Interactive Elearning Application for Cycle-Consistent Adversarial Networks

Marilena-Catalina Dragomir[1], Valentin Pupezescu[2]

(1) Department of Applied Electronics and Information Engineering, Faculty of Electronics, Telecommunications and Information Technology, ETTI, Bucharest, Romania, E-mail: catalina.dragomir[at]protonmail.com
(2) Department of Applied Electronics and Information Engineering, Faculty of Electronics, Telecommunications and Information Technology, ETTI, Bucharest, Romania, E-mail: valentin.pupezescu[at]upb.ro

Abstract

The COVID-19 pandemic has disrupted the education systems worldwide and in this context the virtual learning software solutions have been of critical importance in supporting the continuous learning process when the social distancing makes very difficult the physical attendance for classes. In order to support the learning process for the students in the domain of generative modelling we have developed an interactive web application that streamlines the process of understanding Generative Adversarial Networks (GANs) and Cycle-Consistent Adversarial Networks (CycleGANs), a class of unsupervised machine learning models that enables the generation of synthetic images that follow the statistical distribution of the images in the training data set. Our work consists in training from scratch a CycleGAN model and in designing and building a responsive web interface that allows the students to experiment with an application of the CycleGANs, the style transfer. We strive to inspire the students and to motivate them to deepen their knowledge in this area by implementing an interactive web application and by structuring it into two major sections: the first section is oriented towards a practical approach and we encourage the students to upload a photo at their choice and analyse the result of applying the style transfer from the images in the training data set to their uploaded image. The second part emphasizes the theoretical part and introduces the students to the concepts behind GANs and CycleGANs.

Keywords: Generative Adversarial Networks, Cycle-Consistent Adversarial Networks, Virtual learning, Responsive web interface

1 Introduction

In order to support the online learning in the context of the Covid-19 pandemic we have implemented an interactive application that helps the students interested in learning more about the GANs and CycleGANs models by providing them a tool that allows them to experiment with an application of the CycleGANs, the style transfer.

1.1 Generative Adversarial Networks

Generative Adversarial Networks (GANs) (Goodfellow et. al., 2014) are an important class of generative models and they belong to the unsupervised learning field. They were introduced by Ian J. Goodfellow et. al. in the paper “Generative Adversarial Nets” in 2014 (Goodfellow et. al., 2014) and they represent a turning point in the evolution of the generative models.

The GANs framework consists of 2 intrinsic parts that are modelled as fully differentiable deep neural networks: Generator (G) and Discriminator (D) (Fig. 1).

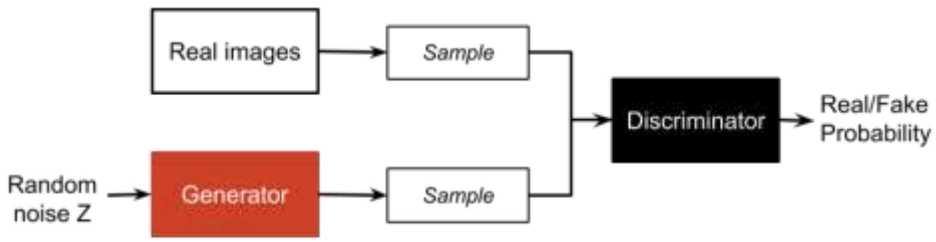

Fig. 1. GAN architecture

The core idea that lies at the heart of the great success of the GANs is the adversarial training between the two networks. They play a two player minimax game with the value function V(D, G) illustrated in Fig. 2.

D and G are trained simultaneously using the backpropagation algorithm and they compete with each other: D is trained to maximize the probability of correctly classifying the samples applied to its input, it maximizes the term log(D(x)) from V(D, G). G is trained to minimize the only term where it contributs in the mathematical expression ofV(D, G): log(1 – D(G(x))).

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

Fig. 2. GANs value function [1]

The Generator network produces synthetic samples from noise (Z) and its mission is to continously improve the generated data until it becomes indistinguishable from the training data. The Discriminator network is a classifier that has the goal to correctly classify the synthetic data from the real data. It outputs the probability that the input data is real of fake.

The training ends when none of the two competing networks can improve anymore, they reach the Nash equilibrium (Goodfellow et. al., 2014)when the distribution of the synthetically generated samples converges to the distribution of the real data set. At this point, the Generator producessamples that follow the distribution of the original data set, whilst the discriminator can no longer distinguish between the original or synthetically generated data, it outputs the probability 0.5.

1.2 Cycle-Consistent Adversarial Networks

Cycle-Consistent Adversarial Networks (CycleGANs) were proposed in the paper “Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks” (Jan-Yan Zhu et. al., 2017).

CycleGANs models belong to the GAN family, but they have architectural particularities that enable image-to-image translation between two domains, X and Y, in the absence of paired images for training.

A CycleGAN model is composed of two generators (G and F) and two discriminators (D_X and D_Y) as illustrated in Fig. 3.

D_x tries to distinguish between the images in the X domain and the translated images from the Y domain, F(y). D_y attempts to distinguish between the images in the Y domain and the translated images from the X domain, G(x).

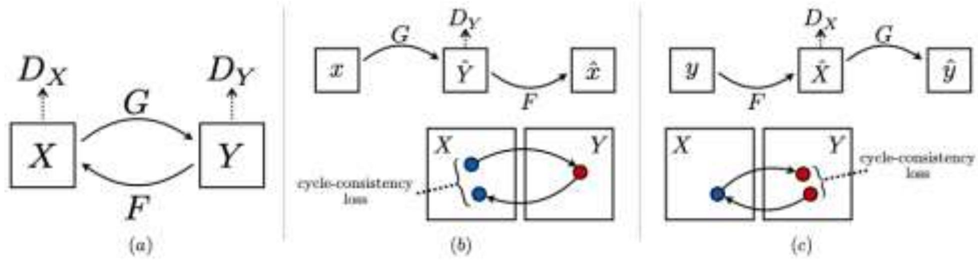

Fig. 3 CycleGAN (Jan-Yan Zhu et. al., 2017)

As it can be observed in Fig. 3, G will translate an image from domain X to domain Y, whilst F will perform the opposite translation between the domains, it will translate an image from domain Y to domain X. With this setup, the following assumptions apply: for an image x that belongs to domain X, the function F(G(y)) = y. The same reasoning applies to the images in the domain Y: G(F(x)) = x.

In order to achieve the mapping from the X to Y domain, the objective function of the CycleGAN model [1] has additional constraints comparing to the traditional GAN models, applied in the form of the Cycle consistency loss [3].

[1]

$$G^*, F^* = \arg\min_{G,F}\max_{D_x,D_Y} \mathcal{L}(G,F,D_X,D_Y)$$

$$\begin{aligned}\mathcal{L}(G,F,D_X,D_Y) =& \mathcal{L}_{\text{GAN}}(G,D_Y,X,Y) \\ &+ \mathcal{L}_{\text{GAN}}(F,D_X,Y,X) \\ &+ \lambda\mathcal{L}_{\text{cyc}}(G,F),\end{aligned}$$

[2]

$$\begin{aligned}\mathcal{L}_{\text{GAN}}(G,D_Y,X,Y) =& \mathbb{E}_{y\sim p_{\text{data}}(y)}[\log D_Y(y)] \\ &+ \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log(1-D_Y(G(x))]\end{aligned}$$

[3]

$$\begin{aligned}\mathcal{L}_{\text{cyc}}(G,F) =& \mathbb{E}_{x\sim p_{\text{data}}(x)}[\|F(G(x))-x\|_1] \\ &+ \mathbb{E}_{y\sim p_{\text{data}}(y)}[\|G(F(y))-y\|_1].\end{aligned}$$

Fig. 4. CycleGAN loss functions (Jan-Yan Zhu et. al., 2017)

The loss functions that are involved in the training process of the CycleGAN model are highlighted in Fig. 4: [1] reflects the full CycleGAN objective function, [2] reflects the adversarial loss for the mapping functions, F and G. [3] Defines the CycleConsistency loss, it has a very important contribution to the total loss function because it enforces that the cycle consistency property is maintained, so that a translated image can be mapped back to the original image.

2 Our work

In this paper we present a multifunctional tool that helps the students interact with the GANs: on the one hand they can experiment with the style transfer from pictures painted by Claude Monet toimages at their choice and see the generated results in real time. On the other hand they can expand their knowledge on the subject with the help of the documentation section.

2.1 CycleGAN model

The network architecture that we used for the CycleGAN follows the implementation described in the "Transforming the World Into Paintings with CycleGAN"web article (Sebastian Theiler, web article), where the generator network is slightly different from the original architecture from (Zhu et. al., 2017). The generator was modelled using a pix2pix architecture (Isola et. Al, 2017), a modified version of the U-Net architecture. (Ronneberger, 2015)(Sebastian Theiler , web article) We trained from scratch the CycleGAN model for 50 epochs. The data set used for training the generative model was the monet2photo data set (291.09 MiB), available in TensorFlow Datasets collection. Fig. 5 highlights the architectural details that define the discriminator and generator networks. After training was complete we saved the model in Keras H5 format and we used the trained model to power the interactive application to generate new images translated to Monet style.

2.2 Interactive application

We designed and implemented a responsive web application using the HTML5 markup language for structuring the content as well as Bootstrap 4, a popular frontend framework that allowed customizing the layout through the use JavaScript plugins and extensive components which are essential for building responsive websites. The backend that powers the functionality was built using the Flask micro web framework, it is written in Python and allowed us to define the routes for the application (index, results and documentation)and the associated content to be served (views).

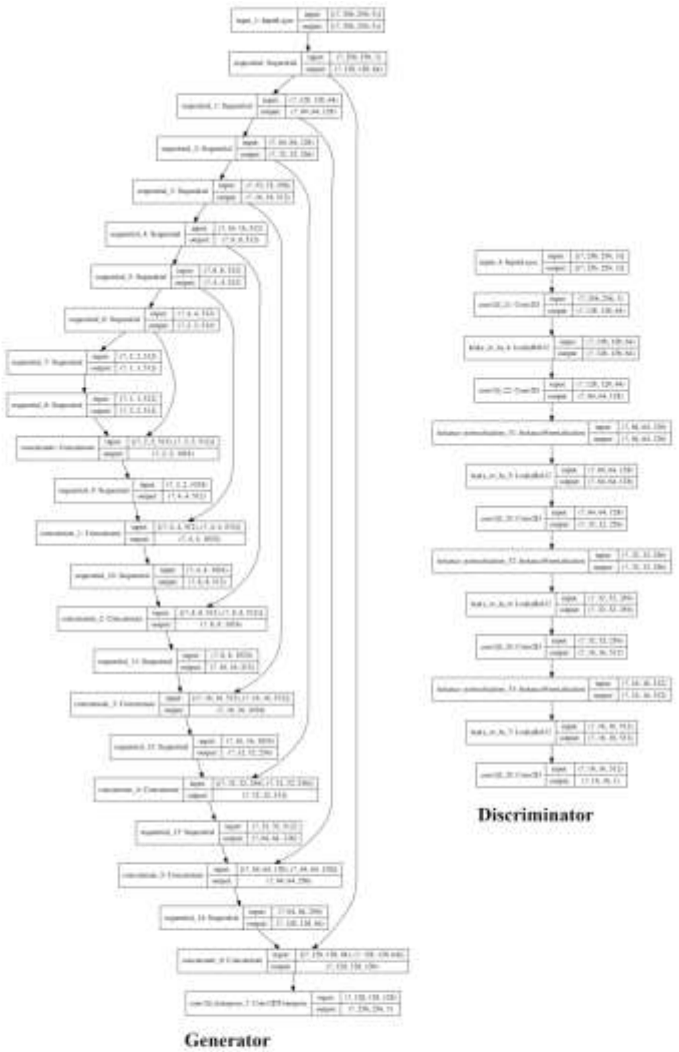

Fig. 5 Architectures for Generator and Discriminator networks

The web application is composed of two major sections. The first section is dedicated to the practical approch for learning an application of a CycleGAN model, the style transfer from the images in the data set to a new image. In this section, the students can upload an image of their choice and see, the original image as well as the generated image “painted” in the Claude Monet style, as illustrated in Fig. 6.

When an image is uploaded, a POST request is sent to the /results route. The request is processed, the image uploaded is temporary stored using the BytesIO buffer module from Python. Before being fed to the neural network the image is preprocessed: it is resized to 256x256 px and normalised. After the generative model predicts the image, it is converted to base64 encoding and set as image source in the HTML template.

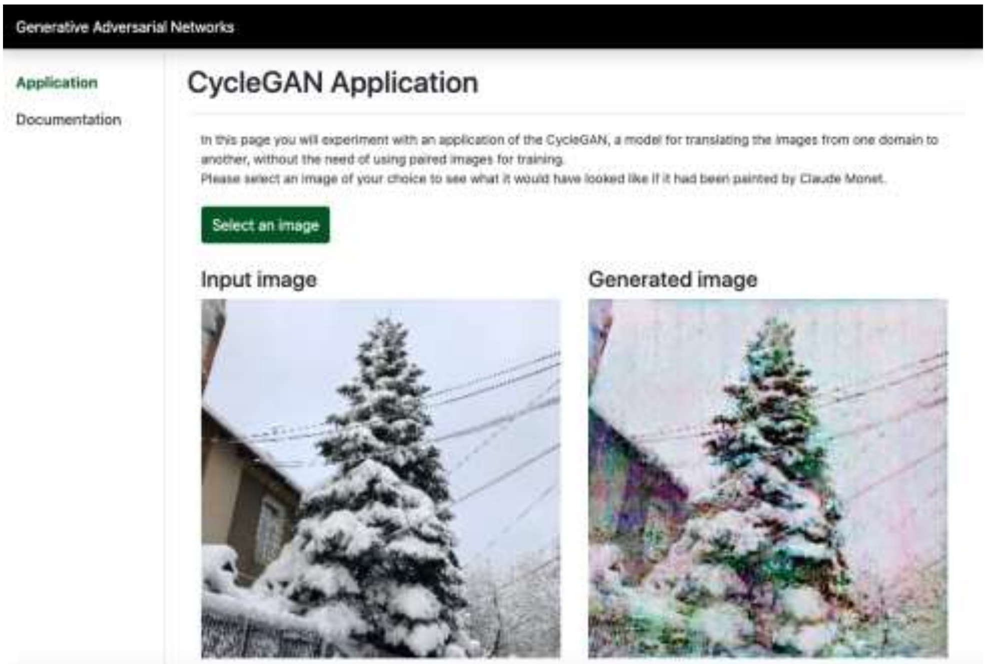

Fig 6. Application section

The second section, illustrated in Fig. 7, is oriented towards the theoretical concepts behind GAN and CycleGAN frameworks and encourages the students to delve into the theory that powers the generative models that make possible, but not limited to, the image translation from one domain to another, as they experimented in the Application page.

The students can easily navigate through through the fundamental topics related to GANs and CycleGANs models using the left side menu to learn more about: GANs basic notions, GANs training process, GANs objective function as well as the theory behind the CycleGANs, CycleGANs training process and the particularities of the CycleGAN objective function.

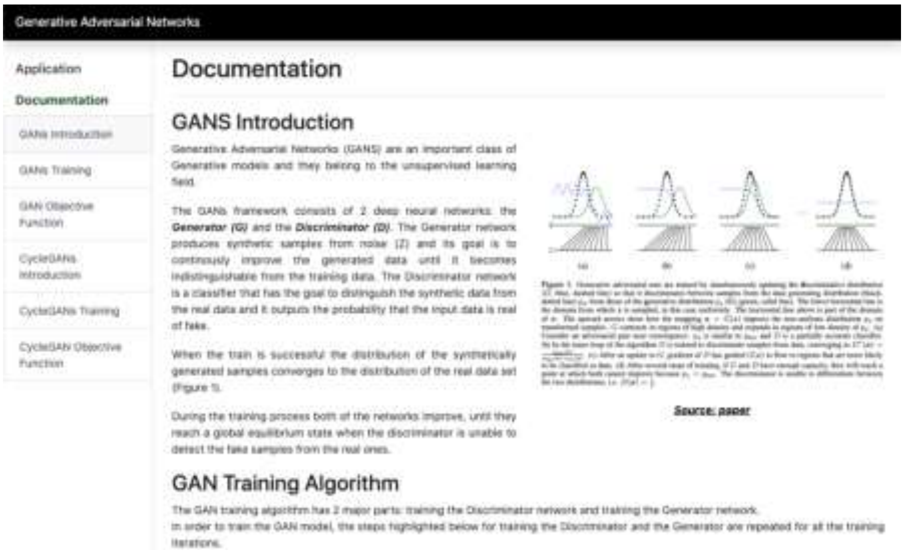

Fig. 7 Documentation section

Conclusions

In this paper we described an interactive web application that we designed and implemented from scratch whichemphasizes an application of a CycleGAN model: the style transfer between two domains. This application is dedicated to the students that want to expand their knowledge about the generative adversarial networks and their impressive applications. We described the theoretical notions behind the GANs and CycleGANs, we presented the architecture of the CycleGAN model that was trained from scratch using the monet2photo data set and was used to generate the new images in the Monet style in the app. We also highlighted the frontend and the backend technologies used to build the web application.

In the next version of the interactive tool we will enhance it to include more applications of the generative adversarial networks.

References

Goodfellow, I. and Pouget-Abadie, J. and Mirza M. and Xu B. and Warde-Farley D. and Ozair S. and Courville, A. and Bengio, Y. (2014): Generative Adversarial Nets, in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2,* Montreal, Canada, 2672–2680.

Isola, P and Zhu, J. and Zhou, T and Efros, A. (2017), Image-to-Image Translation with Conditional Adversarial Networks, *CVPR.*

Ronneberger, O. and Fischer, P. and Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In LNCS. 9351. 234-241. 10.1007/978-3-319-24574-4_28.

Zhu, J.* and Park, T.* and Isola, P. and Efros, A. (2017): Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks, in *IEEE International Conference on Computer Vision (ICCV)*, 2017.

https://medium.com/analytics-vidhya/transforming-the-world-into-paintings-with-cyclegan-6748c0b85632, 2020