Adjustable Text to Image Synthesis

Ankit Vani
ankit.vani@nyu.edu

Srivas Venkatesh
srivas.venkatesh@nyu.edu

Abstract

Generative adversarial networks have been shown to generate very realistic images by learning through a min-max game. Furthermore, these models are known to model image spaces more easily when conditioned on class labels. In this work, we consider conditioning on fine-grained textual descriptions, thus also enabling us to produce realistic images that correspond to the input text description. Additionally, we consider the task of learning disentangled representations for images through special latent codes, such that we can move them as knobs to alter the generated image. These latent codes take on very interpretable roles and are learnt in a completely unsupervised manner, using ideas from InfoGAN. We show that the learnt latent codes that encode much more variance and semantic interpretability as compared to standard GANs by experimenting on two datasets.

1 Introduction

Building models of our complex real world is a fascinating and a very difficult problem. In this project, we tackle the problem of modeling concepts from textual data, and mapping them to the image space. We use generative adversarial networks (GANs) (Goodfellow et al. [2014]) to generate realistic images from raw textual descriptions. However, to go one step further, we also wish to learn disentangled representations of images that can be used to adjust the generated images in an interpretable way. Ideally, we would want a small number of controls to do so, rather than hundreds of knobs with no idea how those variables interact with each other.

To achieve this, we consider the information theoretic extension of GAN, known as InfoGAN (Chen et al. [2016]), to maximize the mutual information between certain latent codes and the image generated from them. We extend the previous text to image work (Reed et al. [2016b]) using InfoGAN, and extend InfoGAN by conditioning on textual representations. We find that these latent codes can learn to pick up global semantics from the data distribution, so that once they are learnt, they can be labeled by the interpretable effect they have on the image to enable manual fine-tuning after generating an image from its caption.

2 Background

2.1 Generative Adversarial Network (GAN)

Generative Adversarial Networks (GANs) (Goodfellow et al. [2014]) have been tremendously successful in learning to generate realistic images by playing a min-max game. A discriminator D is trained to classify input images as coming from the data distribution ('real') or coming from the generator ('fake'). A generator G is trained to fool the discriminator by producing images that are indistinguishable from real images for the discriminator. The generator takes as input a noise vector z, coming from a prior distribution, and maps it to a point in the pixel space, which represents an image. The prior noise distribution usually has zero mean and is a Gaussian or a uniform distribution.

The GAN objective can thus be formulated as

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

The signal for the generator to update its generative model comes from the discriminator itself, by using gradients of the discriminator loss with respect to the generated image. In practice, gradients from the discriminator are reversed and backpropagated to the generator, thereby asking the generator to maximize the discriminator's loss.

2.2 Deep Convolutional GAN (DCGAN)

To get better quality generations from the generators of GANs, Radford et al. [2015] presented an architecture called deep convolutional GAN (DCGAN) where the generator and the discriminator are convolutional networks that mirror each other in structure.

They also replace the pooling layers with strided convolutions in the discriminator and fractional-strided convolutions in the generator. Additionally, they use the LeakyReLU activation in the discriminator for all layers instead of ReLU. These two additions ensure that there are no sparse gradients in the discriminator, which helps with the stability of the min-max objective training. To alleviate internal covariate shift and help make training more stable, they use batch normalization (Ioffe and Szegedy [2015]) in both the generator and the discriminator. There are no fully connected hidden layers, enabling deeper models.

The text to image architecture we use is based on the DCGAN architecture.

2.3 Information Maximizing GAN (InfoGAN)

Information maximizing GAN (InfoGAN) (Chen et al. [2016]) is an extension to GAN that learns disentangled representations in an unsupervised manner. Unlike the standard GAN, where an image is generated from a noise vector z, an image in this case is generated from a noise vector z and latent codes c. Like z, we define a prior distribution over c with zero mean, which is usually taken to be a Gaussian or a uniform distribution. c can also contain discrete latent codes, which can have a discrete uniform prior.

To learn disentangled representations, the model maximizes the mutual information between c and the generated image $G(z, c)$, where the mutual information is given by

$$\mathcal{I}(c; G(z,c)) = \mathcal{H}(G(z,c)) - \mathcal{H}(G(z,c) \mid c) = \mathcal{H}(c) - \mathcal{H}(c \mid G(z,c)) \tag{2}$$

The authors derive a variational lower bound to Equation (2):

$$\mathcal{I}(c; G(z,c)) \geq \mathop{\mathbb{E}}_{x \sim G(z,c)} \left[\mathop{\mathbb{E}}_{c' \sim p(c|x)} [\log Q(c' \mid x)] \right] + \mathcal{H}(c) \tag{3}$$

where Q is a variational approximation for $p(c \mid x)$. They show that under suitable regularity conditions, Equation (3) can be simplified to

$$\mathcal{I}(c; G(z,c)) \geq \mathop{\mathbb{E}}_{c \sim p(c), x \sim G(z,c)} [\log Q(c \mid x)] + \mathcal{H}(c) = L_I(G,Q) \tag{4}$$

Thus, by maximizing $L_I(G, Q)$, we would maximize the mutual information between c and $G(z, c)$. L_I can be added to the GAN's objective and maximized jointly with respect to G and Q as

$$\min_{G,Q} \max_D V_{\text{InfoGAN}}(D,G,Q) = V(D,G) - \lambda L_I(G,Q) \tag{5}$$

where $V(D, G)$ is defined in Equation (1). λ is a hyperparameter, that is tuned to weigh the mutual information criterion in the learning objective. The authors suggest setting this value such that λL_I is on the same scale as the GAN objective V.

In practice, Q shares all of its layers and parameters with D upto D's penultimate layer, following with an additional layer that outputs the sufficient statistics of the distribution $Q(c \mid x)$. For a Gaussian, these are the mean and diagonal covariance values. For discrete latent codes, these are the softmax logits. Q converges very quickly during training, and thus it essentially comes for free with GAN, without the need for additional training time or model capacity beyond a single layer.

2.4 Deep symmetric structured joint embedding

Given an image $v \in \mathcal{V}$ and its corresponding fine-grained textual description $t \in \mathcal{T}$, Reed et al. [2016a] suggest learning the encodings $\theta(v)$ for images and $\varphi(t)$ for text, such that a joint embedding $F(v,t) = \theta(v)^\top \varphi(t)$ can be defined. They define the classifiers

$$f_v(v) = \arg\max_{y \in \mathcal{Y}} \mathop{\mathbb{E}}_{t \in \mathcal{T}(y)} \left[F(v,t) \right], \tag{6}$$

$$f_t(t) = \arg\max_{y \in \mathcal{Y}} \mathop{\mathbb{E}}_{v \in \mathcal{V}(y)} \left[F(v,t) \right] \tag{7}$$

to classify data points (v,t) into labels $y \in \mathcal{Y}$. By jointly training the text and image encoders to minimize the classification loss, they are able to learn visually descriptive embeddings $\varphi(t)$ for text and embeddings $\theta(v)$ for images, such that they have a higher compatibility score F when their class is the same.

They explore various text encoder models, such as a convolutional network, a convolutional recurrent network and a recurrent network. For our work on text to image synthesis, we use pretrained text encoders provided by the authors, that use a character-level convolutional recurrent network.

3 Text to Image InfoGAN

3.1 Objective

Text to Image GAN (Reed et al. [2016b]) is a conditional GAN, where the discriminator D and the generator G are conditioned on a text encoding $\varphi(t)$ for an image caption t. Thus, given a text caption, the generator is tasked with producing a realistic image described by the input caption. When training the generator, the authors also propose creating auxiliary text embeddings by interpolating between embeddings of text descriptions in the data, essentially providing infinite samples from the text embedding space to condition on.

Based on Equation (1), we can formulate the text to image GAN objective as

$$\begin{aligned} \min_G \max_D \mathcal{V}(D,G,\varphi) = & \mathop{\mathbb{E}}_{(x,t) \sim p_{\text{data}}(x,t)} \left[\log D(x, \varphi(t)) \right] + \\ & \mathop{\mathbb{E}}_{z \sim p_z(z), t \sim p_{\text{intdata}}(t)} \left[\log(1 - D(G(z, \varphi(t)), \varphi(t))) \right] \end{aligned} \tag{8}$$

where $t \sim p_{\text{intdata}}(t)$ is short for sampling through the process $t_1 \sim p_{\text{data}}(t), t_2 \sim p_{\text{data}}(t), \beta \sim [0,1], t = \beta t_1 + (1-\beta) t_2$.

We extend the text to image GAN by adding the ability to manipulate the output images, adjusting it manually to possibly better suit the textual description. We learn disentangled latent codes c, enabling each latent code to take on interpretable and semantically meaningful roles. We do this by extending InfoGAN to the text to image synthesis setting, where we want to maximize the mutual information between the latent codes c and the conditionally generated image $G(z, c, \varphi(t))$:

$$\mathcal{I}(c; G(z,c,\varphi(t))) \geq \mathop{\mathbb{E}}_{t \sim p_{\text{intdata}}(t), x \sim G(z,c,\varphi(t))} \left[\mathop{\mathbb{E}}_{c' \sim p(c|x,t)} \left[\log Q(c' \mid x, t) \right] \right] + \mathcal{H}(c) \tag{9}$$

As described for InfoGAN, under suitable regularity conditions, we can obtain a variational lower bound for $\mathcal{I}(c; G(z,c,\varphi(t)))$:

$$\mathcal{I}(c; G(z,c,\varphi(t))) \geq \mathop{\mathbb{E}}_{c \sim p(c), t \sim p_{\text{intdata}}(t), x \sim G(z,c,\varphi(t))} \left[\log Q(c \mid x, t) \right] + \mathcal{H}(c) = \mathcal{L}_I(G, Q, \varphi) \tag{10}$$

We experiment with two possible models, one where the variational approximation Q takes into consideration the textual description when looking at the generated image, and one where it doesn't. In the case where Q does not consider the textual description, we assume in Equation (10) that the output of Q is conditionally independent of t given x, in which case $Q(c \mid x, t)$ simplifies to $Q(c \mid x)$. This is equivalent to removing an edge between t and c sampled from Q in the graphical model.

Our text to image InfoGAN objective is given by

$$\min_{G,Q} \max_D \mathcal{V}_{\text{t2i2GAN}}(D,G,Q,\varphi) = \mathcal{V}(D,G,\varphi) - \lambda \mathcal{L}_I(G,Q,\varphi) \tag{11}$$

where $\mathcal{V}(D,G,\varphi)$ is defined in Equation (8), and $\mathcal{L}_I(G,Q,\varphi)$ in Equation (10).

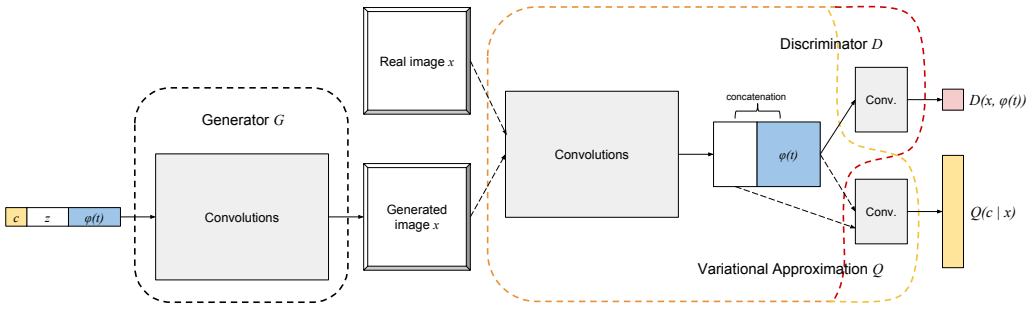


Figure 1: High-level architecture for Text to Image InfoGAN model. The grey convolution blocks abstract the DCGAN architecture. Dashed arrows represent alternate flows.

3.2 Model architecture

Our model architecture is very similar to Reed et al. [2016b], which itself is based on the DCGAN architecture. The generator and the discriminator are both conditioned on the text embedding $\varphi(t)$, which comes from the data distribution in case the discriminator is looking at a real image, or from interpolations of true text embeddings in case of generations. Our model architecture is illustrated in Figure 1.

The generator takes as input a vector formed by the concatenation of latent codes c, noise z, and the text embedding. We choose the prior for z to be a Gaussian with zero mean and identity covariance. We sample c from a multivariate uniform distribution between -1 and 1, but consider the prior $p(c)$ to be a Gaussian with zero mean and identity covariance for the mutual information lower bound defined in Equation (10). The text embedding $\varphi(t)$ is passed through a fully-connected network and reduced to a size of 128 dimensions, followed by a LeakyReLU, before concatenation to the input vector.

Another transformation on the text embedding $\varphi(t)$ followed by rectification is done for the discriminator. When the spatial dimension of the DCGAN discriminator is 4×4, we replicate this reduced embedding spatially and concatenate it to the 4×4 discriminator output. A 1×1 convolution followed by rectification and another 4×4 convolution is performed obtain the final discriminator value.

If we want to consider $Q(c \mid x, t)$ (that is, consider the textual description in Q in addition to the generated image), then the depth concatenated 4×4 output is passed through a different 1×1 convolution followed by rectification. The output is then flattened and a fully connected layer maps it to a vector representing the mean and diagonal covariance for the latent codes in $Q(c \mid x, t)$. If we instead want to only consider $Q(c \mid x)$, then we use the 4×4 discriminator output before depth concatenation for the convolutional layer of Q.

4 Experimental Results

4.1 Datasets

For our experiments, we used 2 datasets:

- The Caltech University Birds Dataset (cub)
  It is an image dataset with 6033 photos of 200 bird species. It also contains details such as species label, bounding boxes, segmentation and positional attributes which we don't use for our purposes.
- The Oxford Flowers Dataset (flo)
  It is an image dataset of 102 flower categories with between 40 to 358 images per category. It also contains class labels, segmentation masks which we don't make use of for our problem.

We do however need text captions for these datasets which we obtain from the data curated by Reed et al. [2016a].

4.2 Conditioning image generation with captions

We aim to compare the latent codes used to generate the images by Reed et al. [2016b] with our model. In the original model, it has been shown that the text encodings capture the image content whereas the latent noise vector captures the style of the image such as orientation, background, etc. However to illustrate this, Reed et al. [2016b] invert the generator to obtain the noise code z for similar images (based on bird orientation, background color etc) and shows that they have similar noise vectors and that the style can be transferred to other images by simply changing the text encodings. However the style isn't completely disentangled as it is present in very high dimensional space. This can be seen in the Table 1 where variations of the first 5 dimensions of the latent codes don't really offer any drastic style change.

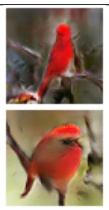
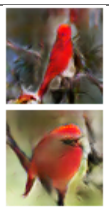
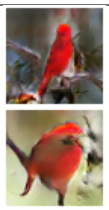
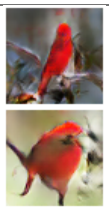
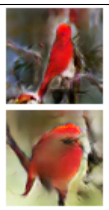
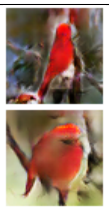
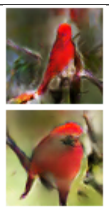
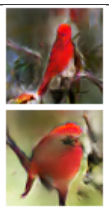
Latent Code	-2	-1	0	1	2
1					
2					
3					
4					
5					

Table 1: Table of generated images with change of the first 5 latent code dimensions from -2 to 2 for the original model.

In Table 1 we use the query text "this vibrant red bird has a pointed black beak". We shall continue to use the same query to make meaningful comparisons.

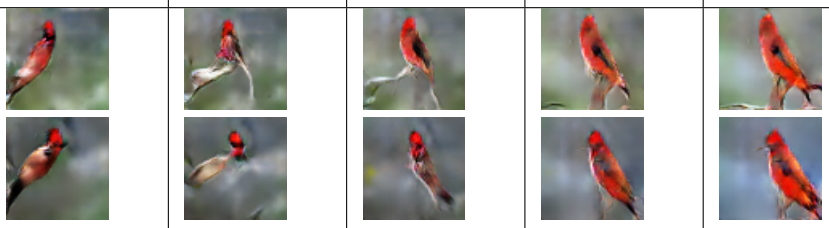
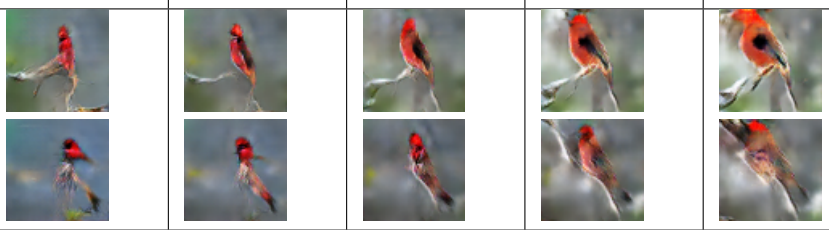
Latent Code	-2	-1	0	1	2
1					
2					
3					

Table 2: Table of generated images with change of the designated 3 continuous latent code dimensions from -2 to 2 for our model.

4.3 Effect of the mutual information criterion

By incorporating the mutual information maximization objective, our model succeeds in disentangling the style variable onto distinct dimensions of the noise vector. In fact, on the birds dataset, it manages to disentangle all the style/content factors mentioned in Reed et al. [2016b] such as orientation, shape and background. The network learns to disentangle these factors onto the latent dimensions we try to maximize the mutual information with. Generations from this model are shown in Table 2. As we can see in Table 2, the first dimension has learnt to disentangle the direction the bird is facing, the second dimension has learnt to disentangle the bird's size, and the third dimension has learnt to disentangle the background.

Our model accomplishes the same effect in the flowers dataset as well. The latent code disentangling of the flowers dataset is shown in Table 3. In Table 3 we use the query text "the center is yellow surrounded by wavy dark purple petals". We see that in the case of the flowers dataset, it has learnt to disentangle elevation of the flower on the first dimension, rotation of the flower with respect to the stem is captured by the second dimension and the elongation along $-45/+45$ degree axis is captured by the the third dimension.

4.4 Other Experiments

For this task we have a couple of hyperparameters which had to be tuned to give relevant results. Some of these are detailed below along with some values we tried for them:

- The Number of continuous latent codes to apply the mutual information criteria on

  Ideally this should be decided based on the structure of the data. For our case, we started of with 5 continuous latent codes to realize that only 3 of those were identifying distinct features such as orientation, background and size. The other 2 were learning a combination of these 3 on the birds dataset. Hence we decided to use 3 continuous latent codes for the mutual information criterion. We decided to use the same number for the flowers dataset as well as there were only about so much variation in that dataset as well.

Latent Code	-2	-1	0	1	2
1					
2					
3					

Table 3: Table of generated images for the flowers dataset with change of the designated 3 continuous latent code dimensions from -2 to 2 for our model.

- Weight of the mutual information criteria λ

  We want the mutual information criteria to be scaled by a factor such that it falls in the scale of the generator and discriminator losses. This avoids the mutual information criteria from dominating the losses, which in turn would affect the stability of the adversarial network. For our problem, we noticed that the mutual information criteria was about 3-10 times larger than the generator/discriminator losses and hence we tested with $\lambda = 0.1$, $\lambda = 0.3$. Both these gave similar results as far as resultant generations are concerned.

- Conditioning Q on the text embeddings

  We found that choosing to condition Q on the text embeddings or not made little difference in the latent code semantics learnt. Even when explicitly providing the text embeddings to the subnetwork that estimates $Q(c \mid x, t)$, the latent codes learnt to capture the same meanings as the ones where Q is conditioned only on the generated image.

4.5 Convergence of the mutual information criterion

As indicated in InfoGAN (Chen et al. [2016]) the mutual information criterion quickly maximizes to a value of around 8-10 depending on the dataset and our choice of λ. This indicates that the bound is tight and that our network maximizes the mutual information quite efficiently. This is shown in the figure 2 for some of our experiments.

5 Conclusion

Our goal was to use a textual description to generate an image corresponding to it, and then adjust it to get reasonable generations. To minimize the number of knobs we need to move to make such adjustments, we considered using the InfoGAN idea to maximize the mutual information between latent codes and the generated image.

We experimented with the Caltech University Birds dataset and the Oxford Flowers dataset, and showed that we are able to learn disentangled and interpretable semantic roles for each latent code. We compare our interpolations to interpolations without the mutual information criterion, and observed

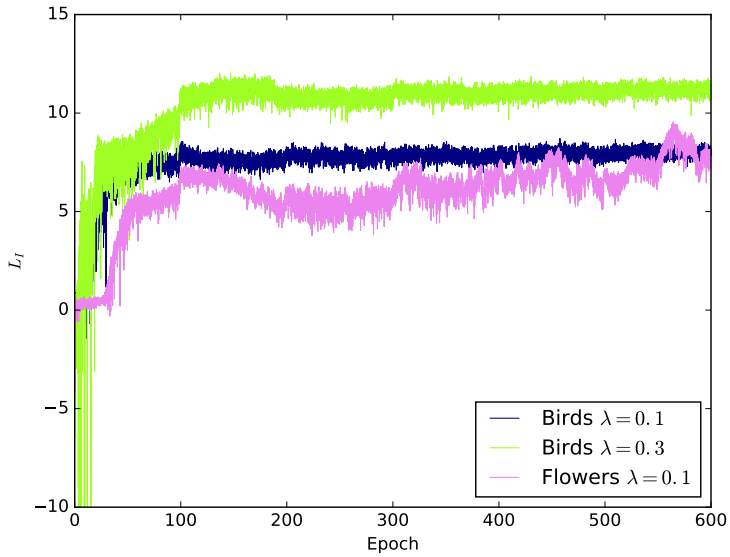


Figure 2: Plot of the mutual information between latent codes and generated image vs. epochs.

Figure 3: A screenshot of a web interface to move sliders to adjust generated images from the caption. The sliders change the values of the latent codes.

that the latent codes encode much more variance suitable for interpretable exploration of the image space.

Source code

We started our implementation from Scott Reed's original codebase for Text to Image Synthesis, from `https://github.com/reedscot/icml2016`.

Our final codebase can be found at `https://github.com/alemc2/text2img`. The `README.md` file in the code repository explains how to learn the model and generate images for queries.

You can also choose to enable the argument `web` of `txt2img_demo.lua` to start a web server where you can submit a query, generate images from that query and then adjust those images with sliders representing their learnt latent codes. A screenshot of this web interface we developed is shown in Figure 3.

References

Caltech-UCSD Birds 200. Technical report. URL http://www.vision.caltech.edu/visipedia/CUB-200.html.

Visual Geometry Group Home Page. URL http://www.robots.ox.ac.uk/~vgg/data/flowers/102/.

X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. *arXiv:1606.03657 [cs, stat]*, June 2016. URL http://arxiv.org/abs/1606.03657. arXiv: 1606.03657.

I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Networks. *arXiv:1406.2661 [cs, stat]*, June 2014. URL http://arxiv.org/abs/1406.2661. arXiv: 1406.2661.

S. Ioffe and C. Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv:1502.03167 [cs]*, Feb. 2015. URL http://arxiv.org/abs/1502.03167. arXiv: 1502.03167.

A. Radford, L. Metz, and S. Chintala. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *arXiv:1511.06434 [cs]*, Nov. 2015. URL http://arxiv.org/abs/1511.06434. arXiv: 1511.06434.

S. Reed, Z. Akata, B. Schiele, and H. Lee. Learning Deep Representations of Fine-grained Visual Descriptions. *arXiv:1605.05395 [cs]*, May 2016a. URL http://arxiv.org/abs/1605.05395. arXiv: 1605.05395.

S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative Adversarial Text to Image Synthesis. *arXiv:1605.05396 [cs]*, May 2016b. URL http://arxiv.org/abs/1605.05396. arXiv: 1605.05396.