Pune Institute of Computer Technology
Dhankawadi, Pune

A SEMINAR REPORT
ON

DEEP CONVOLUTIONAL GENERATIVE ADVERSARIAL NETWORKS

SUBMITTED BY

Nachiket Nandkishor Erlekar
Roll No. 31436
Class TE 4

Under the guidance of
Prof. Ranjeet Bidwe

DEPARTMENT OF COMPUTER ENGINEERING

Academic Year 2019-20

DEPARTMENT OF COMPUTER ENGINEERING

Pune Institute of Computer Technology

Dhankawadi, Pune-43

CERTIFICATE

This is to certify that the Seminar report entitled

"DEEP CONVOLUTIONAL GENERATIVE ADVERSARIAL NETWORKS"

Submitted by

Nachiket Nandkishor Erlekar Roll No. 31436

has satisfactorily completed a seminar report under the guidance of Prof. R Bidwe towards the partial fulfillment of third year Computer Engineering Semester II, Academic Year 2019-20 of Savitribai Phule Pune University.

Prof. R Bidwe
Internal Guide

Prof. M.S.Takalikar
Head
Department of Computer Engineering

Place:PICT (Pune)
Date:

ACKNOWLEDGEMENT

I sincerely thank our Seminar Coordinator Prof.B.D. Zope and Head of Department Prof. M.S.Takalikar for their support.

I also sincerely convey my gratitude to my guide Prof. R. Bidwe, Department of Computer Engineering, for their constant support, providing all the help, motivation and encouragement from beginning till end to make this seminar a spectacular success.

I have enjoyed working on this Seminar and I am glad that I have learned a lot of new things in the field of Deep Learning.

Contents

Abstract

Generative adversarial networks (GANs) provide a way to learn deep representations without extensively annotated training data. A GAN is a generative model that is trained using two neural network models. One model is called the "generator" or "generative network" model that learns to generate new plausible samples. The other model is called the "discriminator" or "discriminative network" and learns to differentiate generated examples from real examples. The two models are set up in a contest or a game (in a game theory sense) where the generator model seeks to fool the discriminator model, and the discriminator is provided with both examples of real and generated samples. In this work, the combination of CNN and GAN for unsupervised learning is done, it bridges the gap between the success of CNNs for supervised learning and unsupervised learning. A new class of CNNs called deep convolutional generative adversarial networks (DCGANs) is introduced, that have certain architectural constraints, and demonstrate that they are a strong candidate for unsupervised learning.

Keywords

Generative network, adversarial network, fractional strided-convolutions, discriminator, unsupervised DCGANs

1 INTRODUCTION

Artificial intelligence (AI) is a thriving field with vast practical applications and active research fields. It strives to simulate human intelligence to think and react like humans. We use AI to automate routine labor, understand speech or images, make diagnostic analysis in medicine and make advances in scientific research of various fields. A true conundrum for AI is to solve the tasks that are easyily performed by people but are hard for people to describe problems that we solve intuitively, like recognizing spoken words or faces in images.

Machine Learning is an approach that allows a machine to solve a task by learning from experience. A machine learning algorithm is an algorithm that is able to learn from a dataset, be it labelled or unlabelled data. One of the most famous definitions of Machine Learning is given by Mitchell : "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P , if its performance at tasks in T, as measured by P , improves with experience E".

A major source of difficulty in many real-world AI applications is that many of the factors of variation influence every single piece of data we are able to observe and quality and quantity of data equally matters for AI-based problems. The individual pixels in an image of a red car might be very close to black at night. The shape of the car's silhouette depends on the viewing angle. Most applications require us to extricate the elements of variation and discard the ones that are least cared about.

Extracting such high level abstract features from data might seems a very difficult task. This is where Deep Learning comes into picture and saves the day. Deep learning allows representations to be constructed from other simpler representations. For example, for face recognition in an image, a DL model first learns the various edge formations in the image, it then tries to understand how various edges converge to form simple shapes and finally starts to learn more abstract facial features like the shape of a nose, eyes, ears, chin, etc.

The most commonly used deep learning model is a Multi-layer Perceptron, more popularly known as the deep Neural Network. Neural Network is inspired by the biological structure of neurons in a brain. While we don't have a very strong understanding of how exactly a neuron functions, however, an ANN does a good job of at least imitating the abstract behavior of neuron. Many deep learning engineers have been thriving to emulate the working of an actual neuron of the human brain.

Deep learning has been largely successful in various classification and Detection tasks. Classification involves classifying input dataset into various target classes based on a labelled dataset. Detection is the next step where a deep learning model tries to understand the various entities in a given input. It has also been successful in many fields such as computer vision, natural language processing, image processing, etc.

Ian Goodfellow and other researchers made a breakthrough in Deep Learning through the introduction of Generative Adversarial Networks(GAN) in 2014. GANs can be imagined as being a game between two adversaries - the generator tries to generate fake samples and trick the discriminator network, whereas the discriminator network tries to detect the generated fake samples with more accuracy. Competition in this game drives both teams to improve their methods until the counterfeits are indistinguishable from the genuine dataset.

Adversarial networks have the advantage that backpropagation is used to obtain gradients, no inference is required during representational learning, and a wide variety of factors and interactions can easily be incorporated into the model. In a basic GAN, there is no control over the type of data that will be generated and the model is unstable. This is also called as the mode control problem of a GAN. However, additional information provided along with some noise, it is possible to direct the data generation process. Such conditioning could be based on class labels, on some part of data for inpainting or even on data from different modality. Conditional GANs, a new type of GAN, uses this method to generate specific data samples depending on the input information provided.

To counter the shortcomings of the research paper published by Ian Goodfellow and his team, a new model of GANs known as Deep Convolutional Generative Adversarial Network(DCGAN). A more stable version and a stable architecture was proposed by the authors of ”Unsupervised Representational Learning with Deep Convolutional Generative Adversarial Networks”.

2 MOTIVATION

Deep learning is a kind of representation Learning where there are multiple levels of features. These features are learned progressively in a deep learning algorithm. Deep learning strategies now provided state of the art solution to many longstanding problems in computer vision and image processing. Generative Adversarial Network is one of the latest hot topic of research in deep learning. One of the most interesting parts of Generative Adversarial Networks is the design of the Generator network. The Generator network is able to take random noise and map it into images such that the discriminator cannot tell which images came from the dataset and which images came from the generator. While it is still being researched heavily, it is already proving to be very useful in real word situation. The biggest achievement is generation of new samples of any kind such as audio, video or texts. They also have been used for generating realistic images of human faces, objects and natural scenery. DCGANs shows how convolutional layers can be used with GANs and providing a series of various architectural guidelines for doing this. It also discusses topics such as Visualizing GAN features, Latent space interpolation, using discriminator features to train classifiers, and evaluating results. DCGAN architecture being one of the first GAN models to be stable providing insightful techniques and methods of how new samples can be created and acting as a base architecture upon which many other GAN architectures are built which are used for achieving super resolution, Video prediction, text-to-image translation and many timeless applications.

3 A SURVEY ON PAPERS

3.1 Generative Adversarial Networks [1]

Deep Learning attempts to learn probability distribution, features and representations over data encountered in AI. Classification Models have so far been the most successful. This success comes from backpropagation and dropout algorithms which increases accuracy of the algorithms. Deep generative models have less successful as they involve many intractable computations.

The authors propose a adversarial framework between two adversaries - the generator and the discriminator are pitted against each other. The discriminator D gives a single scalar output-the probability that the input came from data rather than from the generator output. The generator takes in some random noise and tries to generate data similar to that of sample space. D is trained to maximize the probability of correctly classifying a input. G is trained to fool the discriminator.

The objective function in this case can be given by [1],

$$\underset{G}{min}\underset{D}{max} V\,(D,G) = E_{x \sim p_{data}(x)}[\log \boldsymbol{D}(\boldsymbol{x})] + E_{z \sim p_z(z)}[\log(1 - \boldsymbol{D}(\boldsymbol{G}(\boldsymbol{z})))]. \tag{1}$$

3.2 Unsupervised representational learning using Deep Convolutional Generative Adversarial Networks [2]

For about a year after the first GAN paper, training GANs resembled art rather than science – the models were unstable and required a bunch of hacks to work. In 2016, the authors published the paper titled "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks" describing the model that subsequently became famous as DCGAN. The most noteworthy thing about DCGAN was that this architecture was stable in most settings. It was one of the first base papers to show the vector arithmetics as an inherent property of the representations learned by the Generator network: it's the same analogy as with the word vectors in Word2Vec, but with images too. For example, neutral man + smiling woman - neutral woman = smiling man. DCGANs were trained on three different datasets, i.e. the Large-scene understanding (LSUN), ImageNet-1K and a Faces dataset. The authors train the discriminator network for image classification problems, showing competitive performance with other unsupervised algorithms and visualize the filters learnt by GANs and empirically show that specific filters have learned to draw specific objects. It also conveys that the generator has interesting vector arithmetic properties that allows easy manipulation of many semantic qualities of generated samples.

3.3 Building high-level features using large-scale unsupervised learning [3]

The authors from this paper were interested in making, from a set of unlabeled data a class specific features detector. (For example, making a face detector from unlabeled face images.) To achieved this goal, the researchers trained a 9 layered sparse auto encoders on a large dataset. Contrasting to popular belief, it was possible to build a face detector without any labelled dataset, moreover surpassing the performance of the state of the art performance on ImageNet data. the architecture the authors used can be thought of deep sparse auto encoders, with some twist and they are local receptive fields, pooling and local contrast normalization. (They used L2 pooling). The style of switching between selectivity and tolerance layers and stacking a series of uniform modules, are argued to be the architecture employed by the brain. The test images consists of 37,000 images from Labeled Faces In the Wild dataset and ImageNet dataset. After training the researchers used the test dataset to measure the performance of each neuron to detect faces. In conclusion, using huge computational power as well as large amount of data, it is possible to learn high level features such as face, various body parts and human body only using unlabeled data.

4 PROBLEM DEFINITION AND SCOPE

4.1 Problem Definition

To study the behaviour and nature of deep convolutional generative adversarial networks and how new images are generated.

4.2 Scope

We will try to understand adversarial training and how two different networks compete against each other to increase their accuracy.

We first try to understand what generative adversarial networks are, allowing us to gain insight into how this particular deep learning strategy is useful for developing an efficient solution for learning about DCGANs. We then understand the nature of deep convolutional GAN which are the central model for our problem. DCGANs has wide range of applications from audio, video, texts, super resolution, generation, and many more.

5 METHODOLOGY

5.1 Basic Architecture

Following is the basic architecture of the Deep convolutional generative adversarial network.

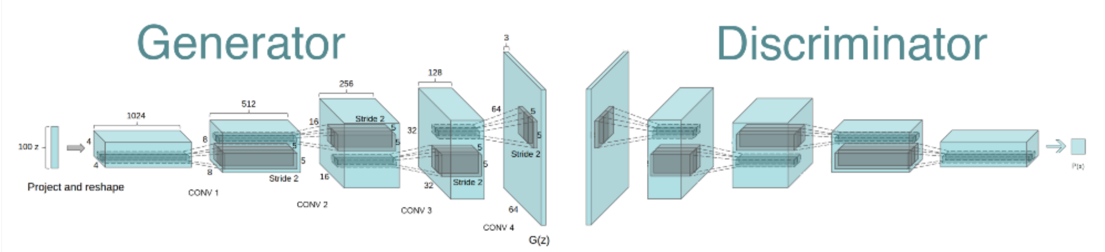


Figure 1: Overview of the proposed DCGAN [9]

5.2 Deep convolutional Generative adversarial network

A DCGAN specifically emphasizes on using Deep Convolutional networks instead of fully-connected networks. Convolutional network in general finds areas of correlation within an image, i.e. they look for spatial correlations. This means a DCGAN would likely be more fitting for image/video data since they use convolutional networks and hence demonstrate that they are a strong candidate for unsupervised learning, whereas the general idea of a GAN can be applied to wider domains, as the model specifics are left open to be addressed by individual model architectures. DCGAN consists of two networks, first one being the generator network and the second one being the discriminator network. Generator or the Generative model tries to capture the data distribution and The Discriminator or the Discriminative model estimates the probability that a sample came from the training data rather than the Generator.

5.2.1 Generator network:

A generator's job is to generate real-looking images in order to fool the discriminator. The generator model takes a fixed-length random vector as input and generates a sample in the domain. The vector is drawn randomly from a Gaussian distribution, and the vector is used to seed the generative process. This sample is feeded to the discriminator to classify as real or a fake classification. Then we calculate the loss from the discriminator classification and backpropagate through the discriminator and the generator to obtain gradients and use the gradients to change only the generator weights.

5.2.2 Discriminator network:

Discriminator's input comes from two sources, ie. from the real dataset and from the samples generated by the generator's network. Discriminator's main job is to classify the sample as real or fake. The discriminator connects to two loss functions. During discriminator training, the discriminator ignores the generator loss and just uses the discriminator loss. The discriminator loss penalizes the discriminator for misclassifying a real instance as fake or a fake instance as real and it updates its weights through backpropagation from the discriminator loss through the discriminator network.

5.2.3 Approach for a stable Deep Convolutional GAN

Replace any pooling layers with strided convolutions in the discriminator and fractional-strided convolutions in the generator. Using Batch Normalization in the generator and discriminator. Remove fully connected hidden layers for deeper architectures. Use ReLU activation in generator for all layers except for Tanh in output (These images are normalized between [-1, 1] rather than [0,1] , thus Tanh over sigmoid). Use LeakyReLU activation in the discriminator for all layers. [2]

5.2.4 Representations learnt by Generator and Discriminator

Unsupervised DCGAN when trained on a large image dataset can learn a hierarchy of features that are interesting. On a dataset of bedrooms, a discriminator activates typically on recognizing features such as beds and windows. The quality of generated data suggests that the generator learns specific object representations of major elements in a bedroom such as beds, windows, doors, and miscellaneous furniture. Previous work by researchers demonstrates that simple arithmetic operations reveal rich linear structure in representation space. One typical example demonstrated that of the vector(”King”) - vector(”Man”) + vector(”Woman”) resulted in a vector whose nearest neighbor was the vector for Queen. Same was applied for DCGANs, which gave us representations for examples like, ’Man with glasses’ - ’Man without glasses’ + ’Woman without glasses’ = ’Woman with glasses’. A ”turn” vector was created from four averaged samples of faces looking left vs looking right. By adding interpolations along this axis to random samples we were able to reliably transform their pose from left to right or vice-versa [2].

5.3 Implementation details

For each epoch, in each batch of images, generate noise vectors of dimension 100 and feed it as input to the generator network. The noise vector z is fed into a fully connected layer with 7x7x256 (12544)hidden units. 128 filters of size 5x5x256, with (1,1) strides, “same” padding* and bias=0 are used in the generator which will produce samples of size 28*28*1. Now, input the fake images and real images to the discriminator to get fake output and real output respectively. Ideally, the discriminator predicts 0 for a fake image and 1 for a real image. [12]

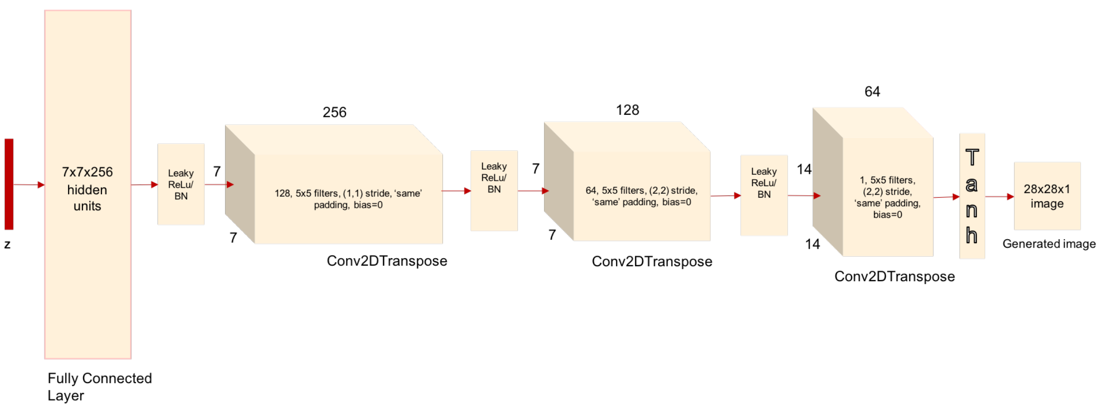


Figure 2: The Generator Network [12]

Image (generated/real) is given as input to a Conv2D in the discriminator with the following features: 64 filters of size 5x5x1, (2,2) strides, “same” padding and bias=0. The output (14x14x64) is fed into a Leaky ReLu activation function and dropout rate of 0.3. The output from the dropout is fed into another Conv2D with the following features:128 filters of size 5x5x64, stride (2,2), “same ” padding and bias=0. ”same” padding keeps the output size same as the input size when stride =(1,1). The output (7x7x128) is fed into Leaky ReLu and dropout with a rate of 0.3 and then flattened into a vector of size 6272(7x7x128=6272) which is given as input to to fully connected layer with one hidden unit. The output of the fully connected layer is the discriminator output. Adam optimizer with a learning rate 0.0001 is used for optimization. Training is done for 50 epochs with a batch size of 256 [12].

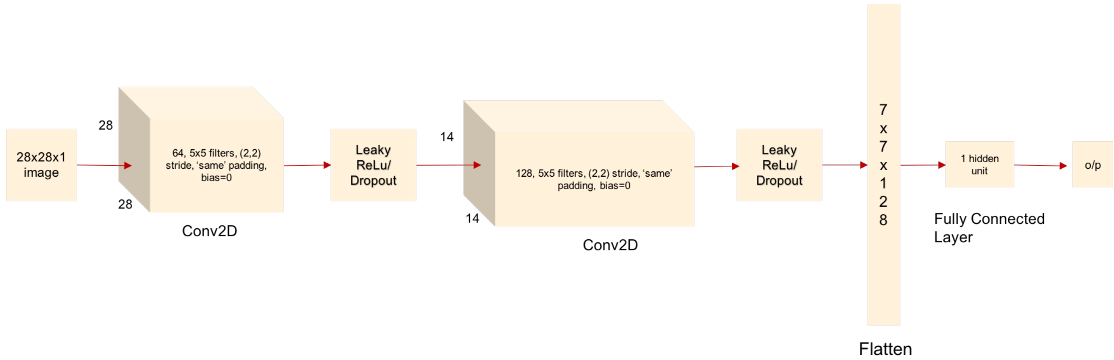


Figure 3: The Discriminator Network [12]

6 Experimental Details

6.1 Dataset

Three different datasets are used.
(1) LSUN dataset [11]: (Large Scale Scene Understanding dataset) To demonstrate how the model scales with more dataset and higher resolution generation, the model is trained on the LSUN bedrooms dataset containing over 3 million training samples. Data augmentation was not applied to the images. Deduplication technique is used to detect and remove approximately 275,000 near duplicates, suggesting a high recall.
(2)FACES dataset: Human faces images scraped from random web image queries of peoples names. The people's names were acquired from dbpedia, with a criterion that they were born in the modern era. This dataset has 3M images from 10K people. An OpenCV face detector is run on these images, keeping the detections that are sufficiently high resolution, which gives approximately 350,000 face boxes. Face boxes were used for training. No data augmentation was applied to the images.
(3)IMAGENET -1 K dataset [10]: Imagenet-1k used as a source of natural images for unsupervised training. Train on 32 32 min-resized center crops. No data augmentation was applied to the images.

6.2 Parameter Setting

No preprocessing was applied to the training dataset besides scaling to the range of the tanh activation function [-1, 1]. All models were trained with mini-batch stochastic gradient descent (SGD) with a mini-batch size of 128. All weights were initialized from a zero-centered Normal distribution with standard deviation 0.02. In the LeakyReLU, the slope of the leak was set to 0.2 in all models. The learning rate of 0.0002 is used. Additionally, leaving the momentum term 1 at the suggested value of 0.9 resulted in training oscillation and instability while reducing it to 0.5 helped stabilize training. We feed random values to our generator, which will learn to create digits out of this noise. We also take care that neither the generator nor the discriminator becomes too strong by balancing their losses. [2]

7 Results

7.1 CLASSIFYING CIFAR-10 USING GANS AS A FEATURE EXTRACTOR:

Model	Accuracy	Accuracy (400 per class)	max # of features units
1 Layer K-means	80.6%	63.7% (±0.7%)	4800
3 Layer K-means Learned RF	82.0%	70.7% (±0.7%)	3200
View Invariant K-means	81.9%	72.6% (±0.7%)	6400
Exemplar CNN	84.3%	77.4% (±0.2%)	1024
DCGAN (ours) + L2-SVM	82.8%	73.8% (±0.4%)	512

Table 1: CIFAR-10 classification results using the pre-trained model. This DCGAN is not pre-trained on CIFAR-10, but on Imagenet-1k, and the features are used to classify CIFAR-10 images [2].

7.2 CLASSIFYING SVHN DIGITS USING GANS AS A FEATURE EXTRACTOR

1000 uniformly class distributed training examples are randomly selected and used to train a regularized linear L2-SVM classifier on top of the same feature extraction pipeline used for CIFAR-10.

Model	error rate
KNN	77.93%
TSVM	66.55%
M1+KNN	65.63%
M1+TSVM	54.33%
M1+M2	36.02%
SWWAE without dropout	27.83%
SWWAE with dropout	23.56%
DCGAN (ours) + L2-SVM	22.48%
Supervised CNN with the same architecture	28.87% (validation)

Table 2: SVHN classification with 1000 labels [2].

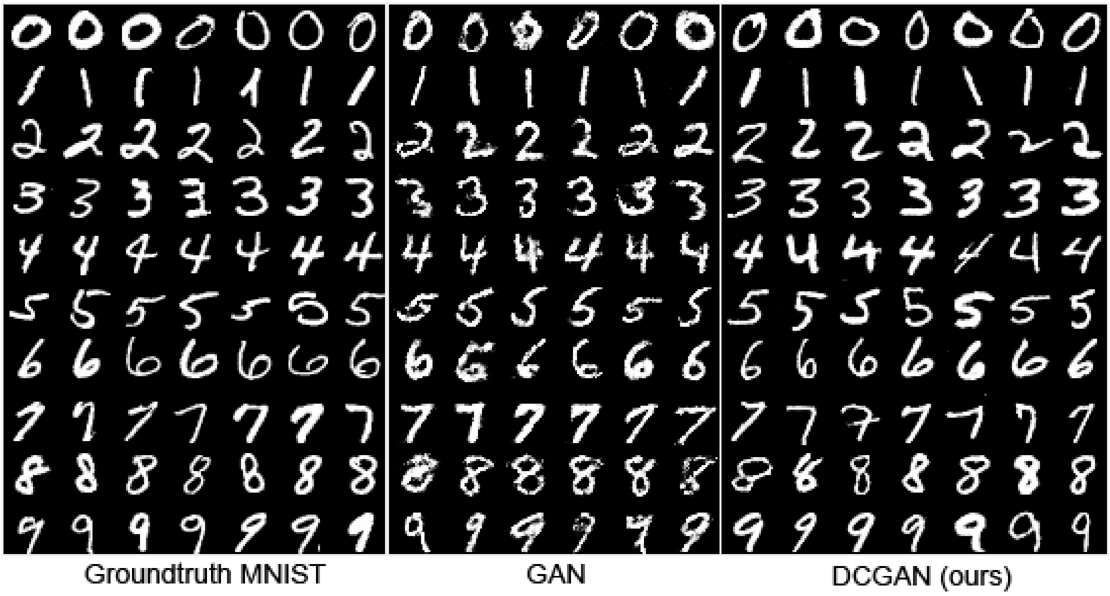
7.3 EVALUATING DCGANS CAPABILITY TO CAPTURE DATA DISTRIBUTIONS

Model	Test Error @50K samples	Test Error @10M samples
AlignMNIST	-	1.4%
InfiMNIST	-	2.6%
Real Data	3.1%	-
GAN	6.28%	5.65%
DCGAN (ours)	2.98%	1.48%

Table 3: Nearest neighbor classification results [2].

7.4 COMPARISON OF REAL DATA AND SAMPLES GENERATED BY GANS AND DCGANS

Figure below [2] shows Side-by-side illustration of (from left-to-right) the MNIST dataset, generations from a baseline GAN, and generations from DCGAN.



As we can see, the baseline GAN generates samples that are blurry and contain some noise but are however recognizable. DCGANs produces samples with a better resolution and comparatively less noise.

8 Conclusion

We have looked at the basics and the adversarial nature of GANs and DCGANs. We have learned about the roles of generator network and discriminator network in a GAN and how they work in a competition to increase their accuracies. We have also learnt how the DCGAN is made more stable than the traditional GAN via its architecture and how convolutions are used and architectural changes are made to traditional GANs. We have looked how representations are learned by the networks and what features they detect and generate. We also obtain conclusive results where less noise is generated and better resolution images are created. Experiments on popular datasets might give better results. DCGANs however require very high computation power such as GPUs and TPUs and are quite expensive in terms of time and power but have tremendous scope of improvement in its research and architecture. We also infer that DCGAN has wide range for future scope for other domains such as audio or video or texts.

References

[1] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S.,Courville, A., and Bengio, Y., ”Generative Adversarial Nets.”, Proceedings of the International Conference on Neural Information Processing Systems (NIPS 2014), pp. 2672–2680.

[2] Radford, A., Metz, L., Chintala,S., ”Unsupervised Representational Learning with Deep Convolutional Generative Adversarial Networks”, Conference paper at ICLR 2016, Nov. 2015.

[3] Le, Q., V., Ranzato, M., Monga, R., Devin, M., Chen, K., Corrado, G.,Dean, J., Ng, A., Y., ”Building High-level Features Using Large Scale Unsupervised Learning”, 29th International Conference on Machine Learning, Edinburgh, Scotland, UK, 2012.

[4] Jost, Z., ”Generative Adversarial Networks - Part II”, https://towardsdatascience.com/generative-adversarial-networks-part-ii-6212f7755c1f, Nov. 2017

[5] Shorten, C., ”Deeper into DCGANs.”, https://towardsdatascience.com/deeper-into-dcgans-2556dbd0baac, Feb. 2019.

[6] Umezawa, K., ”DCGAN: Generate the images with Deep Convolutional GAN.” https://medium.com/@keisukeumezawa/dcgan-generate-the-images-with-deep-convolutinal-gan-55edf947c34b, Apr. 2018.

[7] Seo, J.,D., ”[Google / ICML /Paper Summary] Building High-level Features Using Large Scale Unsupervised Learning”, https://towardsdatascience.com/google-icml-paper-summary-building-high-level-features-using-large-scale-unsupervised-fa1ae8fb8678, 2018.

[8] Colyer, ”Unsupervised learning and GANs.” https://blog.acolyer.org/2017/03/02/unsupervised-learning-and-gans/, Mar. 2017.

[9] Gluon, ”Deep Convolutional Generative Adversarial Networks”, https://gluon.mxnet.io/chapter14generative-adversarial-networks/dcgan.html, 2017.

[10] Deng, J., Dong, W., Socher, R., Li, L., Li, K., and Fei-Fei, L.. ”Imagenet: A large-scale hierarchical image database.”, In Computer Vision and Pattern Recognition, 2009. CVPR 2009.IEEE Conference on, pp. 248–255. IEEE, 2009

[11] Yu, F., Seff, A., Zhang, Y., Song, S., Funkhouser, T., Xiao, J., ”Construction of a large-scale image dataset using deep learning with humans in the loop.”, arXiv:1506.03365v3 [cs.CV], Jun. 2016

[12] Gowda, ”DCGAN Implementation in Keras explained”, https://medium.com/@ramyahrgowda/dcgan-implementation-in-keras-explained-e1918fc930ea, Apr. 2019

9 PLAGIARISM REPORT

URKUND

Urkund Analysis Result

Analysed Document: STC_31436_Nachiket_Erlekar.pdf (D67972272)
Submitted: 4/12/2020 9:53:00 AM
Submitted By: rvbidwe@pict.edu
Significance: 21 %

Sources included in the report:

https://machinelearningmastery.com/impressive-applications-of-generative-adversarial-networks/
https://towardsdatascience.com/dcgans-deep-convolutional-generative-adversarial-networks-c7f392c2c8f8
https://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf
https://chainer-colab-notebook.readthedocs.io/en/latest/notebook/official_example/dcgan.html
https://www.mdpi.com/2076-3417/10/5/1729/pdf
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6430964/
https://blog.usejournal.com/introduction-to-gans-and-its-common-applications-29c8f22bc2c6
https://towardsdatascience.com/must-read-papers-on-gans-b665bbae3317
https://blog.acolyer.org/2017/03/02/unsupervised-learning-and-gans/,
https://arxiv.org/pdf/1511.06434
https://www.semion.io/doc/unsupervised-representation-learning-with-deep-convolutional-generative-adversarial-networks
https://medium.com/@liyin2015/dcgan-79af14a1c247
https://www.hann.wang/2017/01/13/gan-visual-recognition-summary/

Instances where selected sources appear:

32