Using Generative Neural Networks to generate artist-inspired artwork

Mihai Bojescu
Master in Artificial Intelligence and optimisation
Faculty of Computer Science
University "Alexandru Ioan Cuza" of Iași
Iași, Romania
bojescu.mihai@gmail.com

Radu Șolcă
Master in Artificial Intelligence and optimisation
Faculty of Computer Science
University "Alexandru Ioan Cuza" of Iași
Iași, Romania
radu.ssolca@gmail.com

***Abstract*—This document contains a study on how Generative Neural Networks could be applied in order to transform noise into artwork inspired by renoun artists. The architecture of the Network consists of a Convolutional Neural Network for the discriminator and a U-Net Convolutional Neural Network for the genetrator. The data on which the network is trained is a dataset with faimous paintings by Jean Monnet.**

***Index Terms*—Generative Neural Networks, GANs, Convolutional Neural Networks, U-Nets.**

I. Introduction

In this document, we will study how GANs could be used to generate pieces of art in the Jean Monnet style. GANs have shown remarkable success in generating realistic images by using random noise as input, and we aim to use them as an application for producing fine art.

Monnet's paintings are characterised by their vibrant colors and expressive brushwork, and we will try to replicate this signature in the pictures that the GAN produces, effectively creating a 'digital Monnet'. The model will play by rotation the role of an art critic and an artist in order to improve itself until - as an art critic - it cannot distinguish between real, pre-existing art and newly created art.

During the building of this model, several intriguing questions were raised, ranging from the possibility of intersecting arts with artificial intelligence, to the creativity and the originality of said results. What would the implications of using such models in the industry of arts be? This paper also will explore the ethical implications of computer-generated craft.

II. GANs

Generative Adversarial Networks, a type of Neural Network that was originally built in 2014 by Ian Goodfellow and his colleagues in June 2014 [1] is a type of machine learning system that involves two neural networks competing against each other. The competition can be seen a game where two teams have to play a zero-sum game: one team's win is the other team's loss. The system is part of the unsupervised learning category, but can also be used for semi-supervised learning, fully-supervised learning and for reinforcement learning. The alforementioned system can be used for a vast set of use-cases, ranging from image generation - the classical use-case - to music generation.

In our paper, we present a classical use-case for GANs: image generation. As seed, we will be using random noise from the uniform distribution and we will use the seed to generate art in the Monet style.

III. Discriminator

For a discriminator, we will use a classical cone-shaped convolutional neural network that receives an image with 3 channels and in turn creates, 64, 128, 256, 512 feature maps, which are later flattened to a 1-dimensional array. The network makes use of a Leaky Rectified Linear Unit activation function for the feature maps building, and a Sigmoid activation function at its output.

The trainer of the discriminator uses - for each time it is used - a batch of real images and a batch of fake ones, which are used to "teach" the model how to classify images: fake ones are considered negative, while real ones as positive. For training we use the Wasserstein distance - also named Earth Mover's distance - to compute the loss of the discriminator:

$$\begin{aligned} loss_D = &-\frac{\sum_{i=0}^{n} real\_output^i}{n} \\ &+ \frac{\sum_{i=0}^{n} fake\_output^i}{n} \\ &+ gradient\_loss(fake\_image, real\_image) \end{aligned} \quad (1)$$

Where $gradient\_loss$ is a function that computes, given a real and a fake image, a penalty that encourages the gradients of the critic to have norm 1 almost everywhere. This is used in order to enforce the Lipschitz continuity constraint, which in turn should improve the results.

IV. Generator

The generator of the GAN is comprised of a classical U-Net architectured CNN. U-Nets are well known for producing pictures with the same shape as the input, which in our case proved beneficial, as we could provide noise for each pixel of the image, noise. After training, the generator can output realistic-looking Monet-inspired pictures.

The generator uses multiple layers of convolutional and max-pooling layers that downscale the image and extract features, data which is later used to upscale the image back using multiple transposed convolutional and classical convolutional layers. For the activation function we also use the Leaky Rectified Linear Unit function.

V. Shortfalls

GANs are well-known for being unstable in training, and are also well-known for suffering of the mode collapse problem - learning so well the training data of the discriminator, that it becomes a generator that outputs only the training data.

References

[1] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengie (2014). "Generative Adversarial Nets". Département d'informatique et de recherche opérationnelle Université de Montréal Montréal, QC H3C 3J7. arXiv: 1406.2661v1