Text to Image Synthesis Using Generative Adversarial Networks

Liangjie Cao

July 16, 2018

1. Text Embeddings

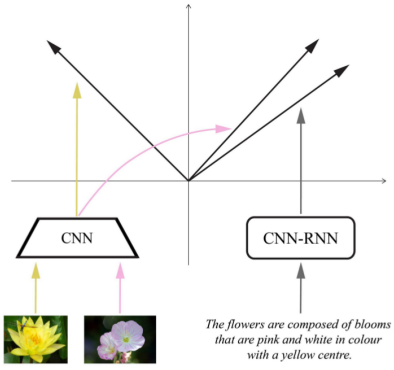

Figure 1. The char-CNN-RNN encoder maps images to a common embedding space. Images and descriptions which match are closer to each other. Here the embedding space is $\mathbb{R}^2$ to make visualisation easier. In practice, the preprocessed descriptions are in $\mathbb{R}^{1024}$

The text descriptions must be vectorised before they can be used in any model. These vectorisations are commonly referred to as text embeddings. Text embedding models were not the focus of this work, and that is why the already computed vectorisations by Reed *et al.* [4] are used. Other state of the art models [5, 6] use the same embeddings and their usage makes comparisons between models easier.

The text embeddings are computed using the char-CNN-RNN encoder proposed in [4]. The encoder maps the images and the captions to a common embedding space such that images and descriptions which match are mapped to vectors with a high inner product. For this mapping, a Convolutional Neural Network (CNN) processes the images, and a hybrid Convolutional-Recurrent Neural Network (RNN) transforms the text descriptions(Figure 1).

A common alternative is Skip-Thought Vectors [2] which is a pure language-based model. The model maps sentences with similar syntax and semantics to similar vectors. Nevertheless, the char-CNN-RNN encoder is better suited for vision tasks as it uses the corresponding images of the descriptions as well. The embeddings are similar to the convolutional features of the images they correspond to, which makes them visually discriminative. This property reflects in a better performance when the embeddings are employed inside convolutional networks.

2. State of the Art Models

2.1. Method

TensorFlow is an open-source library developed by researchers from Google Brain and designed for high performance numerical and scientific computations. It is one of the most widely used libraries for machine learning research. TensorFlow offers both low level and high-level APIs which make development flexible and allow fast iteration. Moreover, TensorFlow makes use of the capabilities of modern GPUs for parallel computations to execute operations on tensors efficiently. Actually I try to use this framework these days.

2.2. GAN-CLS (Conditional Latent Space)

Reed *et al.* [5] were the first to propose a solution with promising results for the problem of text to image synthesis. The problem can be divided into two main subproblems: finding a visually discriminative representation for the text descriptions and using this representation to generate realistic images.

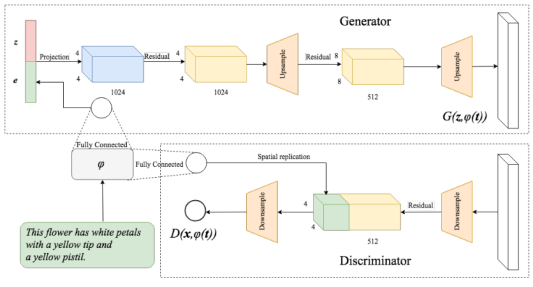

Figure 2. Architecture of the customised GAN-CLS. Two fully connected layers compress the text embedding $\psi(t)$ and append it both in the generator and the discriminator. In the discriminator, the compressed embeddings are spatially replicated (duplicated) before being appended in depth.

The functions $G(z)$ and $D(x)$ encountered in regular GANs become in the context of conditional GANs, $G(z, \psi(t))$ and $D(x, \psi(t))$, where $\psi : \sum \rightarrow \mathbb{R}^{N_\psi}$ is the char-CNN-RNN encoder, $\sum$ is the alphabet of the text descriptions, t is a text description treated as a vector of characters and N_ψ is the number of dimensions of the embedding. The text embedding $\psi(t)$ is used as the conditional vector c.

2.3. Model Architecture

GAN-CLS uses a deep convolutional architecture for both the generator and the discriminator, similar to DC-GAN (Deep Convolutional-GAN) [3].

In the generator, a noise vector z of dimension 128, is sampled from $N(0, I)$. The text t is passed through the function ψ and the output $\psi(t)$ is then compressed to dimension 128 using a fully connected layer with a leaky ReLU activation. The result is then concatenated with the noise vector z. The concatenated vector is transformed with a linear projection and then passed through a series of deconvolutions with leaky ReLU activations until a final tensor with dimension 64*64*3 is obtained. The values of the tensor are passed through a tanh activation to bring the pixel values in the range [-1, 1].

In the discriminator, the input image is passed through a series of convolutional layers. When the spatial resolution becomes 4*4, the text embeddings are compressed to a vector with 128 dimensions using a fully connected layer with leaky ReLU activations as in the generator. These compressed embeddings are then spatially replicated and concatenated in depth to the convolutional features of the network. The concatenated tensor is then passed through more convolutions until a scalar is obtained. To this scalar, a sigmoid activation function is applied to bring the value of the scalar in the range [0, 1] which corresponds to a valid probability.

The focus of the GAN-CLS paper is not on the details of the architecture of the discriminator and the generator. Thus, to obtain better results, the author deviated slightly from the DC-GAN architecture, and I added one residual layer [1] in the discriminator and two residual layers in the generator. These modifications increase the capacity of the networks and lead to more visually pleasant images. Figure 2 shows the architecture of the customised GAN-CLS.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2

[2] R. Kiros, Y. Zhu, R. Salakhutdinov, R. S. Zemel, A. Torralba, R. Urtasun, and S. Fidler. Skip-thought vectors. In *NIPS*, 2015. 1

[3] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *Computer Science*, 2015. 2

[4] S. Reed, Z. Akata, H. Lee, and B. Schiele. Learning deep representations of fine-grained visual descriptions. In *CVPR*, 2016. 1

[5] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. 2016. 1

[6] H. Zhang, T. Xu, and H. Li. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *ICCV*, 2017. 1