GENERATIVE ADVERSARIAL NETWORKS

A SEMINAR REPORT

submitted by

SANTHISENAN A

TVE15CS050

to
the APJ Abdul Kalam Kerala Technological University
in partial fulfilment of the requirements for the award of the degree

of

Bachelor of Technology
In
Computer Science and Engineering

Department of Computer Science and Engineering

College of Engineering Trivandrum
Thiruvananthapuram

SEPTEMBER 24, 2018

DECLARATION

I undersigned hereby declare that the seminar report "Generative Adversarial Networks", submitted for partial fulfilment of the requirements for the award of degree of Bachelor of Technology of the APJ Abdul Kalam Technological University, Kerala is a bona fide work done by me under supervision of Dr. Ajeesh Ramanujan. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place
Date

Santhisenan A

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
COLLEGE OF ENGINEERING TRIVANDRUM

CERTIFICATE

This is to certify that this report entitled "Generative Adversarial Networks" is a bona fide record of the seminar presented by ***Santhisenan A, Roll No. TVE15CS050*** *under my guidance towards the partial fulfilment of the requirements for the award of Bachelor of Technology in Computer Science and Engineering of Kerala Technological University.*

Dr Ajeesh Ramanujan
Assistant Professor
Dept. of CSE
College of Engineering Trivandrum

Prof Vipin Vasu A V
Assistant Professor
Dept. of CSE
College of Engineering Trivandrum

Dr Shreelekshmi R
Head of the Department
Dept. of CSE
College of Engineering Trivandrum

ABSTRACT

Generative adversarial networks (GANs) is a type of generative model. A GAN consist of two neural networks, a generator network and a discriminator network. The generator outputs data, taking as input a random noise. The discriminator has to classify whether data given as input to it is a real data from the database or a fake one created by the generator. The generator competes against its adversary, the discriminator. As the training for this generative model progresses, the discriminator learns to classify the fake data accurately, while the generator learns to create realistic samples. An equilibrium is reached when the data created by the generator is indistinguishable from real data. GANs are frequently used in image generation and, they produce sharp images too. A downside for GAN is that it does not have a well−defined loss function, which makes training GANs difficult. Many variations of GANs have been introduced by researchers all around the globe. GANs are one of the most promising approaches to generative modelling present today and extensive research is being done to explore their potential.

Keywords: Generative algorithms, Adversarial training

List of Figures

Contents

Chapter 1

INTRODUCTION

In Artificial Intelligence (AI), we can think of machine learning (ML) as one of AI's smaller subsets. Machine learning uses statistical techniques to provide computer systems ability to progressively improve its performance on a specific task with data without being explicitly programmed. [14] Unsupervised learning algorithms are a subset of machine learning algorithms which tries to describe the structure of unlabelled data.

Generative models [4] is an approach to unsupervised learning. The goal of a generative model is to generate data similar to the ones in the dataset. Generative Adversarial Network (GAN) is a type of Generative Model. Other types of generative models include Variational Autoencoders (VAEs) and autoregressive models like PixelRNN. GANs have been successfully applied to solve problems in various domains like generating images, videos and audio, text to image synthesis etc.

GANs were originally introduced by Ian Goodfellow and his collaborators in University of Montreal in 2014 [5]. Yann LeCun, Director of AI Research at Facebook and Professor at NYU called adversarial training as "the most interesting idea in the last 10 years in ML" [9]. In this chapter, this report goes through generative classification algorithms.

1.1 Generative Classification Algorithms

Consider a classification problem in which it is required to classify an input image as that of either a cat $(y = 0)$ or a dog $(y = 1)$. What an algorithm like logistic regression will try to do is that it will try to find a decision boundary that separates images of cats from that of dogs. Then to classify an image as that of a cat or a dog, the algorithm attempts to find out on which side of the decision boundary does the new image fall.

We can attempt to solve the above classification problem in a slightly different way. First, going through all the cat images, we can try to build a model of what a cat looks like and then create a separate model for dogs. Later to classify a new animal using this model, we first match the input image against the cat model and then against the dog model. If the picture looks more like the dogs, the classifier outputs $y = 1$ and $y = 0$ otherwise.

Discriminative learning algorithms try to learn $p(y|x)$ directly. These algorithms (such as logistic regression) learn the mapping directly from the input vector X to the set of labels $\{0,1\}$. The second category of algorithms that try to model $p(x|y)$ and $p(y)$ is called generative learning algorithms. If y indicates whether an example is a dog (1) or a cat (0), then $p(x|y=1)$ models the distribution of dogs' features and $p(x|y=0)$ models the distribution of cats' features.

In Bayesian statistical inference, a prior probability distribution of an uncertain quantity is the probability distribution that expresses one's beliefs about this quantity before taking evidence into account. Here $p(y)$ is the class prior.

After modelling $p(x|y)$ and $p(y)$, our algorithm uses the Bayes rule to derive the posterior distribution on y given x [11] :

$$p(y|x) = \tfrac{p(x|y)p(y)}{p(x)}.$$

Here, the denominator is given by $p(x) = p(x|y=1)p(y=1) + p(x|y=0)p(y=0)$.

According to OpenAI [4], generative models are one of the most promising approaches towards understanding the enormous amount of data out there in the real world. Generative models are trained on immense quantities of data from a specific domain to generate data like it. The intuition behind this approach follows a famous quote from Richard Feynman *"What I cannot create, I do not understand"* [1].

Chapter 2

GAN: THE IDEA

A GAN comprises two components, a generator (G) and a discriminator (D). The goal of the generator model is to produce new data similar to the required one. The discriminator's task is to classify the data presented to it as real or fake. Real data belong to the original dataset, and fake data are those forged by the generator. The generative model competes against its adversary, the discriminative model that learns to determine whether a sample is from the model distribution or the data distribution.

2.1 Analogy

Generative networks can be thought of as a team of counterfeiters trying to produce fake currency notes and use it without being caught by the police. Here discriminative networks play the role of police trying to detect fake currency. Initially, both the police and counterfeiters are not very experienced, but as the game between them progresses, both parties master what they were doing. The game continues until the fake currency notes produced by the counterfeiters are indistinguishable from real currency.

2.2 A Mathematical Model

Assume that the generator represented by the neural network $G(z, \theta_1)$ converts the input noise z into the required output space. Conversely, a second neural network $D(x, \theta_2)$ represents discriminator, and it calculates the probability that x came from the real dataset. Here, θ_1 and θ_2 represents the weights that describe each neural network.

The discriminator is trained to classify the input data as either real or fake. The weights of the discriminator are updated so that it maximizes the probability that real images from the database are classified as real and minimizes the probability that images generated by G are classified as fake. The loss function used for the discriminator maximizes $D(x)$ and minimizes $D(G(z))$.

The generator's weights are trained to maximise the probability that it can fool the discriminator using the images it generates. The loss function maximizes $D(G(z))$.

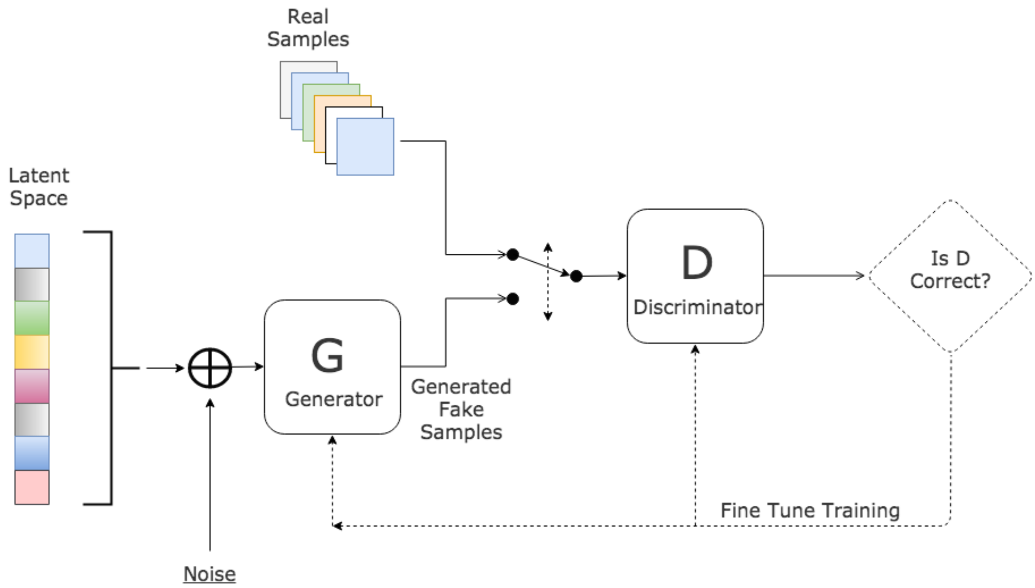

Figure 2.1: Generative adversarial networks [3]

Figure 2.1 shows a visual representation of the high level overview of GANs discussed in this section. During training, the generator is trying to maximize $D(G(z))$ and discriminator is trying to minimize $D(G(z))$. Thus we can think of the scenario as the generator and discriminator as playing a minimax game.

Chapter 3

VARIATIONS OF GAN

Ian Goodfellow and his collaborators introduced GANs in 2014. Since then they were studied extensively by researchers around the globe. As a result, more than 450 different types of GANs have been proposed by researchers [15].

The idea of GAN has been applied extensively in many domains, which resulted in a variety of interesting deep learning models. GANs have been used for "image to image" translation. Image-to-image translation is a class of vision and graphics problems where the goal is to learn the mapping between an input image and an output image using a training set of aligned image pairs. CycleGAN [19] transfers pictures from one domain to another.

It is also capable of performing artistic style transfer, adding bokeh effect to phone camera photos, creating outline maps from satellite images or convert zebras to horses and vice versa, to list a few [17]. Figure 3.1 shows CycleGAN transforming Monet paintings to landscape images.

Input Output

Figure 3.1: CycleGAN mapping Monet paintings (input) to landscape photographs (output) [17]

In the following sections, the report gives a brief overview of some of the most prominent applications of GAN. Section 3.1 discusses text-to-image synthesis using GAN, and section 3.2 describes a GAN for image super resolution.

3.1 Generative Adversarial Text to Image Synthesis

Advancements in Recurrent Neural Networks has enabled the creation of powerful architectures to learn discriminative text feature representations. Meanwhile, Deep Convolutional GANs (DCGANs) have begun to create highly realistic images belonging to specific categories. Generative Adversarial Text to Image Synthesis [12] bridges these advancements in text and image modelling. The architecture aims to transform human descriptions of images in single sentences directly into image pixels (See figure 3.2).

Text to image synthesis is a challenging problem consisting of two parts: first, learn a text feature representation that captures the essential visual details and second, synthesise an image that seems real to a human. Fortunately, deep learning has made significant progress in both areas. However, one difficult issue not solved by deep learning is that there exist many plausible combinations of pixels that correctly illustrate the definition. By conditioning both the discriminator and generator on side information, we can naturally model this phenomenon as the discriminator acts as a smart adaptive loss function.

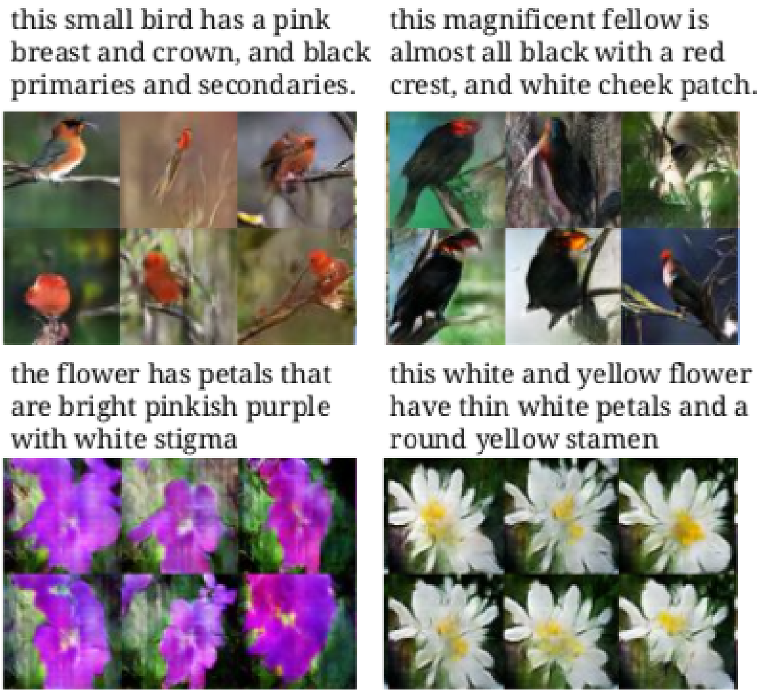

Figure 3.2: Images generated by the GAN along with the text description used to generate them [12]

3.2 SRGAN

Estimation of the high-resolution image from its low-resolution counterpart is called super-resolution (SR). The goal of supervised SR algorithms is to reduce the Mean Squared Error (MSE) between the generated high-resolution image and the target image. Minimizing MSE also maximises Peak Signal to Noise Ratio (PSNR). MSE and PSNR are the most common methods used to judge SR algorithms. As MSE and PSNR are evaluated on pixel-wise differences, they fail to identify perceptually relevant differences such as high texture detail.

SRGAN [10] employ dee residual network with skip-connection. It defines a novel perceptual loss using high-level feature maps of VGG network, and a discriminator that favours solutions perceptually hard to differentiate from high-resolution reference images. As figure 3.3, suggests, the image produced by SRGAN is able to produce sharper images when compared to other networks.

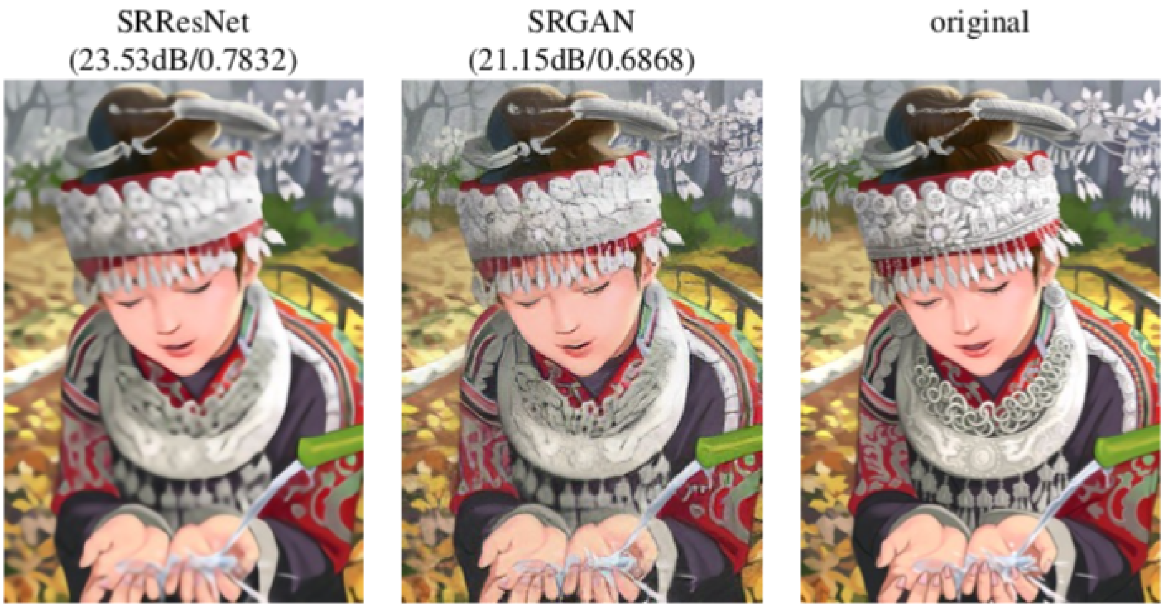

Figure 3.3: From left to right: bicubic interpolation, deep residual network optimized for MSE, deep residual generative adversarial network optimized for a loss more sensitive to human perception, original HR image. Corresponding PSNR and SSIM are shown in brackets. (4 x upscaling) [10]

In chapter 4, the report describes another prominent approach to generative modelling, the variational autoencoders.

Chapter 4

VARIATIONAL AUTOENCODERS

4.1 The Standard Autoencoder

Autoencoders are unsupervised neural networks that can learn efficient data encodings. Autoencoders learn the representation for the data, typically for dimensionality reduction. Initially, an autoencoder compresses the input data into a latent vector. The latent vector can be used to regenerate the input data later. Traditionally, autoencoders were used for dimensionality reduction and feature learning. In the recent times, autoencoders are being used to learn generative models of data.

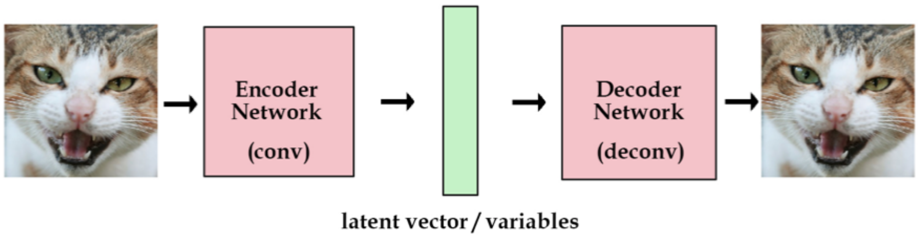

Figure 4.1: Autoencoders [18]

As shown in figure 4.1, the autoencoder has two component neural networks, an encoder and a decoder. The encoder neural network can be represented by $h = f(x)$ and it takes the input image and encodes it into a latent vector. The decoder, $r = g(h)$ then produces a reconstruction. An autoencoder will be not useful if it simply replicates the input given to it. Thus, the autoencoder is not allowed to copy the input directly instead it learns useful parameters from the input [6].

4.2 Variational Autoencoder

We cannot create a new image using a standard autoencoder since the latent vectors are obtained by encoding an input image. We can make an autoencoder create new images by adding a constraint to the encoder neural network. A variational autoencoder differs from a standard one by the fact that the encoder of the variational autoencoder is forced to create latent vectors that roughly follow a unit Gaussian distribution. For generating new images, we can sample new latent vectors from the Gaussian distribution and pass it onto the decoder. In practice, there is a tradeoff between how exactly the sampled latent vector matches the Gaussian distribution and how accurate our network is.

The loss function of the autoencoder has two components. They are the generative loss and the latent loss. The generative loss estimates the mean squared error of how accurately the network re-generated the image. The latent loss is the Kullback−Leibler divergence [1] that measures how closely the latent variables match the Gaussian distribution [18].

generation loss = *mean squared error between generated image and real image*
latent loss = *KL-Divergence between latent variable and unit Gaussian*
total loss = *generation loss* + *latent loss*

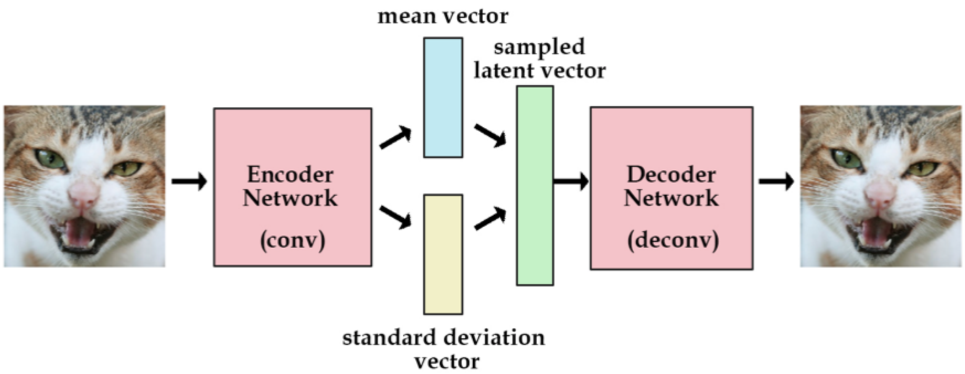

Figure 4.2: Optimizing KL divergence for variational autoencoders [18]

As shown in figure 4.2, if the encoder generates a vector of means and standard deviations, we can optimize KL divergence. This report will discuss the performance of Variational Autoencoders and its comparison with GANs in chapter 5.

[1] In mathematical statistics, Kullback−Leibler divergence (also called as relative entropy) measures how one probability distribution differs from another. [8]

Chapter 5

CONCLUSION

In this chapter, this report compares the two approaches to generative models described in chapters 2 and 4, namely Generative Adversarial Networks and Variational Autoencoders.

When it comes to images, GANs in general, produce crispier images than VAEs. GANs use an adversarial method to train the network. The discriminator forces the generator to create better images that look more realistic. Meanwhile, VAEs use an user-defined loss function. This gives GANs an edge over VAEs when it comes to generating new images. Unlike variational methods, GANs does not introduce any deterministic bias. This seems to be the reason why variational methods produce blurry images [7].

As GANs does not have an easily understood loss function, training a GAN may to be prove tricky. Unlike standard loss functions like squared error or log-loss, the loss function that GANs try to optimize does not have any closed form. Thus during training, a lot of trial and error is required to determine the network structure and the training protocol. This is why GANs have not been applied yet to more complex data such as text or voices [2].

GANs are a relatively new model and, it is expected to see rapid improvements to GANs soon. A semi-supervised approach to GANs [13] introduced by researched at OpenAI has achieved remarkable results. This novel approach included the discriminator producing an additional output indicating the label of the input. On the MNIST[16] dataset, 99.14% accuracy was achieved by using only 10 labelled examples per class in a fully connected neural network [4].

Thus both GANs and VAEs have their own pros and cons. Both the approaches to generative models are equally promising and both are undergoing extensive research.

Bibliography

[1] On his blackboard at time of death in 1988; as quoted in The Universe in a Nutshell by Stephen Hawking.

[2] *Chatavut Viriyasuthee's answer to What are the pros and cons of using generative adversarial networks (a type of neural network)? Could they be applied to things like audio waveform via RNN? Why or why not? on Quora*. https://www.quora.com/What-are-the-pros-and-cons-of-using-generative-adversarial-networks-a-type-of-neural-network-Could-they-be-applied-to-things-like-audio-waveform-via-RNN-Why-or-why-not. Accessed: 17-09-2018.

[3] *Generative Adversarial Networks: Hot Topic in Machine Learning*. https://www.kdnuggets.com/2017/01/generative-adversarial-networks-hot-topic-machine-learning.html. Accessed: 10-09-2018.

[4] *Generative Models*. https://blog.openai.com/generative-models/. Accessed: 09-09-2018.

[5] I. J. Goodfellow et al. "Generative Adversarial Networks". In: *ArXiv e-prints* (June 2014). arXiv: 1406.2661 [stat.ML].

[6] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[7] *Ian Goodfellow's answer to What are the pros and cons of using generative adversarial networks (a type of neural network)? Could they be applied to things like audio waveform via RNN? Why or why not? on Quora*. https://www.quora.com/What-are-the-pros-and-cons-of-using-generative-adversarial-networks-a-type-of-neural-network-Could-they-be-applied-to-things-like-audio-waveform-via-RNN-Why-or-why-not. Accessed: 17-09-2018.

[8] *Kullback Leibler divergence*. https://en.wikipedia.org/wiki/Kullback\x{fffd}\x{fffd}\x{fffd}Leibler_divergence. Date Accessed: 17-09-2018.

[9] Yann LeCun. *Answer to What are some recent and potentially upcoming breakthroughs in deep learning?* https://www.quora.com/What-are-some-recent-and-potentially-upcoming-breakthroughs-in-deep-learning. Accessed: 09-09-2018. 2016.

[10] Christian Ledig et al. "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network". In: *CoRR* abs/1609.04802 (2016). arXiv: 1609.04802. URL: http://arxiv.org/abs/1609.04802.

[11] Andrew Y Ng and Michael I Jordan. "On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes". In: *Advances in neural information processing systems*. 2002, pp. 841–848.

[12] Scott E. Reed et al. "Generative Adversarial Text to Image Synthesis". In: *CoRR* abs/1605.05396 (2016). arXiv: 1605.05396. URL: http://arxiv.org/abs/1605.05396.

[13] Tim Salimans et al. "Improved Techniques for Training GANs". In: *CoRR* abs/1606.03498 (2016). arXiv: 1606.03498. URL: http://arxiv.org/abs/1606.03498.

[14] Arthur L. Samuel. "Some studies in machine learning using the game of Checkers". In: *IBM JOURNAL OF RESEARCH AND DEVELOPMENT* (1959), pp. 71–105.

[15] *The GAN Zoo on GitHub*. https://github.com/hindupuravinash/the-gan-zoo/blob/master/gans.tsv. Accessed: 10-09-2018.

[16] *THE MNIST DATABASE of handwritten digits*. http://yann.lecun.com/exdb/mnist/. Accessed: 17-09-2018.

[17] *Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks*. https://junyanz.github.io/CycleGAN/. Accessed: 10-09-2018.

[18] *Variational Autoencoders Explained*. http://kvfrans.com/variational-autoencoders-explained/. Accessed: 17-09-2018.

[19] J. Zhu et al. "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks". In: *2017 IEEE International Conference on Computer Vision (ICCV)*. Oct. 2017, pp. 2242–2251. DOI: 10.1109/ICCV.2017.244.