LaTeX Author Guidelines for CVPR Proceedings

Bradley Beyers
Institution1
Institution1 address
bbeyers@u.rochester.edu

Santiago Loane
Institution2
First line of institution2 address
sloane@u.rochester.edu

Abstract

In this work, we explore the applicability of Generative Adversarial Networks (GANs) to the task of generating novel works of art. Utilizing the Deep Convolutional GAN (DCGAN) architecture detailed by Radford et al. *in [5], we train generative models on a dataset of 203,275 visual works of art. We show empirically that the DCGAN architecture is able to learn and reproduce salient features of visual artwork such as color, shape, composition and style. We more closely examine the representations learned by the generator network by employing smooth interpolation between points in its input space.*

1. Introduction

The comprehension and production of artwork is considered by many to involve skills available uniquely to humans. Human artists incorporate a wealth of cultural knowledge, personal experiences, and creativity in their work. A model which could produce output human observers would accept as novel works of art could be said to possess some representation of the artistic knowledge that humans tap into when they create art. By examining these learned representations, we can learn more about what they are able to encode and how they are able to encode it. In the case of convolutional neural networks, whose architecture is designed to roughly mirror the structures of neurons that make up our brains, analysis of the representations learned by such a model may even yield some insight into the way humans process visual information.

The computer-based generation of images which appear natural has in the past been limited to the rendering of images based on models and textures carefully designed by humans. More recently, deep learning has opened up the possibility of realistic image generation through the use of neural networks. Goodfellow *et al*. [1] introduced Generative Adversarial Networks (GANs), a framework for training generative models using backpropagation. Radford *et al*. [5] present techniques for the effective training of Deep Convolutional GANs (DCGANs), or deep convolutional neural networks within the GAN framework.

By training a DCGAN on a dataset of visual works of art, we create models which can generate images which capture salient properties of artwork produced by human artists. We empirically assess the quality of samples generated by our model and discuss the features it was able to learn. Using analysis techniques discussed by Radford *et al*. in [5] as well as 'deconvnets' discussed by Zeiler and Fergus in [6], we investigate the representations learned by our model more directly.

2. Related Work

2.1. Generative Adversarial Networks

First proposed by Goodfellow *et al*. in [1], GANs are a very general framework which may be used to learn generative models of data distributions. In GANs, two models are employed, a discriminator D and a generator G. G's goal is to produce output which mimics the features of some known data distribution p_{data}, and D's goal is to distinguish real samples drawn from p_{data} from fake samples generated by G. To produce its output, G takes as input a sample z from a noise distribution p_z. Formally, D and G play a minimax game whose value function is defined as

$$\min_G \max_D \mathbb{E}_{x\sim p_{data}(x)}[log(D(x))] + \mathbb{E}_{z\sim p_z}[log(D(G(z)))]. \quad (1)$$

In other words, D attempts to maximize its chances of correctly correctly classifying its input as real or fake, and G attempts to minimize D's chances of doing so. If suitable choices are made for the model architecture of D and G, GANs may be trained using backpropagation.

2.2. Deep Convolutional Generative Adversarial Networks

In [1], Goodfellow *et al*. explored the potential for convolutional neural networks in a GAN framework to learn a generative model of the well-known CIFAR-10 dataset [2].

Radford *et al*. expand on this idea in [5], introducing DCGANs. Their key results include a general set of guidelines for effective training of DCGANs such as the replacement of pooling and unpooling layers with convolutional and transpose convolutional layers respectively, the use of batch normalization layers, and the use of the LeakyReLU activation function [4] in the discriminator network.

2.3. Network Analysis Methods

Radford *et al*. also discuss several methods of investigating the representations learned by the generator network of a DCGAN. One method is walking through the latent space of G's input. If G has managed to learn relevant and meaningful features of p_{data}, then walking through the latent space will yield smooth transitions between semantic concepts in G's output.

Another is searching for evidence that vector arithmetic in G's input space can result in meaningful manipulation of the features of its output. For example, if $z_{glasses}$, z_{man} and z_{woman} are noise vectors which cause G to output images of a man with glasses, a man without glasses, and a woman without glasses respectively, then $z_{glasses} - z_{man} + z_{woman}$ should produce an image of a woman with glasses when passed to G as input. The presence of such structure shows that visual concepts like glasses may be associated with vectors in G's input space.

'Deconvnets' or deconvolutional networks, presented by Zeiler and Fergus in [6], are a technique for visualizing the representations learned by convolutional layers in convolutional neural networks. Deconvnets approach the problem of visualizing the concepts learned by high-level convolutional layers by taking a fully trained convolutional neural network and a choosing a single neuron in a higher layer. An image is passed forward through the network, and then the activations of all neurons other than the target neuron are set to 0. Then, the activation is passed backward through a deconvnet, which reconstructs the activations of the layers preceding that of the target neuron. This progressive reconstruction will eventually reach the input layer, where it will reconstruct the pattern of pixels which caused the activation of the target neuron. By visualizing the reconstructed activations of the intermediate layers and input layer, we can get a sense of the features each neuron in the network examines, as well as the patters in input which tend to activate a particular high-level neuron.

3. Methods

3.1. Data

Before attempting to train a DCGAN on a dataset of artwork, we first validated the architecture of our models by training them on the well-known MNIST dataset of handwritten digits [3]. The advantage to training on MNIST is that the visual features of a well-formed digit are simple and obvious, whereas the visual features of a well-formed painting are difficult to concretely define. As a result, training models on MNIST serves as a useful proof-of-concept, since visual inspection of a model's output can verify whether it is able to learn useful features of the dataset. To conduct our experiments on artwork, we compiled images of visual works of art from two sources, Kaggle and Wikiart.

Kaggle is a platform for hosting data science and machine learning competitions in which teams compete to complete a task defined by those hosting the competition. The Kaggle "Painter By Numbers" competition challenged competitors to predict whether pairs of paintings were created by the same artist. The accompanying dataset contains 79,420 images labeled with a hash of the artist's name, the title of the work, the style, the genre, and the date it was created.

Wikiart is an online encyclopedia of visual artwork maintained by users in a manner similar to Wikipedia. On their "About" page, Wikiart claims to host around 250,000 pieces of artwork from around 3,000 artists. Using a script to scrape images from their site, we were able to gather 123,854 images of paintings and some other kinds of visual artwork. Most, but not all of the images contain most of the same metadata as the Kaggle dataset. 98.29% of images are tagged with their artist, 95.14% are tagged with their genre, and 94.57% are tagged with their style.

Combining these two sets of data, we managed to assemble a dataset of 203,274 paintings and other works of visual artwork, which we then used to train a DCGAN.

3.2. Model Construction

The model architecture we used for our experiments is based very closely on the model architecture for DCGANs presented by Radford *et al*. in [5].

4. Experiments

5. Conclusion

References

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[2] A. Krizhevsky, V. Nair, and G. Hinton. The cifar-10 dataset. *online: http://www. cs. toronto. edu/kriz/cifar. html*, 2014.

[3] Y. LeCun, C. Cortes, and C. Burges. Mnist handwritten digit database. *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, 2, 2010.

[4] A. L. Maas, A. Y. Hannun, and A. Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3, 2013.

[5] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[6] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.