Generative Adversarial Text to Image Synthesis

Hongzhi Liu

June 30, 2018

Abstract

Automatic synthesis of realistic images from text would be interesting and useful but current AI systems are still far from this goal. However, generic and powerful recurrent neural network architectures have been developed to learn discriminative text feature representations in recent years. Meanwhile, deep convolutional generative adversarial networks (GANs) have begun to generate highly compelling images of specific categories. Today, I read a thesis written by Scott Reed who is from University of Michigan. His team introduce a novel deep architecture and GAN formulation to effectively bridge these advances in text and image modeling, translating visual concepts from characters to pixels. They demonstrate the capability of our model to generate plausible images of birds and flowers from detailed text descriptions.

1. Overview of Model for Generating Images

Many researchers have recently exploited the capability of deep convolutional decoder networks to generate realistic images. Dosovitskiy *et al.* [2] trained a deconvolutional network to generate 3D chair renderings conditioned on a set of graphics codes indicating shape, position and lighting. Yang *et al.* [9] added an encoder network as well as actions to this approach. They trained a recurrent convolutional encoderdecoder that rotated 3D chair models and human faces conditioned on action sequences of rotations. Reed *et al.* [8] encode transformations from analogy pairs and use a convolutional decoder to predict visual analogies on shapes, video game characters and 3D cars.

Generative adversarial networks have also benefited from convolutional decoder networks for the generator network module [3]. Denton *et al.* [1] used a Laplacian pyramid of adversarial generator and discriminators to synthesize images at multiple resolutions. This work generated compelling high-resolution images and could also condition on class labels for controllable generation. Radford et al. [6] used a standard convolutional decoder but developed a highly effective and stable architecture incorporating batch normalization to achieve striking image synthesis results.

In this paper, Scott Reed and his team develop a simple and effective GAN architecture and training strategy that enables compelling text to image synthesis of bird and flower images from human-written descriptions [7]. They mainly use the Caltech-UCSD Birds dataset and the Oxford-102 Flowers dataset along with five text descriptions per image they collected as their evaluation setting. The model is trained on a subset of training categories and the team demonstrate its performance both on the training set categories and on the testing set.

2. Method of Image Synthesis

Scott Reed's approach is to train a deep convolutional generative adversarial network (DC-GAN) conditioned on text features encoded by a hybrid character-level convolutionalrecurrent neural network. Both the generator network G and the discriminator network D perform feed-forward inference conditioned on the text feature.

2.1. Network Architecture

Reed's team use the following notation. The generator network is denoted $G : \mathbb{R}^Z \times \mathbb{R}^T \rightarrow \mathbb{R}^D$, the discriminator as $D : \mathbb{R}^D \times \mathbb{R}^T \rightarrow \{0, 1\}$, where T is the dimension of the text description embedding, D is the dimension of the image, and Z is the dimension of the noise input to G. We illustrate our network architecture in Fig. 1.

In the generator G, first Reed's team sample from the noise prior $z \in \mathbb{R}^z \sim \mathcal{N}(0, 1)$ and they encode the text query t using text encoder φ. The description embedding $\varphi(t)$ is first compressed using a fully-connected layer to a small dimension followed by leaky-ReLU and then concatenated to the noise vector z. Following this, inference proceeds as in a normal deconvolutional network: they feedforward it through the generator G; a synthetic image $\hat{x}$ is generated via $\hat{x} \leftarrow G(z, \varphi(t))$. Image generation corresponds to feed-forward inference in the generator G conditioned on query text and a noise sample.

In the discriminator D, Reed performs several layers of stride-2 convolution with spatial batch normalization [4] followed by leaky ReLU. The team again reduce the di-

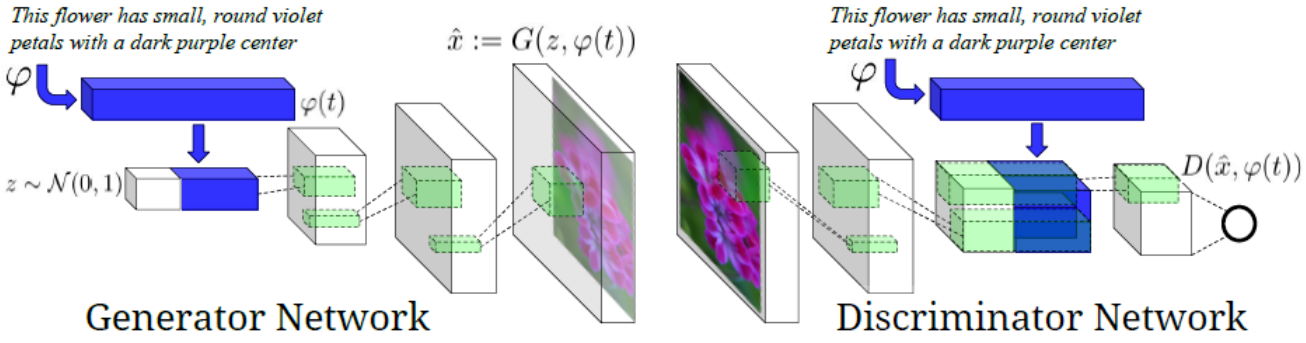

Figure 1. The text-conditional convolutional GAN architecture. Text encoding $\varphi(t)$ is used by both generator and discriminator. It is projected to a lower-dimensions and depth concatenated with image feature maps for further stages of convolutional processing.

mensionality of the description embedding $\varphi(t)$ in a fully-connected layer followed by rectification. When the spatial dimension of the discriminator is 4×4, they replicate the description embedding spatially and perform a depth concatenation. They then perform a 1×1 convolution followed by rectification and a 4×4 convolution to compute the final score from D. Batch normalization is performed on all convolutional layers.

2.2. Learning with Manifold Interpolation

Deep networks have been shown to learn representations in which interpolations between embedding pairs tend to be near the data manifold. Motivated by this property, they can generate a large amount of additional text embeddings by simply interpolating between embeddings of training set captions. Critically, these interpolated text embeddings need not correspond to any actual human-written text, so there is no additional labeling cost. This can be viewed as adding an additional term to the generator objective to minimize:

$$\mathbb{E}_{t_1,t_2 \sim p_{data}}[\log(1 - D(G(z, \beta t_1 + (1-\beta)t_2)))]. \quad (1)$$

where z is drawn from the noise distribution and β interpolates between text embeddings t_1 and t_2. In practice they found that fixing $\beta = 0.5$ works well.

Because the interpolated embeddings are synthetic, the discriminator D does not have "real" corresponding image and text pairs to train on. However, D learns to predict whether image and text pairs match or not. Thus, if D does a good job at this, then by satisfying D on interpolated text embeddings G can learn to fill in gaps on the data manifold in between training points. Note that t_1 and t_2 may come from different images and even different categories.

2.3. Inverting the Generator for Style Transfer

If the text encoding $\varphi(t)$ captures the image content, then in order to generate a realistic image the noise sample z should capture style factors such as background color and pose. With a trained GAN, one may wish to transfer the

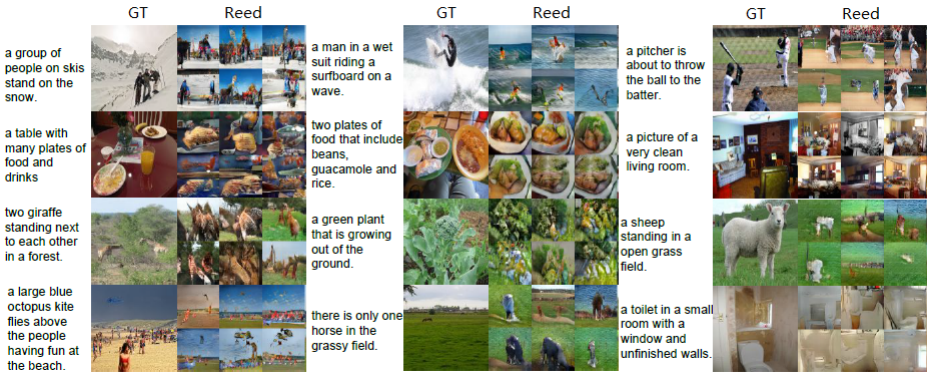

Figure 2. Generating images of general concepts using our GAN-CLS on the MS-COCO validation set. Unlike the case of CUB and Oxford-102, the network must handle multiple objects and diverse backgrounds.

style of a query image onto the content of a particular text description. To achieve this, one can train a convolutional network to invert G to regress from samples $\hat{x} \leftarrow G(s, \varphi(t))$ back onto z. They used a simple squared loss to train the style encoder as Eq. 2:

$$\mathcal{L}_{style} = \mathbb{E}_{t,z\sim\mathcal{N}(0,1)} \parallel z - S(G(z, \varphi(t))) \parallel_2^2 . \quad (2)$$

where S is the style encoder network. With a trained generator and style encoder, style transfer from a query image x onto text t proceeds as follows:

$$s \leftarrow S(x), \quad \hat{x} \leftarrow G(s, \varphi(t)) \quad (3)$$

where $\hat{x}$ is the result image and s is the predicted style.

3. Evaluation of the Method

Reed and his team trained a GAN-CLS on MS-COCO to show the generalization capability of their approach on a general set of images that contain multiple objects and variable backgrounds. They use the same text encoder architecture, same GAN architecture and same hyperparameters as in CUB and Oxford-102. The only difference in training the text encoder is that COCO does not have a single object category per class. However, They can still learn an instance level image and text matching function as in [5].

Samples and ground truth captions and their corresponding images are shown on Fig. 2. A common property of all the results is the sharpness of the samples, similar to other GAN-based image synthesis models. They also observe diversity in the samples by simply drawing multiple noise vectors and using the same fixed text encoding.

References

[1] E. L. Denton, S. Chintala, and R. Fergus. Deep generative image models using a laplacian pyramid of adversarial networks. In *NIPS*, 2015. 1

[2] A. Dosovitskiy, J. T. Springenberg, and T. Brox. Learning to generate chairs with convolutional neural networks. In *CVPR*, 2015. 1

[3] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, 2014. 1

[4] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 1

[5] R. Kiros, R. Salakhutdinov, and R. S. Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*, 2014. 3

[6] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 1

[7] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*, 2016. 1

[8] S. E. Reed, Y. Zhang, Y. Zhang, and H. Lee. Deep visual analogy-making. In *NIPS*, 2015. 1

[9] J. Yang, S. Reed, M. H. Yang, and H. Lee. Weakly-supervised disentangling with recurrent transformations for 3D view synthesis. In *NIPS*, 2015. 1