers them true or fake. The discriminator should not be able to tell the difference between the images produced by the generator and the actual images in the training set at equilibrium, therefore the generator is able to generate images that come from the same distribution as the training set.The foci of this paper are on providing an application of major advanced GAN architectures used for improving image generation and classification as a form of distributed.

Deep learning (LeCun et al., 2015) allows multi-layered computational models to learn data representations with multiple abstraction levels. These methods have improved dramatically the state-of - the-art computer vision, speech recognition, visual object recognition, object detection, and many other domains such as drug discovery and genomics.

Deep learning explores intricate structure in large data sets by using the backpropagation algorithm to show how a computer can change its internal parameters using the representation in the previous layer to determine the representation in each layer.

Deep convolutional networks have produced breakthroughs in image, video, voice, and audio processing, while recurrent networks have shed light on sequential data such as text and speech.

Approach to generative adversarial networks (Goodfellow et al., 2014) proposed by Goodfellow et al., is a paradigm for generative model training. These consist of two' adversarial' models: a generative model G measuring the distribution of data, and a discriminative model D estimating the likelihood of a sample coming from the training data rather than from G. Both G and D constructed by the theory of games might be a nonlinear mapping function, such as a multi-layer perceptron.

To learn a generator distribution $p_g$over data x, the generator builds a mapping function from a prior noise distribution p_z (z) to data space as G (z;θ_g). And the discriminator, D (x;θ_d), outputs a single scalar representing the probability that x came from training data rather than pg.

G and D are both trained simultaneously: It adjusts parameters for G to minimize log(1 - D(G(z)) and adjust parameters for D to minimize log D(X), as if they are following the two-player min-max game with value function V(D, G): (Goodfellow et al., 2014, Kjeldsen 2001, Willem 1996)

$$\min_G \max_D V(D,G) = E_{x\sim p_{data(x)}}[\log D(x)] + E_{z\sim p_{z(z)}}[\log(1 - D(G(z)))] \qquad (1)$$

This encourages G to fit P_{data} (x) so as to fool D with its generated samples p_z (z). Both G and D are trained by back propagating the loss in Eqn. 1 through their respective models to update the parameters.

p_{data} (x) is representing the probability density function over a random vector x which lies in R|x|. It will be used p_g (x) to denote the distribution of the vectors produced by the generator network of the GAN. It used symbols G and D to denote the generator and discriminator networks, respectively. Both networks have sets of parameters (weights), θ_d and θ_g, that are learned through optimization, during training.

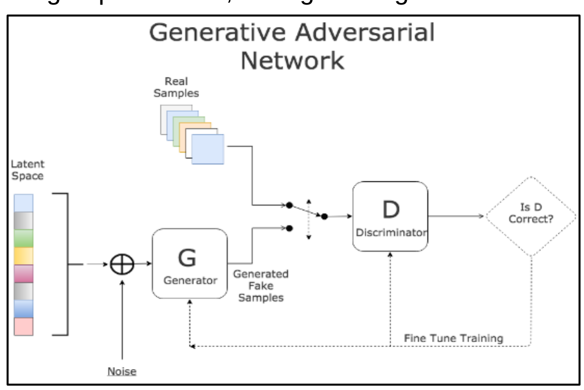

Figure 1: The Framework of GAN

On top of synthesizing novel data samples via GANs, which may be used for downstream tasks such as semantic image editing (Zhu et al., 2016), data augmentation (Bousmalis et al., 2016) and style transfer (Zhu et al., 2017), It's specially interested in using the representations that such models learn for tasks such as classification (Radford et al., 2016) and image retrieval (Creswell et al., 2016). In the next section, it's explained two major types of GAN architectures.

MATERIALS AND METHODS

We're concentrating on MGAN (Hoang et al., 2018) for image generation in this article and the using dataset is MNIST. The MNIST database is a large database that includes 60000 images of handwritten digits that are commonly used for training various image processing systems. GANs ' teaching is time-consuming. The MGAN is more complex than the traditional GANs. Therefore, the training time of MGAN is longer than GANs. Using more data to train neural networks is a better model, but it takes a long time to train. Data parallelization is the way to solve the lengthy training time. So it's used multiple MGAN in our experiments as our novelty that describe in the next section.

Parallelization Stage

Software Parallelism is a way of distributing computation among different machines in such a way that software is distributed across different machines and some processing is done locally in each machine using the entire model, but only for part of the complete data collection.

According to MGAN architecture, we could not use such data parallelism approaches. That generator trains to create specific data groups. During preparation, we don't know about generators (e.g. first generator trains producing the x class in the first machine, first generator trains producing the y class in the second machine) because data parallelism methods can't properly aggregate and update parameters. We proposed the method in this paper that machines should be independent from each other. It ensures that a portion of data is learned by everyone. Through eliminating the communication between computers, network contact has thus been removed, training has been completed more easily and the issue of updating parameters has been resolved. On each unit, Algorithm 1 presents the training pseudo-code.

Yet, for two reasons, this method reduces the accuracy: Reduction of training data and lack of parameter update interaction between machines. So we have to solve this issue.

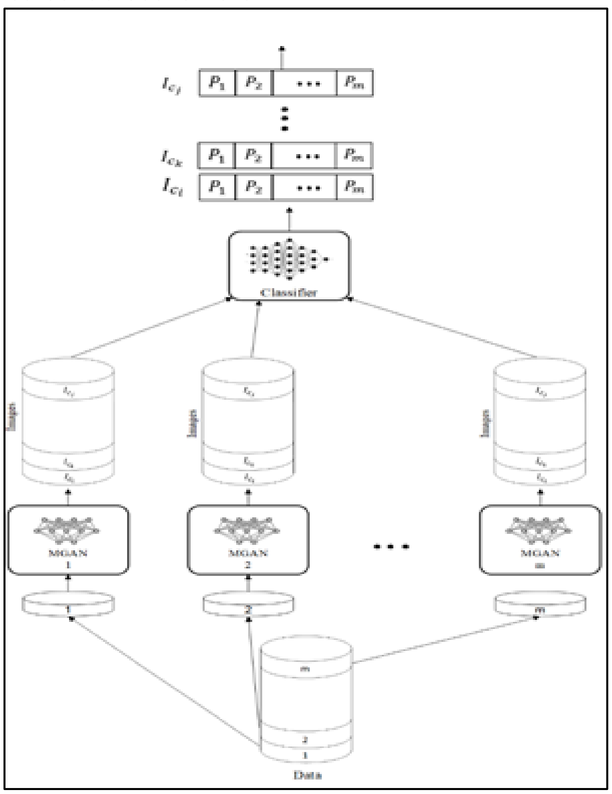

Figure 2: The framework of distributed MGAN

In the next stage, it should be classified MGAN synthetic images that describe in the next section.

Classification Stage

After the training of the parallel networks, in the classification stage, a classifier will classify the synthetic images produced for the evaluation of the experiments. After each machine's training, all the generated images are sent to one classifier in our proposed method. We are graded by the classifier. Figure 2 shows how our solution is being proposed. After the classification, we use a voting algorithm to find the majority of a sequence of labeled. This approach helps to classify more accurate and resolves the mentioned problem.

As shown in Alogorithm1, there are V machines in the cluster. Each MGAN runs in one node in the cluster. Generators G_1, G_2,..., G_K in our proposed method is $\theta_{G_{i,}}$parameterized deep convolutional neural networks. Except for the input layers, these networks share parameters in all layers. The mapping $f_{\theta_{G_{i,}}}(z_i)$ parameterizes the input layer for generator G_k, which maps the sampled noise z to the first hidden layer activation h. The $g_{\theta_{G_i}}(h_i)$ mapping parameterizes the shared layers, which maps the first hidden layer to the generated data. The mixture sampling pseudo-code is defined in Alogorithm1. It's shared parameters in all layers.

RESULTS AND DISCUSSION

In this section, we first introduce the experimental environment, and then we conduct experiments on MNIST dataset (LeCun et al., 2011) that contains handwritten images of digits 0-9 and is used extensively as a benchmark in testing the performance of large scale neural networks and other algorithms, for both speed of computation and the quality of results. It contains 60,000 training and 10,000 testing samples of 28×28 grayscale pixel images of the digits. The experimental hardware setting is shown in Table 1.

We use TensorFlow to implement our model and Distributed-Tensorflow for parallelization. For all experiments, we use: (i) Adam optimizer (Kingma et al., 2014) with learning rate of 0.0002 and the first-order momentum of 0.5; (ii) minibatch size of 64 samples for training discriminators; (iii) ReLU(Nair et al., 2010) activations for generators; (iv) Leaky ReLU (Maas et al., 2013) with slope of 0.2 for discriminator and classifier; and (v) weights randomly initialized from Gaussian distribution N(0; 0:02I) and zero biases.

Table 1: The experimental hardware setting

API	Python
OS	Ubuntu
CPU	Intel Core i7
GPU	NVidia GTX 1050 Ti
	NVidia GTX 1080
	NVidia GTX 1080 Ti
RAM	16 GB
	20 GB
	20 GB
Disk	2 TB
	250 GB
	250 GB

We first evaluate the model by different epoch to find the epoch that makes a model with high accuracy and optimal training time on different nodes in a cluster.

The experiments show that when epoch is equal to 100 make the best model according to time-consuming and the accuracy. After the find the best number of the epoch, we evaluate the model with different batch sizes.

The experiments on the different batch sizes show that the best batch size for the proposed model is 32. By increasing the number of nodes in the cluster, we improve the time consuming of MGAN but there is a problem.

Table 2: shows the result of T-Test

Number of nodes	P-Value
3	3.391e-15
4	2.2e-16
5	2.2e-16
6	8.922e-16

The result of T-Test proves the improvement of the voting algorithm.Because of the reduction of data on each machine, the accuracy is disimproved. So, the voting algorithm helps to increase accuracy. Figure 3 shows the influence of the voting algorithm. The results show that when there are less than 3 nodes, the voting algorithm can't improve because this algorithm needs more than 2 elements to vote. We can see that this algorithm with more elements in voting does better. We do T-Test to prove the influence of the voting algorithm on improving accuracy. A t-test is a type of inferential statistics used to determine if there is a significant difference between the means of two groups, which may be related to certain features. In this test, if p-value be achieved lower than 0.05, conclude that Improvements have been made. Table 2 shows the result of T-Test.

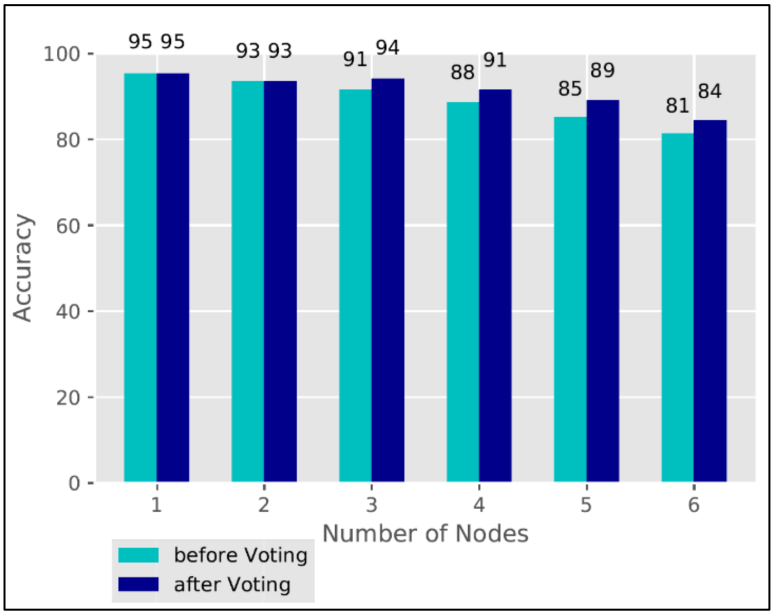

Figure 3: accuracy before and after voting

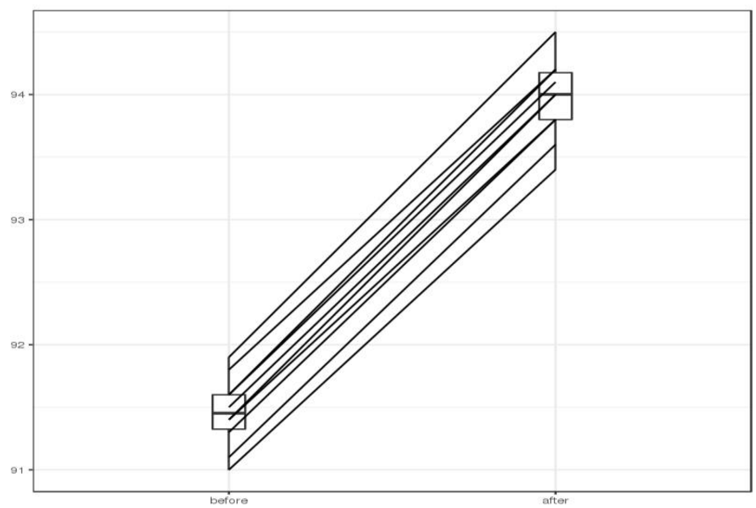

Figure 4: Degree of improvement of the classification accuracy in the 3-classifier by 10 times

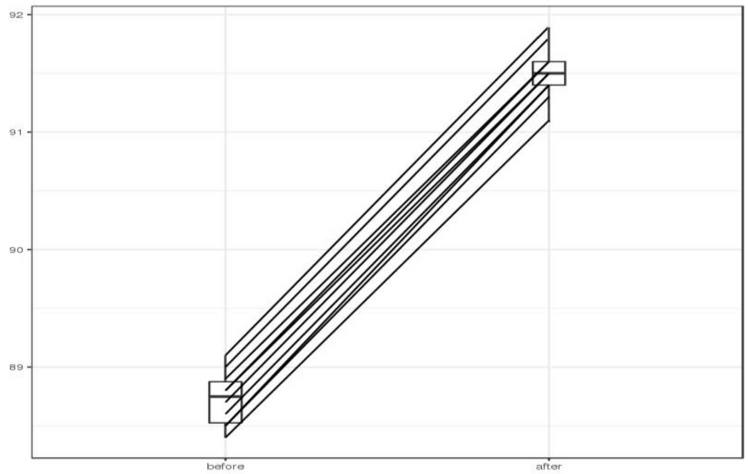

Figure 5: Degree of improvement of the classification accuracy in the 4-classifier by 10 times

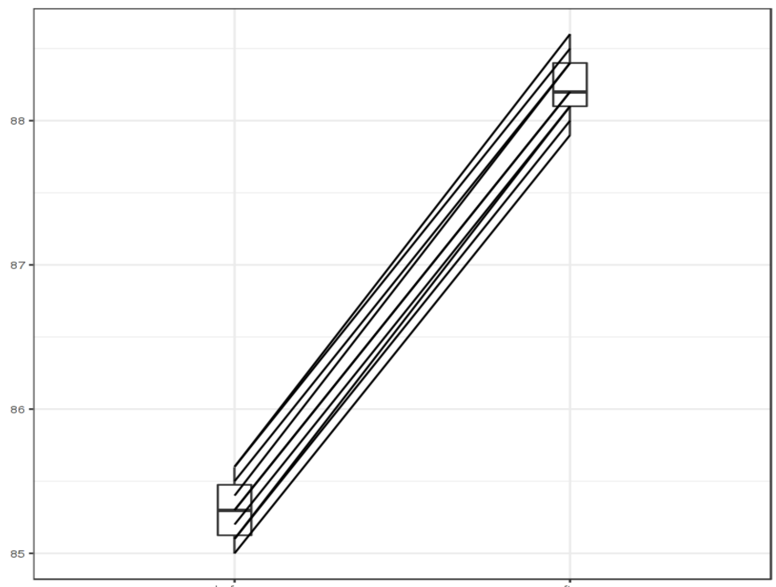

Figure 6: Degree of improvement of the classification accuracy in the 5-classifier by 10 times

Table 2 shows the accuracy of the classification in the case of using a single-classifier and multi- classifier mode. Since at least three data sets are required to vote, at least three classifier and a machine inter clustering are required to implement this algorithm. In order to evaluate, the paired sample T-test is evaluated separately on 3 to 5 classifier networks. The results of this assessment are shown in Figures 4-5-6.

CONCLUSION

We investigated a new technique in bio image generation called Generative Adversarial Networks. GANs designed by game theory is one of the hottest subjects in machine learning right now. GAN's Framework includes pair sub-networks (generator and discriminator) that compare based on game theory. They are one of the few successful techniques in unsupervised machine learning and are quickly revolutionizing our ability to perform bio generative tasks. We focus on MGAN for image generation and we apply this architecture in the form of distributed as our novelty. In this method, we used V machines in the cluster and it's run one MGAN in per machine. The machines trained on the portion of data independently. Then the synthetic images were classified in the classification stage. We experiment on the cluster with a various number of machines. We conclude the proposed method decreases the training time and improves the accuracy by voting algorithm.

CONFLICT OF INTEREST

The authors declared that present study was performed in absence of any conflict of interest.

ACKNOWLEGEMENT

The correspondence author would like to acknowledge the Faculty of Computer engineering, Iran University of Science and Technology, and the Professor Kangavari laboratory, for their support and contribution to this study.

AUTHOR CONTRIBUTIONS

MK advised the road of research and experiments. SM and SD participated in data collection and analysis of the data and designed model architecture and experiment and research. All authors read and approved the final version.

REFERENCES

A. Creswell and A. A. Bharath (2016), "Adversarial training for sketch retrieval," in Computer Vision – ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part I. Springer International Publishing.

A. Radford, L. Metz, and S. Chintala (2016), "Unsupervised representation learning with deep convolutional generative adversarial networks," in Proceedings of the 5th International Conference on Learning Representations (ICLR) - workshop track.

A. Radford, L. Metz, and S. Chintala (2016), "Unsupervised representation learning with deep convolutional generative adversarial networks," in Proceedings of the 5th International Conference on Learning Representations (ICLR) - workshop track.

A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb (2016), "Learning from simulated and unsupervised images through adversarial training," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. (2013) Rectifier nonlinearities improve neural network acoustic models. In Proc. ICML, volume 30. 5

Bengio, Y., Laufer, E., Alain, G., Yosinski, J. (2014): Deep generative stochastic networks trainable by backprop. In: ICML. 226–234.

C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Aitken, A. Tejani,J. Totz, Z. Wang, and W. Shi (2017), "Photo-realistic single image super-resolution using a generative adversarial network," in IEEE Conference on Computer Vision and Pattern Recognition.

Diederik Kingma and Jimmy Ba. Adam (2014): A method for stochastic optimization. arXiv preprint arXiv:1412.6980. 5

Dosovitskiy, A., Tobias Springenberg, J., Brox, T. (2015): Learning to generate chairs with convolutional neural networks. In: CVPR. 1538–1546.

E. L. Denton, S. Chintala, R. Fergus et al.,(2015), "Deep generative image models using a laplacian pyramid of adversarial networks," in Advances in Neural Information Processing Systems, pp. 1486–1494.

E. L. Denton, S. Chintala, R. Fergus et al.,(2015), "Deep generative image models using a laplacian pyramid of adversarial networks," in Advances in Neural Information Processing Systems, pp. 1486–1494.

Elgammal, Liu, Elhoseiny, Mazzone(2017), CAN: Creative Adversarial Networks, Generating "Art" by Learning About Styles and Deviating from Style Norms, International Conference on Computational Creativity (ICCC), held in Atlanta, GA, June 20th-June 22nd.

Gregor, K., Danihelka, I., Graves, A., Wierstra, D. (2015): Draw: A recurrent neural network for image generation. ICML

Hinton, G.E., Salakhutdinov, (2006) R.R.: Reducing the dimensionality of data with neural networks. Science 313(5786) 504–

507
Hoang, Quan, Tu Dinh Nguyen, Trung Le, and Dinh Phung.(2018) "MGAN: Training generative adversarial nets with multiple generators.".
I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio.(2014) Generative adversarial nets. In NIPS, pages 2672–2680.
I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley,S. Ozair, A. Courville, and Y. Bengio (2014) , “Generative adversarial nets,” in Advances in Neural Information Processing Systems, pp. 2672–2680.
J. Wu, C. Zhang, T. Xue, B. Freeman, and J. Tenenbaum (2016), “Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling,” in Advances in Neural Information Processing Systems, pp. 82–90.
J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros,(2016) “Generative visual manipulation on the natural image manifold,” in European Conference on Computer Vision. Springer, pp. 597–613.
J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros,(2017) “Unpaired image-toimage translation using cycle-consistent adversarial networks,” in Proceedings of the International Conference on Computer Vision. [Online]. Available: https://arxiv.org/abs/1703.10593
K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan (2016), “Unsupervised pixel-level domain adaptation with generative adversarial networks,” in IEEE Conference on Computer Vision and Pattern Recognition.
Kingma, D.P., Welling, M. (2014): Auto-encoding variational bayes. ICLR.
M. Mirza and S. Osindero (2014), “Conditional generative adversarial nets,” arXiv preprint arXiv:1411.1784.
M. Zhang, K. T. Ma, J. H. Lim, Q. Zhao, and J. Feng (2017), “Deep future gaze: Gaze anticipation on egocentric videos using adversarial networks,” in IEEE Conference on Computer Vision and Pattern Recognition, pp. 4372–4381.
M.-Y. Liu and O. Tuzel (2016), “Coupled generative adversarial networks,” in Advances in neural information processing systems, pp. 469–477.
Michel Willem(1996),"Minimax Theorems (Progress in Nonlinear Differential equations and their applications)”, woodbine, NJ.
Salakhutdinov, R., Hinton, G.E. (2009): Deep boltzmann machines. In: AISTATS.
T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen(2016), “Improved techniques for training gans,” in Advances in Neural Information Processing Systems, pp. 2226–2234.
Tinne Hoff Kjeldsen (2001).”John von Neumann’s Conception of the Minimax Theorem: A Journey Through Different Mathematical Contexts”, Arch. Hist. Exact Sci. 56 39–68_c Springer-Verlag 2001.
V. Dumoulin, I. Belghazi, B. Poole, O. Mastropietro, A. Lamb,M. Arjovsky, and A. Courville(2017), “Adversarially learned inference,” in (accepted, to appear) Proceedings of the International Conference on Learning Representations.
Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.A. (2008): Extracting and composing robust features with denoising autoencoders. In: ICML.
Vinod Nair and Geoffrey E Hinton (2010). Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th international conference on machine learning (ICML-10), pp. 807–814. 5
X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel(2016), “Infogan: Interpretable representation learning by information maximizing generative adversarial nets,” in Advances in Neural Information Processing Systems.
X. Huang, Y. Li, O. Poursaeed, J. Hopcroft, and S. Belongie (2016), “Stacked generative adversarial networks,” in IEEE Conference on Computer Vision and Pattern Recognition.
X. Yu and F. Porikli (2016), “Ultra-resolving face images by discriminative generative networks,” in European Conference on Computer Vision. Springer, pp. 318–333.
Y. LeCun and C. Cortes(2011), “The MNIST database of handwritten digits,”. [Online]. Available: yann.lecun.com/exdb/mnist/
Y. LeCun, Y. Bengio, and G. Hinton (2015), “Deep learning ,” Nature, vol. 521, no. 7553, pp. 436–444.