SequenceGan text to image synthesis with Seq models and GANs

SequenceGan text to image synthesis with Seq models and GANs

Yiğit Gündüç
Ankara University Development Foundation High School,
Ankara, Turkey
ygunduc@gmail.com

June 6, 2021

Abstract

Generative Adversarial Nets are one of the most popular generative frameworks. In our work, we introduce the SequenceGAN, a method that can generate images based on a given caption by supplying the conditional sequential text input to the generator and the discriminator. Unlike other conditional methods, SequenceGAN uses recurrent layers for better context understanding. We also demonstrated SequenceGANs performance by applying it to the MNIST and Flickr 8k datasets.

1 Introduction

Image synthesis from text is an important field in deep learning and has wide applications. One of the most successful methods benefits from GANs [1], one of the most popular generative frameworks. In a GAN model, two neural networks compete with each other to minimize their losses. GANs produce their samples randomly, new instances of data cannot be influenced by any means. This makes Vanilla GANs unconditional. They produce new instances of data according to a given random noise, therefore, generated images are random among training samples. For instance, if a model is trained on a 10 class dataset, all predictions will be one of those ten classes but cannot produce data associated with a given class on purpose. Recent attempts to control the generated results with the help of techniques from other domains of deep learning like Natural Language Processing have shown promising results [2, 3, 4]. These models and architectures use different methods with the same goal in mind, image synthesis from a given caption or a condition, they use NLP concepts like recurrent neural networks, transformers, attention, and word embeddings to generate realistic and accurate results.

In this work, we introduce the SequenceGAN which lets us control generated results with given sequential text input. Text input might be class labels, image features, or associated captions. As the name explains we use a sequence model encoder to

encode the input, and convolutional layers to decode the encoded vector. A GAN-like generator discriminator architecture is used to ensure that generator generates high-quality images. These combined methods let SequenceGAN have a deep context understanding ability and generate quality images according to given captions.

2 Background

Image synthesis from text has always been a major problem in the field of computer vision and deep learning. There have been some methods that let GANs generate images associated with a given class [5, 6]. There have been works that succeeded significantly thanks to NLP concepts which have been used to help the models [3, 4] with language understanding. VAEs have also shown promising results because of their unique way of mapping and sampling [7].

Most of the text-to-image architectures use simple conditioning methods [5, 6, 8], which is only applicable to a single domain or generates low-quality images.

This work to make a model with a deep context understanding, and applicable to any image-caption dataset with enough training.

3 Preliminaries

subsectionGenerative adversarial networks Generative adversarial networks [1] are generative machine learning models where two adversarial neural networks compete with each other to minimize their loss. The idea in GANs is to train the discriminator with both the original data and the generated data and make it identify which one is the real one which one is fake. In this process discriminator D updates its loss dynamically then this loss passes to the generator G where the G tries to minimize its loss. G never sees the training data itself only makes a guess and sees how well it performs in terms of D loss thus optimes its weight accordingly. Overall during the training stage, G and D plays a min-max game with the following objective.

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

where x, p and z are a real image from the true data, data distribution, and a noise vector sampled from a distribution p_z respectively.

3.1 Sequence Models

Sequence models [9] are a type of machine learning solution to Natural Language Processing. Widely used in areas like language translation, image captioning, conversational models, and text summarization. Sequence Models takes one Sequence and outputs another one. These models consist of encoders and decoders. The encoder takes inputs one at a time and creates a lower-dimensional representation of the inputs via its recurrent layers then passes it to the decoder where it outputs its decisions one at a time.

4 SequenceGAN

SequenceGAN takes two parameters, z, a random noise, and y, as the text input. Text input can be captions, class labels, or any other information that qualifies to generate images on. Like prior conditional generation models, SequenceGAN also uses the text conditional input, y, to generate the desired output. We are not aiming to create a new conditional gan although our model can be used just like an enchanted conditional gan, conditional gan with deeper context understanding. The difference between a traditional conditional generation method and SequenceGAN is SequenceGAN takes sequential language input such as captions or image description and builds the images upon those such that a conditional gan only takes class label attributes as inputs therefore it can only be influenced in a way that can generate a member of the desired class. The application of the text conditional input to both generator and the discriminator lets the model generate the desired output. What makes the SequenceGAN different from other conditional methods is instead of directly applying the condition to the generator and discriminator, Seq2Image uses recurrent layer, embedding, and sequence model architecture to generate a lower-dimensional representation on the input sequence, the context vector, and then applying it instead of the label itself. This method helps the model to have a deeper understanding of the given input. Which helps the model to generate more realistic and accurate images. Please see GitHub repo for implementation details `https://github.com/YigitGunduc/SequenceGAN`.

Figure 1 shows the layout of generator G and discriminator D models. two neural networks(generator and discriminator) adversaries with each other to minimize their losses. In both generator and discriminator, the conditional input y passes to the Seq2Seq encoder where the recurrent layers generate the context vector. The contribution of the condition is to provide a guidance mechanism while producing the output image with the given context. D takes real and fake images alongside the conditional input. This information is used to check if an image is generated according to the given condition.

Generator the sequential language input(desired text to generate caption) passes the generator with the help of an Input layer as a tokenized string representation. Tokenized input passes through a trainable embedding layer which is used to generate a higher-dimensional representation of the given input. The output of the embedding layer is used as the input of the recurrent layers. The generator also takes random noise as input and multiples it with the output of the recurrent layer to produce each sample distinct from the others. These combined inputs pass to the feed-forward layers then reshape to create $2D$ vectors which later on will be processed to be the output image. Lastly, the convolutional block takes the newly formed vector, applies upsampling and convolution operation on them to generate the desired output. Convolutional blocks contain either a convolutional layer followed by an upsampling layer or a convolutional transpose layer. In each case, the convolutional layer is trialed by batch norm and activation function of choice. At the final stage of the generator, a convolutional block with a $tanh$ activation and filter number of desired image channels outputs the generated image.

Both the generator and the Discriminator share the same structure for language understanding. Discriminator, like the Generator, takes the language conditional input and

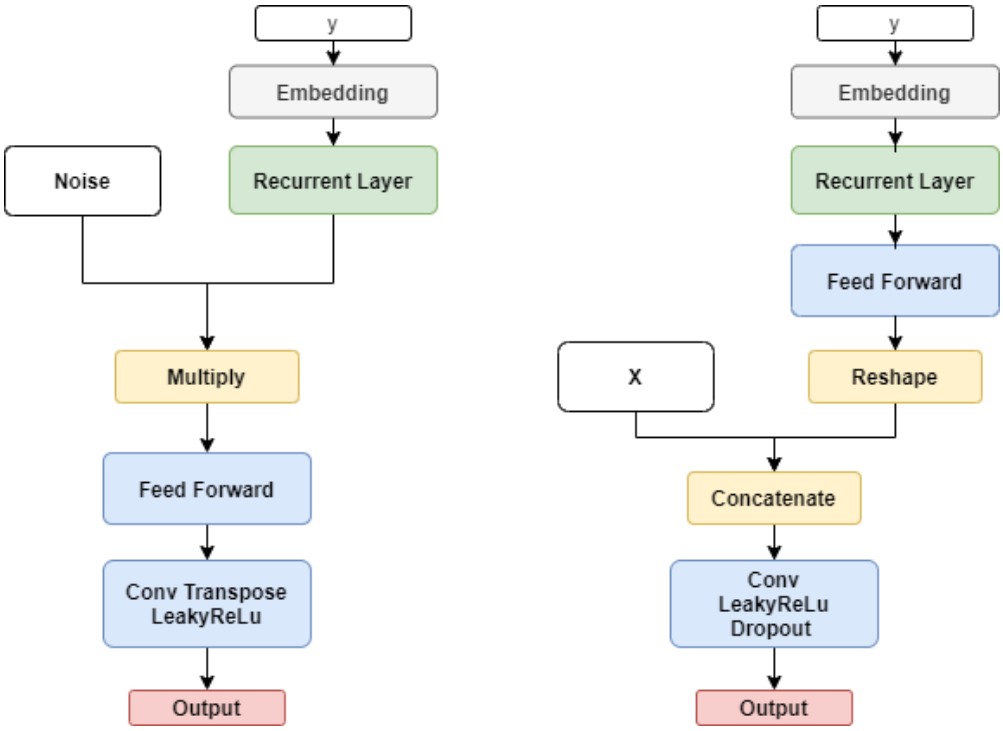

Figure 1: Generator and Discriminator networks.

passes it through the embedding and recurrent layers to create a context vector. The context vector passes to the feedforward layer to generate a higher-dimensional representation. A higher dimensional representation of the context vector is, then, reshaped to have the same dimensions as the original images. Discriminator concatenates both vectors and applies convolution. At the final stage of the Discriminator, a linear layer makes a binary classification on the image. While the discriminator decides the image is real or fake it also questions that do this image is generated according to the given captions.

5 Training and Experimental Results

The model is trained on two datasets to demonstrate its performance. We train the model on the Mnist [10] dataset like a conditional gan(with class labels). It is also trained on Flickr8k [11] with sequential inputs as image captions.

5.1 MNIST

MNIST dataset, a database of handwritten digits that has a training set of $60,000$ examples and a test set of $10,000$ examples, which is a subset of a larger set, called NIST. The MNIST dataset is images divided into 10 separate classes, numbers from 0 to 9. The samples are normalized between 1 and -1, with the shape of $28 \times 28 \times 1$ (width, height, channel). Figure 2 presents reconstructions of the MNIST character set. Each row corresponds to a separate digit, which is under the same class. The class identifiers are fed into SequenceGAN as numbers.

Figure 2: Images generated by SeqeuenceGAN after training on MNIST dataset each row for a class.

5.2 Flickr 8K

Sentence-based image captioning dataset, consisting of $8,000$ images that are each paired with five different captions. All the captions provide a clear description of all the events and objects that take place in the images. Images are reshaped to have shapes of $64 \times 64 \times 3$ (width, height, channel) and normalized between 1 and 0. During the training phase, tokenized captions are fed to the model and expected to generated images from the dataset. SequenceGAN learns to map images during training. At the test stage, it takes captions and tries to understand them with its recurrent layers, and reconstructs the closest sample from the dataset. Figure 3 shows a random selection of original and generated images in pairs with related captions. As it can be seen, the caption leads to the correct understanding of the image and reconstruction of the closest image in the data set.

6 Conclution

In this work, we introduced SequenceGAN, a type of generative model that can understand the natural language and build images upon that. We plan to extend the domain of SequenceGANs to other popular datasets and tasks, enabling it to generate accurate and realizing images according to given inputs sequences that have not been possible before and have a robust position in the world of text-to-image generation. Please see GitHub repo for implementation details `https://github.com/YigitGunduc/SequenceGAN`.

Figure 3: Images generated by SeqeuenceGAN with associated captions.

References

[1] I. Goodfellow, J. Pouget-Abadie, M. Mehdi, et al., Generative Adversarial Networks, COMMUNICATIONS OF THE ACM 139-144, 63, (2020), DOI 10.11453422622

[2] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. Infogan Interpretable representation learning by information maximizing generative adversarial nets. In NIPS, 2016. 2

[3] E. Mansimov, E. Parisotto, J. L. Ba, R. Salakhutdinov, Generating Images from Captions with Attention, arXiv1511.02793, (2016)

[4] A. RAMESH, Aditya, et al. Zero-shot text-to-image generation. arXiv preprint arXiv2102.12092, 2021.

[5] M. Mirza, S. Osindero. Conditional generative adversarial nets. arXiv preprint arXiv1411.1784, 2014.

[6] X. Yan, J. Yang, K. Sohn, and H. Lee. Attribute2image Conditional image generation from visual attributes. In ECCV, 2016. 2

[7] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In ICLR, 2014. 2, 3

[8] A. Odena, C. Olah, and J. Shlens. Conditional image synthesis with auxiliary classifier gans. In ICML, 2017. 2

[9] Sutskever, Ilya; VINYALS, Oriol; LE, Quoc V. Sequence to sequence learning with neural networks. arXiv preprint arXiv1409.3215, 2014.

[10] Yann LeCun, THE MNIST DATABASE of handwritten digits. Courant Institute, NYU Corinna Cortes, Google Labs, New York Christopher J.C. Burges, Microsoft Research, Redmond.

[11] H. Micah, P. Young, J. Hockenmaier. Framing image description as a ranking task Data, models and evaluation metrics. Journal of Artificial Intelligence Research 47, 853-899, (2013)