Artificial Creativity

Federico Matteoni
Intelligent Systems for Pattern Recognition
A.Y. 2022/2023
Mat. 530257

***Abstract*—This work is a survey on some common machine learning systems aimed at image generation that surfaced in the last years. These systems have recently been used by the general public in viral fashion to generate artistic content, ranging from artistic repropositions of submitted photos to completely new concepts produced by text-based prompts. Although exceptional, these systems also raised some concerns regarding copyright and intellectual property.**

I. Introduction

In January 2021, OpenAI labs presented DALL-E [14], a text-to-image generative autoregressive transformer to the public, followed by an unofficial demo. In that demo, users could simply type a prompt and, after less than few minutes, a small selection of artistic interpretations of that prompt would be presented. While not being the first text-to-image machine learning model to be presented, it can be argued that DALL-E is the first that had a big impact on the public: in the following months many examples of images generated with that tool would be posted on social media, and many more sophisticated models would surface, taking advantage of the increasing interest in the image generation and *artificial creativity* field.

Many of the most recent and successful of these systems stem from the "Diffusion Probabilistic Models" [16] family of machine learning approaches, while some of the earliest examples are Generative Adversarial Networks [10] with alternative techniques for the sample generation [12] compared to the classical GAN approaches.

II. Models

Image generation, and broadly speaking every kind of data generated through machine learning, is achieved with a family of models called "Generative Models". These models, in contrast with the family of "Discriminative Models", aim to generate new samples of data after learning its distribution, while the goal of discriminative models is to distinguish between different samples of data. Mathematically speaking, given a set of samples X and the corresponding labels Y, a generative model learns $p(X, Y)$ while a discriminative model learns $p(X \mid Y)$. Generative models can be explicit, when the directly learn $p(X)$, or implicit, if they learn how to sample from $p(X)$ without directly learning it.

Text-to-image generation can be achieved with many different kinds of generative models, but we will focus on three main family of models that recently achieved the best results: Generative Adversarial Networks, Autoregressive Transformers and Diffusion Models.

A. Generative Adversarial Networks

A Generative Adversarial Network (GAN) [10] is composed of two main parts:

- A Generator $\mathcal{G}$ that learns to generate plausible data with the goal of confusing the discriminator
- A Discriminator $\mathcal{D}$ that learns to distinguish between the real data and the fake data, penalizing the generator when it produces non-plausible samples

The trick that GANs employ to be able to sample data from complex, high-dimensional distributions, is to sample from a very simple distribution (usually a Gaussian, $\mathcal{N}(0, 1)$) and to train a differentiable distribution (usually a Neural Network, $\mathcal{G}$) to transform the sampled random noise into data from the target distribution:

- Sample $z \sim \mathcal{N}(0, 1)$
- Generate $x = \mathcal{G}(z)$
- Discriminate $\mathcal{D}(x) = \mathcal{D}(\mathcal{G}(z))$

Given real data x and random noise $z \sim \mathcal{N}(0, 1)$, the objective function of a GAN is C defined in (1)

$$C = \min_{\mathcal{G}} \max_{\mathcal{D}} \left(\mathbb{E}_x[\log \mathcal{D}(x)] - \mathbb{E}_z[\log(1 - \mathcal{D}(\mathcal{G}(z)))]\right) \quad (1)$$

This represents a hard two-player game, where the optimal solution resides in a saddle point that corresponds to the concurrent $\min \max$.

Most of the works in literature that employ GANs for data generation focus on developing different strategies that adapt the general architecture of these models to the interested data domain. Among the first examples of image generations that reached the general public is thispersondoesnotexist.com, a website released by nVidia which employs a particular GAN architecture, called StyleGAN, to generate highly realistic human portraits. Released in December 2018 with StyleGAN [12] and updated in December 2019 with StyleGAN2 [13], this project may represent the first public impact with state-of-the-art automatic image generation via machine learning models.

The original motivation behind StyleGAN was to unbox and better understand the image synthesis process and the properties of the latent spaces employed by the GANs, to both better control the generative process and to better compare different generators between each other. By taking inspiration from the style-transfer literature [11], which studies techniques of merging the content of an image with the artistic style of another, the StyleGAN proposed a redesigned generative process: the input latent code (random noise) z is mapped into an intermediate latent space $\mathcal{W}$ which is used to control

the generator $\mathcal{G}$ at each layer together with added Gaussian noise. This is in contrast to traditional generators, which feed the input directly to the generating network thus using the latent data just in the input layer. StyleGAN uses latent data to control the generative process at each layer, with $\mathcal{W}$ fed via adaptive instance normalization (AdaIN, (2)): it normalizes each feature map x_i separately given the style $y = (y_s, y_b)$ which is synthesized from $\mathcal{W}$ via learned transformations.

$$\text{AdaIN}(x_i, y) = y_{si}\frac{x_i - \mu(x_i)}{\sigma(x_i)} + y_{bi} \tag{2}$$

A main difference from classical style transfer [11] is that the style y of the target image is computed from $\mathcal{W}$, the latent data, instead of an example image.

The noise which is fed independently to each layer of the generative process is used to generate stochastic detail in the final images: these details range from the placement of the skin pores to other details like freckles and hair. By using random noise at each layer the network does not need to use activations from the previous layer, thus reusing data from the only source of input available, to generate stochastic effects: this reduces the occurrence of repetitive patterns, which can be particularly perceivable in the case of human faces. According to style transfer literature, the style of an image is reliably encoded by spatially invariant statistics like channel-wise mean or variance etc., and in StyleGAN the globally defined style affects global effects like lightning, pose and the like, while locally defined noise added independently to each pixel is only used to generate stochastic variation: if the generator tried to control a global effect, e.g. pose, with the noise, this would lead to spatially inconsistent decisions and then penalization by the discriminator. So the network learns to use global and local channels appropriately without explicit guidance, fully employing the power of the adversarial process.

Besides thispersondoesnotexists, StyleGANs have been used in other online demos like thisartworkdoesnotexists and thiscatdoesnotexists: an example of the latter is Fig. 1.

Fig. 1. A cat portrait generated by a StyleGAN.

B. Autoregressive Transformers

Transformers [18] are models that make extensive use of the Attention mechanism [17]. Attention is a technique that enhances small portions of the input data that are deemed important depending on the context. Given h_i as input, e.g. tokens of latent information, and context information S, in its simplest form an attention module computes C in three passes:

- Relevance, where each encoding h_i is fused with current context S yielding the relevance η_i between the two

$$\eta_i = \tanh(S, h_i) \tag{3}$$

- Normalization via a softmax operator

$$\alpha_i = \frac{e^{\eta_i}}{\sum_j e^{\eta_j}} \tag{4}$$

- Aggregation by soft attention voting

$$C = \sum_i \alpha_i h_i \tag{5}$$

Transformers are pure attention models with no recurrence that employs a variant of the attention mechanism called self-attention, used to draw global dependencies of the data. Self-attention maps each input to a set of three vectors, called Query (Q), Key (K) and Value (V): these vectors are used to compute the output vector, which is a weighted sum of the Vs where the weights are computed through a compatibility function of the Q and the corresponding K. The version used traditionally in the Transformers is called Scaled Dot-Product Attention, in Eq. 6 where d_k is the dimension of Q and K.

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{6}$$

A Transformer is made of an encoder and a decoder, both composed by stack of self-attention and fully connected layers. In particular, Autoregressive Transformers use the previous output as an additional input for subsequent passes through the network.

DALL-E [14] is an autoregressive transformer that models the prompt text and image as a single stream of tokens. In order to not use pixels directly as image tokens, and to prevent that much of the modeling capacity is spent capturing high-frequency details (like the stochastic variations of the displacement of hairs and freckles) rather than low-frequency details (like pose and lightning), the training is split into two stages [14]:

- A Discrete Variational Autoencoder (dVAE) is used to compress the images into a grid of tokes, each with 8192 possible values
- Up to 256 text tokens are concatenated with the 1024 image tokens to train an autoregressive transformer, that models the joint distribution over the text and image tokens.

The training is performed by maximizing the ELBO of the joint likelihood of the model distribution over images x, captions y and tokens z of the encoded image. By factorizing

the distribution with $p_{\theta\psi}(x, y, z) = p_\theta(x \mid y, z)p_\psi(y, z)$ we have the following lower bound:

$$\ln p_{\theta\psi}(x, y) \geq \mathbb{E}_{z\sim q_\phi(z \mid x)}[\ln p_\theta(x \mid y, z) - \beta KL(q_\phi(y, z \mid x) \,||\, p_\psi(y, z))] \tag{7}$$

Where q_ϕ is the distribution over the images tokes generated by the dVAE given the image, p_θ is the distribution over the images generated by the decoder given the tokens, p_ψ is the joint distribution over the text and image tokes modeled by the transformer.

The resulting model, called DALL-E and shown in Fig. 2, has been presented in January 2021 [1] and gained online relevance as soon as it came out thanks to a unofficial open-source version formerly called DALL-E mini [8] of which an example of the latest version, called Craiyon, is Fig. 3.

Much of the success that DALL-E faced since its presentation came from the many social media communities that used the tool to quickly generate and share memes, creating a viral phenomenon that contributed to an increased interest in text-to-image models research and in general in the artistic usage of machine learning models.

Fig. 2. "A low-angle view of a capybara sitting on a mountain" by DALL-E

Fig. 3. "A low-angle view of a capybara sitting on a mountain" by Crayion

C. Diffusion Probabilistic Models

One of the main limitation in machine learning, and in generative models in particular, is the tractability of the probability distributions involved. Learning, sampling and evaluation are often still analytically or computationally intractable when using highly flexible families of models, thus requiring a tradeoff between these two conflicting goals: flexibility and tractability. Diffusion Probabilistic Models (DM) [16] aim to achieve both, by taking inspiration from statistical physics.

DMs are built by two main processes:

- A Diffusion Process which converts any complex data distribution into a simple one (e.g. a Gaussian)
- A Reverse Process which defines the generative model

The diffusion process gradually converts an input data distribution $q(x^{(0)})$ into a tractable distribution $p(y)$ by repeated application of a Markov Diffusion Kernel (8) where β is the diffusion rate

$$T_p(y \mid y'; \beta) \tag{8}$$

$$p(y) = \int T_p(y \mid y'; \beta)p(y')\, dy' \tag{9}$$

$$q\left(x^{(t)} \mid x^{(t-1)}\right) = T_p\left(x^{(t)} \mid x^{(t-1)}; \beta\right) \tag{10}$$

The forward trajectory corresponds to performing T steps of diffusion starting from the data distribution and is defined in (11), which exhibits clear Markovian dynamics showing a prior and a first-order transition

$$q\left(x^{(0\ldots T)}\right) = q\left(x^{(0)}\right) \prod_{t=1}^{T} q\left(x^{(t)} \mid x^{(t-1)}\right) \tag{11}$$

The forward diffusion kernel $q\left(x^{(t)} \mid x^{(t-1)}\right)$ for example may correspond to a Gaussian Diffusion into a Gaussian distribution with identity covariance, defined in (12)

$$q\left(x^{(t)} \mid x^{(t-1)}\right) = \mathcal{N}\left(x^{(t-1)}\sqrt{1-\beta_t},\, \mathbf{I}\beta_t\right) \tag{12}$$

The generative distribution is trained to describe the same trajectory of the forward process but in reverse, thus yielding the reverse process described in (13)

$$p\left(x^{(0\ldots T)}\right) = p\left(x^{(T)}\right) \prod_{t=1}^{T} p\left(x^{(t-1)} \mid x^{(t)}\right) \tag{13}$$

During learning, and in the Gaussian diffusion kernel scenario taken as example, only the mean and covariance of the kernel need to be estimated:

- $f_\mu(x^{(t)}, t)$ defines the mean of the reverse Markov transitions
- $f_\sigma(x^{(t)}, t)$, defines the covariance of the reverse Markov transitions

Both can be defined via neural networks or other function fitting technique and the cost of these functions defines the computational cost of the overall algorithm. Those are used in the reverse diffusion kernel defined in (14)

$$p\left(x^{(t-1)} \mid x^{(t)}\right) = \mathcal{N}\left(f_\mu(x^{(t)}, t),\, f_\sigma(x^{(t)}, t)\right) \tag{14}$$

The main limitation of classical DMs is that they operate directly in pixel space and require the sequential run of a large number of steps of the Markov chain, thus resulting in expensive and long optimization (e.g. 150-1000 GPU days [9]) and inference processes (circa 5 days to produce 50k images [9]). This has multiple consequences, beginning with the carbon footprint, the unavailability of such computational resources to many researchers and the unapplicability of these models to real-word scenarios.

To increase the accessibility of these models, the Latent Diffusion Model (LDM) [15] family of approaches have been proposed. These models split training into two distinct phases:

- Train an Autoencoder [7] that provides a lower-dimensional representational space, more efficient than pixel space and perceptually equivalent
- Train the DM in the learned latent, smaller and more efficient, space

The resulting model, a LDM, exhibits better scaling properties and more efficient image generation than traditional DMs. A notable advantage is that the autoencoder may need to be trained just once, and then be used for multiple different LDMs or for completely different models.

A successful application of LDMs is the Stable Diffusion model [2] released in 2022, an open source implementation of an LDMs trained on the LAION-5B dataset [3]. This model has been used in several projects [4] and in a online demo [5]: an example of an image generated by this demo is Fig. 4. Another notable example is Midjourney [6], released in 2022 as well, a commercial application of a DM with the goal of being a tool for artists. An example of an image produced by Midjourney is Fig. 5.

LDMs recently have been implemented in mobile apps, highlighting their generation efficiency and general-purpose. This is one of the main differences between GANs (e.g. StyleGAN) and DMs (e.g. Stable Diffusion), with the first achieving great result only on data with limited variability with an overall small flexibility, while the latter showing more flexibility.

Fig. 4. ”A teacher explaining artificial intelligence with holograms” by Stable Diffusion 2.1

Fig. 5. ”Ukiyo-e, robot, detailed” by Midjourney

REFERENCES

[1] https://openai.com/blog/dall-e/. [Online; Accessed 2022-12-19].

[2] https://github.com/CompVis/stable-diffusion. [Online; Accessed 2022-12-18].

[3] https://laion.ai/blog/laion-5b/. [Online; Accessed 2022-12-18].

[4] https://www.banana.dev/blog/project-ideas-built-with-stable-diffusion. [Online; Accessed 2022-12-18].

[5] https://huggingface.co/spaces/stabilityai/stable-diffusion. [Online; Accessed 2022-12-18].

[6] https://www.midjourney.com/home/. [Online; Accessed: 2022-12-16].

[7] Dor Bank, Noam Koenigstein, and Raja Giryes. Autoencoders. *CoRR*, abs/2003.05991, 2020.

[8] Boris Dayma. https://github.com/borisdayma/dalle-mini/. [Online; Accessed 2022-12-19].

[9] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. *CoRR*, abs/2105.05233, 2021.

[10] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. 2014.

[11] Xun Huang and Serge J. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. *CoRR*, abs/1703.06868, 2017.

[12] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *CoRR*, abs/1812.04948, 2018.

[13] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. *CoRR*, abs/1912.04958, 2019.

[14] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation, 2021.

[15] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and BjÃ¶rn Ommer. High-resolution image synthesis with latent diffusion models, 2022.

[16] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. *CoRR*, abs/1503.03585, 2015.

[17] Derya Soydaner. Attention mechanism in neural networks: where it comes and where it goes. *Neural Computing and Applications*, 34(16):13371–13385, may 2022.

[18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.