Text to Image Synthesis Using Stacked Generative Adversarial Networks

Ali Zaidi
Stanford University & Microsoft AIR
alizaidi@microsoft.com

Abstract

Human beings are quickly able to conjure and imagine images related to natural language descriptions. For example, when you read a story about a sunny field full of flowers, an image of a beautiful field with blossoming flowers might pop into your head. Artificial synthesis of images using text descriptions or human cues could have profound applications in visual editing, animation, and digital design. The goal of this project was to explore succesful architectures for image synthesis from text. In particular, we examined StackGANs ([8]) which attempt to improve the synthesis process by using a two-stage procedure, each of which is it's own manageable GAN implementation. We examined StackGAN results on two large datasets, the Caltech-UCSD Birds-200-2011 and the flowers 102 dataset, and were able to produce highly realistic synthesized images. Our experiments, learnings, and future ideas are described in this paper.

1. Introduction

The advent or realistic image-generation using text descriptions could have a profound impact on a number of fields, ranging from interactive computational graphic design, image fine-tuning, and perhaps even animation. However, generating trealy plausible looking images has not been easy. The majority of advanced methods do not produce photo-realistic details that are faithful text descriptions. The main challenge of this problem is the susceptibility of generative models to mode collapse [3], due to the fact that the space of plausible images given text descriptions is multimodal, in that there are a large number of images that could correctly fit the given text description.

Recent progress in generative models, especially Generative Adversarial Nets (GANs) [3, 2] has made have made significant improvement in synthesizing images and generating plausible samples.

2. Related Work

In [5], Reed et. al provided a two-stage approach for generating images from text. In the first stage, the authors learned a text feature representation that captures the most important visual details of the image. The following stage utilized those feature representations to synthesize the image. The primary novelty of the author's approach was to condition not on a single class label, but rather use a end-to-end differentiable architecture conditioned on a complete text description. The authors used a deep convolutional generative model (DC-GAN) conditioned on text features encoded by a hybrid-character-level convolutional recurrent neural network. The results presented by Reed et. al generated plausible 64 x 64 images, but were not likely to fool a human. Moreover, they did not scale as well to larger datasets like MS COCO images.

Rather than directly generating from text-features as in the previously discussed paper, the authors in [8] decided to break up the generative process into two sub-problems. In the first stage, teh authors used a GAN to learn the basic contours, shape and colors of an image conditional on a text description and generates background regions from a random noise vector sampled from a prior distribution. These initial generated images are of low resolution and substantially coarser than any realistic images would be. This first stage is then followed with a second stage that acts like a super-resolution, i.e., it focuses more directly on improving the image quality and remedying defects in the original low-resolution images.

3. Methods

3.1. Generatiave Adversarial Networks

Generative adversarial networks (GANs) consist of a generator G and a discriminator D that compete in a two-player minimax game: The discriminator tries to distinguish between real and synthetic images, and the gnerator tries to fool the discriminator. Concretely, D and G play the following game on V (D, G):

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

For each stage, we utilize the GAN training procedure that is similar to a two-player min-max game with the following objective function:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim \mathrm{p}_{data}}[\log D(x)] + \mathbb{E}_{z\sim \mathrm{p}_z}[\log(1-D(G(z)))],$$

where x is a real image from the true distribution, and z is a noise vector sampled from p_z, which might be a Gaussian or uniform distribution.

Moreover, in our architecture we will follow the conditional-GAN approach and additionally condition both the generator and the discriminator on additional variables, which will be the text embeddings of our descriptions, denoted by c, therefore giving us generator and discriminator $G(z,c)$ and $D(z,c)$.

Our model architecture is shown in the figure below. In the figure $\mathcal{G}_I$ denote sthe generator from stage-I, which produces low-resolution images, and $\mathcal{G}_{II}$ is the generator from stage-II, which produces higher quality images by conditioning on the text c and $\mathcal{G}_{II}$. We describe each stage more thoroughly below.

3.2. Stage I-GAN: Sketch

The first stage of our architecture involves training a GAN to generate low resolution images. In this stage, we condition on a text description encoded as a text-embedding φ_t. This text-embedding is learned using the deep structured textt embedding approach describe below.

3.2.1 Deep Structured Text Embeddings

The text-embeddings we conditioned on were first pretrained using a structured joint embedding approach. More precisely, we trained functions f_v and f_t that map image features $v \in \mathcal{V}$ and text descriptions $t \in \mathcal{T}$ to class labels $y \in \mathcal{Y}$, i.e., that minimize the empirical risk given by

$$\frac{1}{N}\sum_{n=1}^{N}\Delta(y_n, f_v(v_n)) + \Delta(y_n, f_t(t_n)),$$

where $\Delta : \mathcal{Y}\times\mathcal{Y} \to \mathbb{R}$ is the zero-one loss derived from looking at one-hot encodings of our class labels. To make things differentiable, I used a convex surrogate rather than the discontinuous 0-1 loss. Classifiers f_v and f_t are parameterized as

$$f_v(v) = \arg\max_{y\in\mathcal{Y}} \mathbb{E}_{t\sim\mathcal{T}(y)}\left[\phi(v)^\top \varphi(t)\right]$$
$$f_t(t) = \arg\max_{y\in\mathcal{Y}} \mathbb{E}_{v\sim\mathcal{V}(y)}\left[\phi(v)^\top \varphi(t)\right],$$

where ϕ is the image encoder obtained through convolutional neural network, and φ is our text encoder obtained through an LSTM.

This formluation follow the approach outlined in [5], to train a deep convolutional generative adversarial network (DC-GAN) conditioned on text features encoded by a hybrid character-level convolutional recurrent neural network. Both the generator network G and the discriminator network D perform feed-forward inference conditioned on the text feature.

3.3. Stage II-GAN: Superesolution

The second stage of our StackGAN acts like a superresolution/up-sampling tool. Given a low resolution sample s_0, and conditional on the text embedding φ specified through the same procedure in stage I, Stage-II GAN trains a discriminator D and generator G by alternatively maximizing $\mathcal{L}_D$ and minimizing $\mathcal{L}_G$ in the following:

$$\mathcal{L}_D = \mathbb{E}_{(I,t)\sim\mathrm{p_{data}}}[\log D(I,\varphi_t)] + \mathbb{E}_{s_0\sim \mathrm{p}_{G_0},\, t\sim\mathrm{p_{data}}}[\log(1-D(G(s_0,c),\varphi_t))],$$

and

$$\mathcal{L}_G = \mathbb{E}_{s_0\sim \mathrm{p}_{G_0},\mathrm{t}\sim\mathrm{p_{data}}}[\log(1-D(G(s_0,c),\varphi_t))] + \lambda D_{\mathrm{KL}}\left(\boldsymbol{\mu}_{\varphi_t,\Sigma_{\varphi_t}}\right),$$

where $s_0 = G_0(z, c_0)$ is the generated sample from stage-I, and λ is a regularization hyperparameter and $\boldsymbol{\mu}_{\varphi_t,\Sigma_{\varphi_t}}$ is a Gaussian sampling distribution for our text desription.

4. Dataset and Features

To examine the Stack-GAN architecture, we ran experiments on the Caltech-UCSD Bird (CUB) dataset [7] and Oxford-12 flowers dataset [4].

The CUB dataset conists of 200 different bird species and a toal of 11,788 images. Following the pre-processing step in [8], we cropped the images of all the birds so that they covered at least 75% of the total image size. The Oxford-102 dataset consists of 102 categories of flower species and a total of 8,189 images. In this case, the flowers make up a majority of the image area, and we therefore did not crop the images in any position.

Each image in the CUB and Oxford-102 dataset was coupled with a collection of 10 captions as provided by [1][1]. For evaluation, we split both datasets into disjoint class train and test splits and used the inception score as a quantitiave metric:

$$I = \exp\left(\mathbb{E}_x D_{KL}\left(p(y|\boldsymbol{x})\,|p(y)\right)\right), \tag{1}$$

[1]The captions data were taken from the following github repository: `https://github.com/reedscot/cvpr2016`

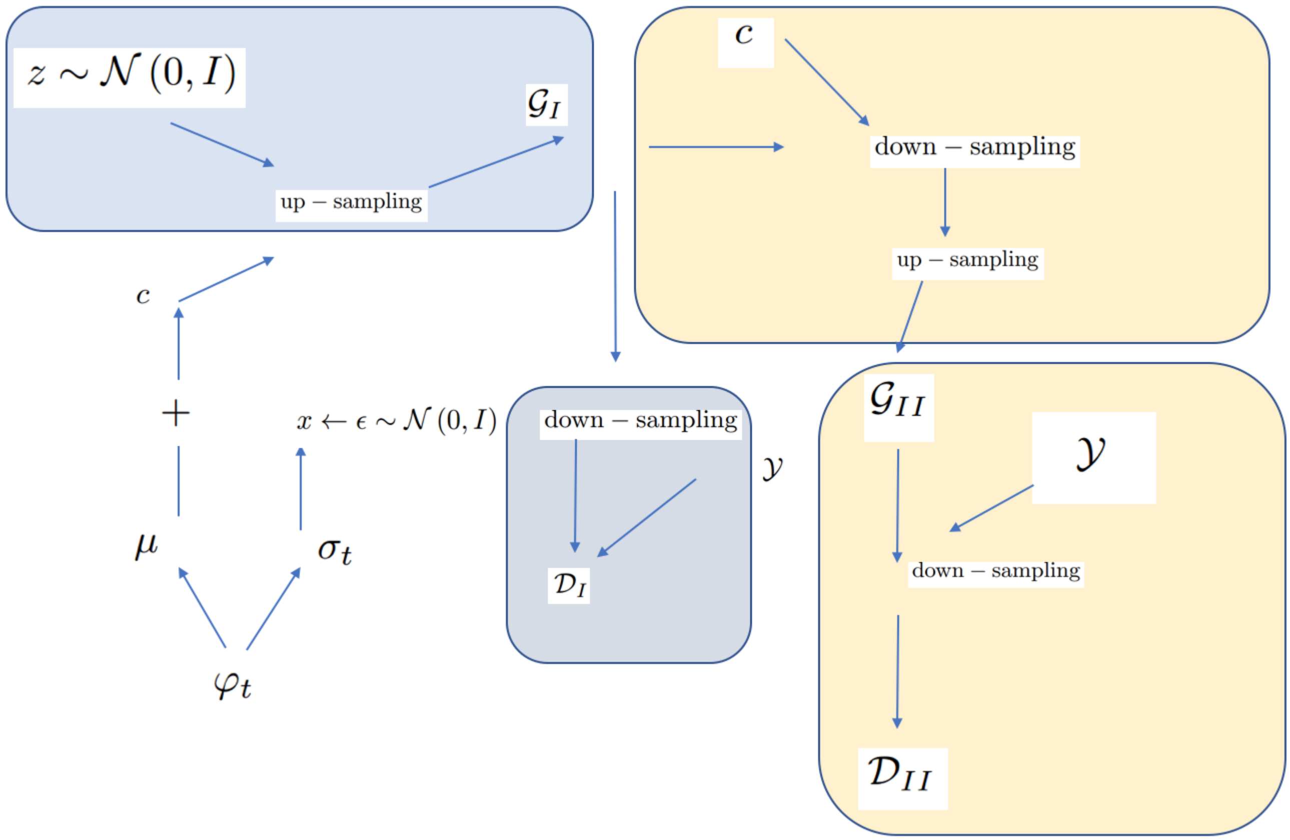

Figure 1. Design of StackGAN

where y is the label preidcted by the Inception model [6]. The larger the metric, the more diverse and rich our image constructions are.

4.1. Implementation and Training

The architecture of our model is pictured in Figure 1. The first stage starts with nearest-neighbor up-sampling followed by 3×3 stride 1 convolution operation. Batch normalization and a leakly ReLU activation were used after each convolution. Similarly, the residual blocks consist of 3×3 stride 1 convolutions, batch normalization, and leakly ReLU activations. Following Stage-I, stage II proceeds initially with 128×128 residual blocks describe dthrough two residual blocks, followed by down-sampling using 4×4 stride 2 convolutions, batch normalization, and Leaky ReLUs.

Training was conducted using a port of the tensorflow implementation provided by Han Zhang[2]. The code was modified to work with tensorflow 1.1 from 0.11, and utilized leaky ReLU activations throughout the architecture.

5. Experiments and Results

Qualitative results from our model can be seen in the images in the following pages. In particular, generative samples from the Birds dataset can be seen in Figure 2, and samples from the flowers dataset can be seen in Figure 3. The first rows show the generated samples from stage-I following $\mathcal{G}_I$. These figures show that the GAN has generated meaningful shapes, colours and depictions of the objects. However, it lacks significant details to pass off as a realistic sample, in some cases, an entire beak is missing, for example. However, in almost all of the generated cases, the second stage generated a highly plausible sample. The major details of the object under consideration are now described at high-resolution.

Quantitative results using the inception metric (1) against a sample of size 10K are shown in the table 1.

[2] `https://github.com/hanzhanggit/StackGAN`

Figure 2. Birds Generation

	CUB	Oxford-102
Stack-GAN	3.50 ± 0.12	3.20 ± 0.04

Table 1. Inception Scores

6. Future Ideas

In this work, we examined the training and evaluation of a Stack-GAN for highly-realistic synthesis of images from text phrases. In future work I'd like to try and scale to larger image-caption datasets like MSCOCO. I'd also like to try a sequential dual-training method, where we train do text-to-image synthesis in tandem with image-to-text synthesis. For multi-category datasets like MSCOCO these might perform better.

References

[1] Learning deep representations of fine-grained visual descriptions, booktitle = IEEE Computer Vision and Pattern Recognition, year = 2016, author = Scott Reed and Zeynep Akata and Bernt Schiele and Honglak Lee,.

[2] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Ad-*

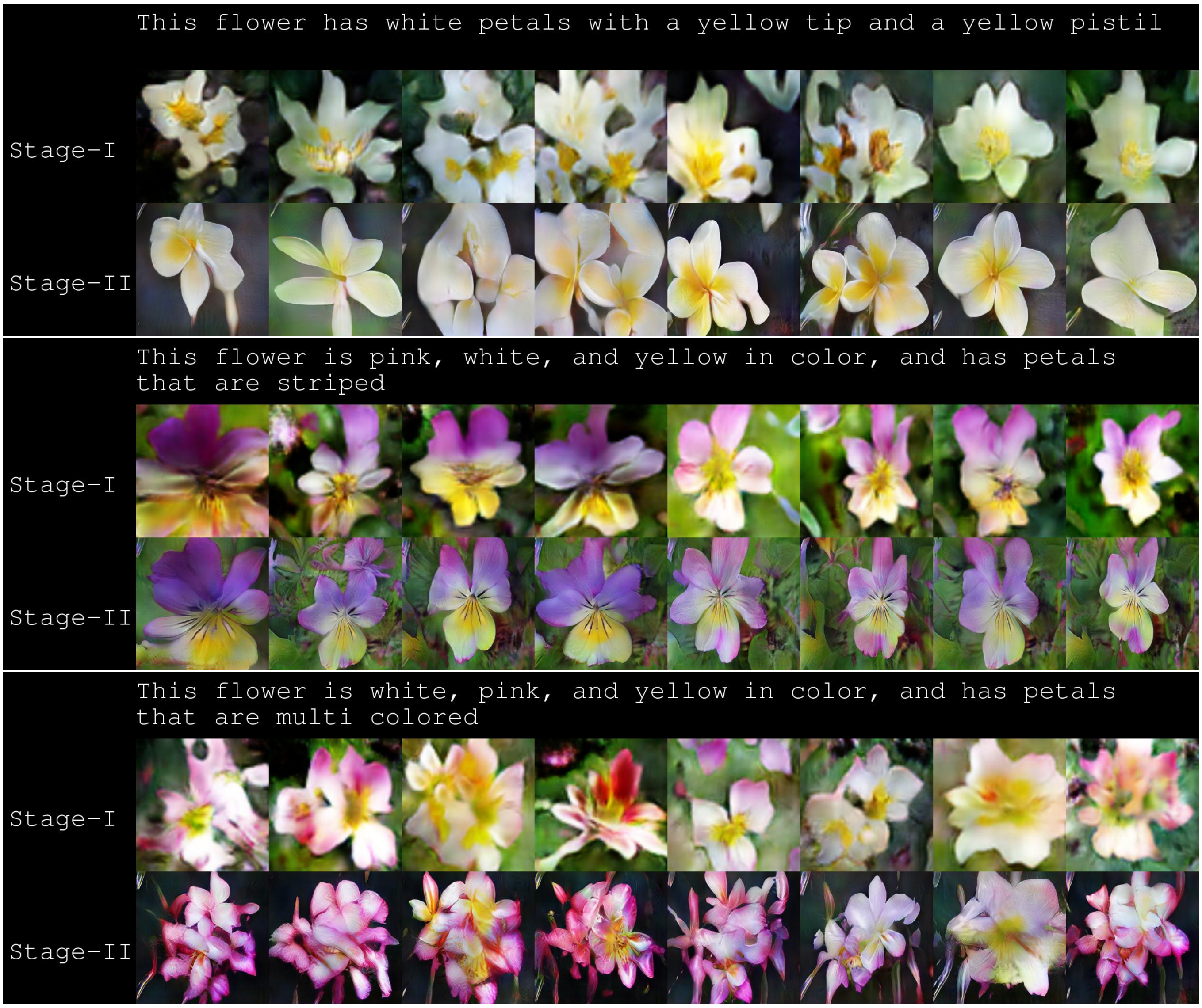

Figure 3. Flowers Generation

vances in Neural Information Processing Systems 27, pages 2672–2680. Curran Associates, Inc., 2014.

[3] Ian J. Goodfellow. NIPS 2016 tutorial: Generative adversarial networks. *CoRR*, abs/1701.00160, 2017.

[4] M-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing*, Dec 2008.

[5] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text-to-image synthesis. In *Proceedings of The 33rd International Conference on Machine Learning*, 2016.

[6] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567, 2015.

[7] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical report, 2011.

[8] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaolei Huang, Xiaogang Wang, and Dimitris Metaxas. Stackgan: Text to photo-realistic image syn-

thesis with stacked generative adversarial networks. *arXiv:1612.03242*, 2016.