Generation of Realistic Image (Photo stack)

Ms. Nirmala J S/ Assistant professor Nitte Meenakshi Institute of Technology -Bangalore
Revanth N Mallol/ Nitte Meenakshi Institute of Technology -Bangalore
Nandan M/ Nitte Meenakshi Institute of Technology -Bangalore
Uday Kiran Chari/ Nitte Meenakshi Institute of Technology -Bangalore
Veeresh R K/ Nitte Meenakshi Institute of Technology -Bangalore

Abstract

The framework for measuring productive models in a competitive process in which we simultaneously train two models: a productive model G. which captures the distribution of the data and also a biased model D. The probability that a sample appeared in the training data than G. The training process G is supposed to increase the chances, that D will make a mistake. This slide is accompanied by a small game for two players. In the artificial space problems G and D, there is a unique solution where G accepts the training data distribution and D is equal to 1 2 everywhere. In case G and D are defined using multilayer perceptron's, the whole system can be trained using backpropagation. To build a well-performing generator, which makes use of machine learning algorithms to produce the required outputs.Solve this using neural network to Generate a Photo Realistic Images using GAN (Generative Adversarial Networks). To gain new information from the generated images. The main idea of generative adversarial network can be compared to game of two players - here two players are generator and discriminator framework for measuring productive models in a competitive process in which we simultaneously train two models: a productive model G. which captures the distribution of the data and also a biased model D. The probability that a sample appeared in the training data than G. The training process G is supposed to increase the chances, that D will make a mistake. This slide is accompanied by a small game for two players. In the artificial space problems G and D, there is a unique solution where G accepts the training data distribution and D is equal to 1 2 everywhere. In case G and D are defined using multilayer perceptron's, the whole system can be trained using backpropagation

DOI Number: 10.48047/nq.2022.20.22.NQ10090 **NeuroQuantology2022;20(22): 1147-1155**

Introduction

The crucial job of a Data Scientist is to collect /create data as much as possible so that we can train the model and get better accuracy. Used to solve real world problems, pre-collected data is not useful. Insufficient data may lead to low accuracy or inefficient use of the model. The key task of a data scientist is to collect as much generated data as possible so that we can train the model and get better accuracy. They are used to solve real world problems pre-collected data is not useful. Insufficient

data can lead to low accuracy or inefficient use of the model. Uber and Google's ML-based self-driving cars are trained using synthetic data. In the research department, synthetic data helps you develop and deliver innovative products for which the necessary data may not be available.

In the research department, synthetic data helps to develop and deliver innovative products for which the necessary data may not be available. The key task of a data scientist is to collect as much generated

data as possible, so that we can train the model and achieve better accuracy. Synthetic data is used to solve real-world problems, as pre-collected data may not be useful. Insufficient data can lead to low accuracy or inefficient use of the model.

Automatically synthesizing realistic images is an extremely difficult task, and even the most advanced AI/ML algorithm suffers from this expectation. Privacy, training, testing, use of generated images for sale in stores.
Generative Adversarial Networks (GANs) are a type of machine learning model that are often used for generating new, synthetic images that are similar to a training dataset. GANs consist of two neural networks: a generator network and a discriminator network. The generator network produces synthetic images, while the discriminator network tries to determine whether an image is real or generated. The two networks are trained together, with the generator trying to produce images that are indistinguishable from real ones, and the discriminator trying to correctly classify the real and generated images. To generate a realistic image using a GAN, you would need to first train the GAN on a large dataset of real images. During training, the GAN will learn to capture the characteristics and features of the real images, and will use this knowledge to generate new, synthetic images that are similar to the real ones. Once the GAN is trained, you can use it to generate new images by providing it with a random noise input and letting it generate an image based on that noise. The resulting image should be a realistic, synthesized version of an image from the training dataset.

The Photo Stack, utilizing the technology of StackGAN (Generative Adversarial Networks), creates photo-realistic images.

Literature Survey

In 1959, David Hubel and Torsten Wiesel described the "simple cells" and "complex cells" of the human visual cortex. They proposed that both cell types are used for pattern recognition. A "simple cell" reacts to edges and stripes in a certain direction. "Complex cells" also respond to edges and bars in a certain direction, but differ from simple cells in that these edges and bars can be moved around the scene and the cells continue to respond. For example, a simple cell might respond only to a horizontal bar at the bottom of the image, and a complex cell might respond to a horizontal bar at the bottom, middle, or top of the image. This property of complex cells is called "spatial invariance". Your paper must be in two
column formats with a space of 4.22mm (0.17") between columns.
[8] Generative Adversarial Nets. Goodfellow Ian; Pouget-Abadie Jean; Mirza Mehdi; Xu Bing; Warde-Farley David; Ozair Sherjil; Courville Aaron; Bengio Yoshua (2014). Output generated by this model is better than the previous models. [9] NIPS 2016: Generative Adversarial Networks Ian Goodfellow Highlights the Advantages of GAN's over other networks.
[10] Conditional GAN. Simon Osindero; Mehdi Mirza; Outputs generated by this model can controlled via text or any other inputs.
[11] Improved techniques for training gans 2016 Salimans, Tim. The main contributions of the paper are the use of weight normalization, which is a method to normalize the weights of the networks to improve the stability of the training process. The use of mini-batch discrimination, which is a method to introduce diversity into the generated samples by forcing the generator to consider multiple examples at once.
[12] Wasserstein Loss 2017 Arjovsky, Martin, Soumith Chintala, and Léon Bottou. Using the Wasserstein Loss in GANs can improve the stability of the training process, by reducing the problem of vanishing gradients or mode collapse that commonly affects traditional GANs which use the Jensen-Shannon divergence.
[13] Spectral normalization for generative adversarial networks 2018 Miyato, Takeru. The idea behind Spectral Normalization is to bound the Lipschitz constant of the generator or discriminator neural network. Lipschitz constant is a mathematical measure of how much the

output of a function can change for a given change in its input, and it's used to ensure that the generator and discriminator networks don't change too quickly during training.
[14] Self-attention generative adversarial networks 2018 Zhang, Han. The idea behind self-attention is to allow the network to focus on specific parts of an image while generating new images. In the generator, the self-attention layers are used to attend to different parts of the input noise, allowing the generator to generate more fine-grained details in the generated images. In the discriminator, the self-attention layers are used to attend to different parts of the real images, allowing the discriminator to better distinguish real images from fake images.

[15] Large Scale GAN Training Andrew Brock, Jeff Donahue, Karen Simonyan 2018. BIGGAN is trained using a two-level hierarchy of generator and discriminator networks. The top-level generator network generates low-resolution images, while the bottom-level generator network generates high-resolution images. The discriminator network evaluates the images generated by both generator networks and provides feedback to help improve the quality of the generated images.

[16] Began: Boundary equilibrium generative adversarial networks David, Thomas Schumm, and Luke Metz. It is a type of GAN that aims to stabilize the training process and generate high-quality images by using a novel equilibrium-seeking objective function. The BEGAN model can learn to generate images that are very similar to real images, and is particularly well-suited for image generation tasks such as creating realistic images of faces or animals.
[17] An Intelligent Method for Predicting the Pressure Coefficient Curve of Airfoil-Based Conditional Generative Adversarial Networks 2021. Convergence will be faster. Even the random distribution that the fake images follow will have some pattern. You can control the output of the generator at test time by giving the label for the image you want to generate.
[18] Synthetic to Real World Image Translation Sreedhar Radhakrishnan C-C Jay Kuo. The authors use a GAN trained on real-world images to generate synthetic images that are like real-world images. The goal is to use these synthetic images to train other computer vision models, such as object detectors, that can perform well on real-world images despite being trained on synthetic images.
[7]AttnGAN, which was introduced in a 2017 research paper by Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, Xiaodong He, and Le Song, uses attention mechanisms in the generator network to focus on specific parts of the input text while generating the image. The generator network is trained to generate images that are consistent with the input text, and the attention mechanisms allow the network to focus on specific words or phrases in the text that are relevant to the image.
[6],[19] DF-GAN, which was introduced in a 2019 research paper by Kan Chen, Jiajun Wu, and Ying Nian Wu, also uses attention mechanisms but in a different way. DF-GAN uses a multi-stage training process, where in the first stage, the generator network is trained to generate diverse images from the input text, and in the second stage, a separate attention network is trained to focus on specific regions of the images that are consistent with the input text.
[4],[20] A Comparison between AttnGAN and DF GAN: Text to Image Synthesis Philo Sumi Sindhuja S Sureshkumar S. Both AttnGAN and DF-GAN have demonstrated the capability to generate high-quality images from text descriptions, but the two methods have slightly different design decisions, AttnGAN focuses more on attention mechanisms in the generator whereas DF-GAN focus more on diverse generation in the first stage and attention network in the second stage.
[5]GAN-powered Deep Distributional Reinforcement Learning for Resource Management in Network Slicing Yuxiu Hua, Rongpeng Li, Zhifeng Zhao, Xianfu Chen, and Honggang Zhang. GAN-DDRL uses a GAN to generate a distribution of states that represents the possible outcomes of the network slicing problem. The generated distribution is then used to train a deep

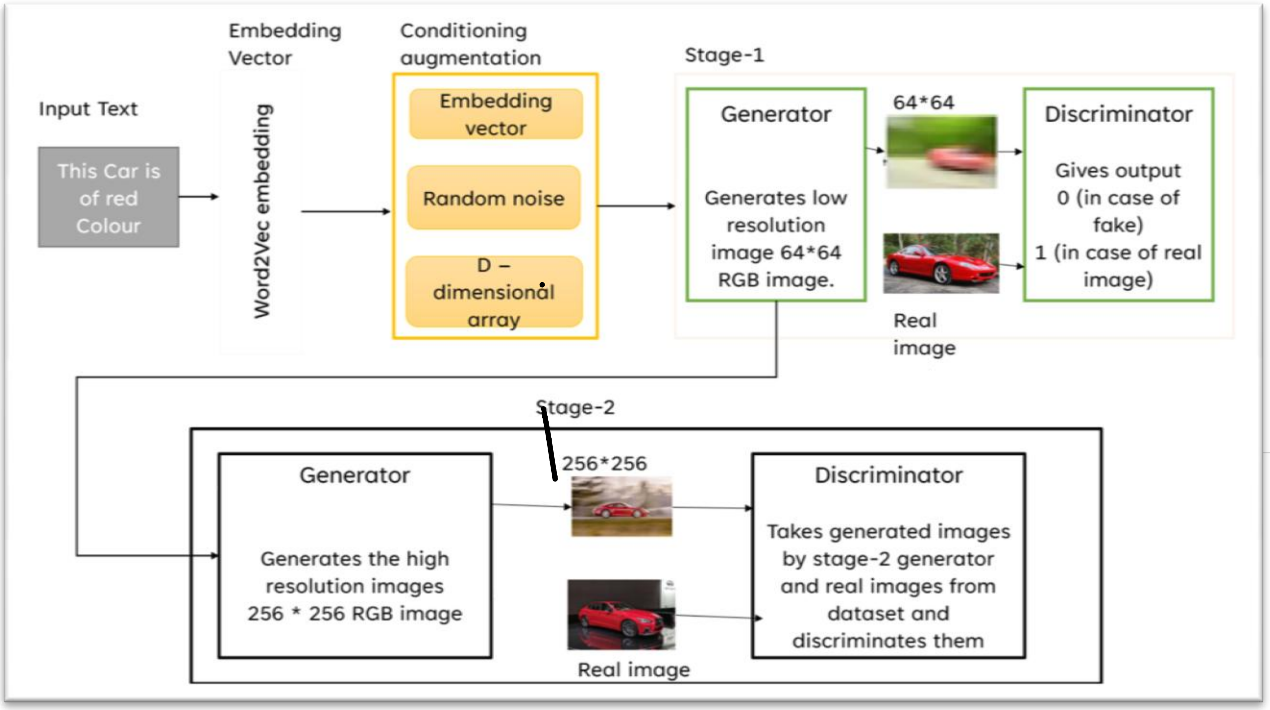
reinforcement learning (DRL) agent to make decisions about how to allocate resources in the network. By incorporating the GAN into the DRL process, the agent can learn from a much broader range of possible scenarios, which improves its ability to make accurate and effective resource allocation decisions.

Methodology

Deep learning is a very CPU intensive program thing to be running, so be prepared to shell out a lot of money for a good enough system. Here are some system requirements to adhere to (parenthesis are what you can maybe get away with) Quad-core Intel Core i7 Skylake or higher (Dual-core isn't the best for this kind of work, but it's manageable) 16GB RAM (8GB is fine, but not for the performance you might want or expect) M.2 PCIe or regular PCIe SSD with at least 256GB of storage, although 512GB is best for performance. The faster you can load and save your applications, the better your system will perform. (SATA III will hinder system performance) Premium graphics cards so something with a GTX 980 or 980Ms would be best for a laptop and a 1080 or 1070 would be best for a desktop setup. (Try not to sacrifice too much here. While the 980TI or 970m might be cheaper, it's also a critical part of the system and otherwise you'll see a drop in performance.).

Fig. 1 shows Stack GAN is a variant of GANs that was specifically designed for generating high-resolution images. It consists of two main components: a Stage-I GAN and a Stage-II GAN. The Stage-I GAN generates a low-resolution version of the target image, while the Stage-II GAN takes the output of the Stage-I GAN and generates a high-resolution version of the image. The Stage-I GAN consists of a generator network and a discriminator network, just like a standard GAN. The generator network takes a random noise vector as input and produces a low-resolution image as output. The discriminator network tries to determine whether the image is real or generated and provides feedback to the generator network to help it improve its image generation capabilities. The Stage-II GAN also has a generator and discriminator network, but it is trained to take the output of the Stage-I GAN as input, rather than a random noise vector. The Stage-II generator produces a high-resolution version of the image, and the discriminator network tries to determine whether the image is real or generated. Together, the Stage-I and Stage-II GANs work to generate high-resolution images that are realistic and faithful to the training data. The Stage-I GAN provides the basic structure and features of the image, while the Stage-II GAN refines the details and texture to produce a high-resolution version of the image.

eISSN 1303-5150



The two player game where one tries to minimize the loss while the other tries to maximize it. The loss function is defined as below:

Min G max D V(D,G) =

$E_{x\sim Pdata(x)} \log D(x) + E_{Z\sim} p_{z(Z)}[\text{Log}1\text{-}D](G(Z)))]$ [1]

The binary cross-entropy loss is used by both the generator and the discriminator during their training process.

$$L(y, \hat{y}) = -(y.log\hat{y} + (1-y).\log(1-\hat{y}))$$

The Discriminator's job is to classify between real and fake so it tries to minimize this loss

The Stage-I generator takes in a text description and generates a low-resolution version of the desired image. The Stage-II generator then takes both the text description and the output of the Stage-I generator as input and produces a high-resolution version of the image.

One key aspect of the StackGAN model is its use of a "conditioning augmentation" technique, in which the text description is encoded and concatenated with the noise vector used to generate the image. This allows the model to better capture the underlying structure of the text and produce more realistic images.

The Stage-I generator is trained using a combination of supervised and unsupervised learning, while the Stage-II generator is trained using only supervised learning. This allows the model to improve its performance as it progresses through the two stages of training.

- Overall, the StackGAN model is able to generate high-quality images that are highly correlated with the input text descriptions and has been successful in a variety of image generation tasks. Generative Adversarial Networks are a recent development and have shown huge promises already.
- It is an active area of research and new variants of GANs are coming up frequently.
- In complex domains or domains with a limited amount of data, generative modeling provides a path towards more training for modeling.
- GANs have seen much success in this use case in domains such as deep reinforcement learning.

Implementation

There are a few key steps involved in implementing the StackGAN model:

1. Pre-processing: The first step is to pre-process the data, which includes tokenizing the text descriptions and converting them into a numerical representation that can be used as input to the model.
2. Model architecture: Next, you'll need to define the architecture of the StackGAN model, including the number of layers, the type of layers (e.g., convolutional, fully connected), and the number of filters in each layer.
3. Training: Once the model is defined, you'll need to compile it and specify the loss function and optimizer to use during training. You'll also need to define the training and validation datasets and specify the number of epochs to train for.
4. Evaluation: After training, you can evaluate the performance of the StackGAN model using metrics such as reconstruction loss and perceptual quality. You can also generate new images by providing the model with text descriptions and visualizing the output.
5. Fine-tuning: You may want to fine-tune the model by adjusting the architecture or training hyperparameters to improve performance on a particular task. This can involve repeating some of the steps above, such as adjusting the model architecture or re-compiling the model with a different loss function or optimizer.

Results and Discussion

The output of the StackGAN model is an image generated from a text description and a noise vector. The generated image should be highly correlated with the input text and should have a high level of perceptual quality, meaning that it should look realistic to a human observer.

The output image is produced by the generator network of the StackGAN model, which consists of multiple layers of convolutional and fully connected layers. The generator is trained to produce images that are indistinguishable from real images, so the output image should be similar to a real image of the object or scene described in the input text.Fig.2 shows the output result of StackGan model and Fig.3 shows the output result of other models (Dcgan ,Txt2img,Attngan)

It is important to note that the output image will not necessarily be an exact replica of the object or scene described in the text. Instead, it will be a generated version of the object or scene that is based on the input text and the underlying structure of the StackGAN model.

As we can see the comparison results in Fig. 4. we can conclude that stackGAN is more effective and faster in learning than its variants.

Fig. 1 Sample Images generated by our model(stack GAN)

A bird with bright yellow and colours of orange on tail	
The bird has bright yellow body brown wings	
The bird is red with lack wings	
A bird with blue feathers and blak tail	
	A) dc gan) B) Text2Img C) Attngan Fig3: Images generated by other models and stackgan

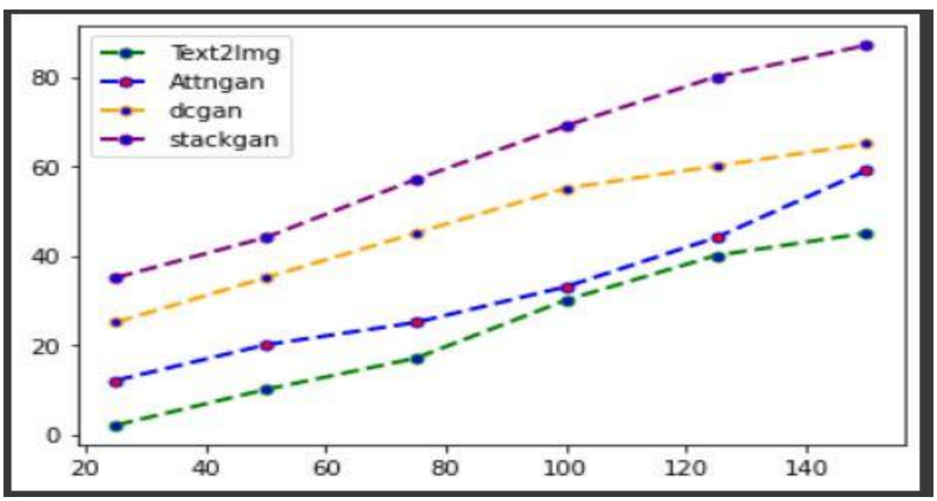


Fig4: Illustration of accuracy on various GAN's

Conclusion

Stacked Generative Adversarial Networks, Stack GAN-v1 and StackGAN-v2 were designed to break down the difficult problem of creating realistic high-resolution images into more manageable subproblems. StackGAN-v1 with conditioning extensions was first proposed for text-to-image stacking with a novel sketch enhancement process. It is possible to generate a 256x256 resolution image with realistic details from the text description.

Overall, the StackGAN model has been successful in generating high-quality images that are highly correlated with the input text descriptions, and has been applied to a variety of image generation tasks. It is an important contribution to the field of generative models and has opened up new possibilities for generating images from text descriptions. STACK GAN IS A RESEARCH MODEL DEVELOPED BY ZHANG, XU, AND SAENKO IN THEIR PAPER "STACKGAN: TEXT

References

[1] Goodfellow, Ian; Pouget-Abadie, Jean; Mirza, Mehdi; Xu, Bing; Warde-Farley, David; Ozair, Sherjil; Courville, Aaron; Bengio, Yoshua (2014). Generative Adversarial Nets (PDF). Proceedings of the International Conference on Neural Information Processing Systems (NIPS 2014). pp. 2672–2680.

[2] Isola, Phillip; Zhu, Jun-Yan; Zhou, Tinghui; Efros, Alexei (2017). "Image-to-Image Translation with Conditional Adversarial Nets". Computer Vision and Pattern Recognition.

[3] A Gentle Introduction to Generative Adversarial Networks (GANs) Machine Learning Mastery by Jason Browniee.

[4] A Comparison between AttnGAN and DF GAN: Text to Image Synthesis Philo Sumi Sindhuja S Sureshkumar S.

[5] GAN-powered Deep Distributional Reinforcement Learning for Resource Management in Network Slicing Yuxiu Hua, Rongpeng Li, Zhifeng Zhao, Xianfu Chen, and Honggang Zhang.

[6] DF-GAN, which was introduced in a 2019 research paper by Kan Chen, Jiajun Wu, and Ying Nian Wu, also uses attention mechanisms but in a different way.

[7] AttnGAN, which was introduced in a 2017 research paper by Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han

eISSN 1303-5150

Zhang, Zhe Gan, Xiaolei Huang, Xiaodong He, and Le Song.
[8] Generative Adversarial Nets. Goodfellow Ian; Pouget-Abadie Jean; Mirza Mehdi; Xu Bing; Warde-Farley David; Ozair Sherjil; Courville Aaron; Bengio Yoshua (2014).
[9] NIPS 2016: Generative Adversarial Networks Ian Goodfellow Highlights the Advantages of GAN's over other networks.
[10] Conditional GAN. Simon Osindero; Mehdi Mirza; Outputs generated by this model can controlled via text or any other inputs.
[11] Improved techniques for training gans 2016 Salimans, Tim. The main contributions of the paper are the use of weight normalization.
[12] Wasserstein Loss 2017 Arjovsky, Martin, Soumith Chintala, and Léon Bottou.
[13] Spectral normalization for generative adversarial networks 2018 Miyato, Takeru. The idea behind Spectral Normalization is to bound the Lipschitz constant of the generator or discriminator neural network.
[14] Self-attention generative adversarial networks 2018 Zhang, Han.
[15] Large Scale GAN Training Andrew Brock, Jeff Donahue, Karen Simonyan 2018.
[16] Began: Boundary equilibrium generative adversarial networks David, Thomas Schumm, and Luke Metz.
[17] An Intelligent Method for Predicting the Pressure Coefficient Curve of Airfoil-Based Conditional Generative Adversarial Networks 2021.
[18] Synthetic to Real World Image Translation Sreedhar Radhakrishnan C-C Jay Kuo.
[19] A Mathematical Introduction to Generative Adversarial Nets (GAN) Yang Wang
[20] Improving Text-to-Image Synthesis Using Contrastive Learning Hui Ye1, Xiulong Yang1, Martin Takác, Shihao Ji

eISSN 1303-5150