International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 05 | May 2020 www.irjet.net p-ISSN: 2395-0072

Self-Attention Generative Adversarial Network: The Latest Advancement in GAN

Aishwarya Poojary[1], Ankesh Phapale[1], Roshni Salpekar[1] , Sonal Balpande[2]

[1]*Student,Dept.of Computer Engineering, Excelsior Education Society's K.C College of Engineering and Management Studies and Research, Maharashtra, India*
[2]*Professor,Dept.of Computer Engineering, Excelsior Education Society's K.C College of Engineering and Management Studies and Research, Maharashtra, India*

Abstract - *Generative Adversarial Network (GAN) is one of the most fascinating ideas in the history of machine learning. So far the results of GAN are obtained with the help of Convolution Neural Networks (CNN). But there are few drawbacks of CNN which gave rise to a new mechanism 'self-attention'. In this paper, we have briefly discussed the weaknesses of Convolution Generative Adversarial Networks (CNN-GAN), how Self-Attention Generative Adversarial Networks (SAGAN) came into picture, what is it all about and why it is better than CNN-GAN.*

Key Words: **GAN, SAGAN, generator, discriminator, datasets, images,**

1. INTRODUCTION

Image composition is definitely a vital issue in the field of Computer Vision. GANs have greatly contributed to solve some of these problems with the help of deep convolution networks. GAN has two main models: a generator which produces required output and a discriminator is trained to distinguish real data from generators output. The generator keeps fooling discriminator till it reaches to a stage where the discriminator classifies fake generated image as real. However, it was observed that in CNN-GAN models, it is much more tedious to model some image classes when trained on multi-class datasets (e.g., ImageNet). For example, while the state-of-the-art ImageNet GAN model excels at synthesizing image classes with some structural constraints (e.g., ocean, sky and landscape classes which are distinguished more by texture than by geometry), it could not capture geometric or structural patterns that occur consistently in some classes [1]. The actual reason behind this is that all the previous models highly relied on convolution to model the dependencies across different image regions. Since the convolution operator has a local receptive field, long range dependencies can only be processed after passing through several convolutional layers [1].This makes GAN more unstable. The solution to balance computational efficiency and have a large receptive field at the same time is Self- Attention**.** It helps create a balance between efficiency and long-range dependencies that are equal to large receptive fields with the help of a mechanism from NLP called attention. With this approach, the generator can create images with fine details at every point coordinated with distant portions of the image [1]. Also the discriminator could more accurately enforce complicated geometric constraints on the global image structure [1].

2. GENERATIVE ADVERSARIAL NETWORK

A Generative Adversarial Network, or GAN, is neural network architecture for generative modeling. It is used to train two main models: generator and discriminator.

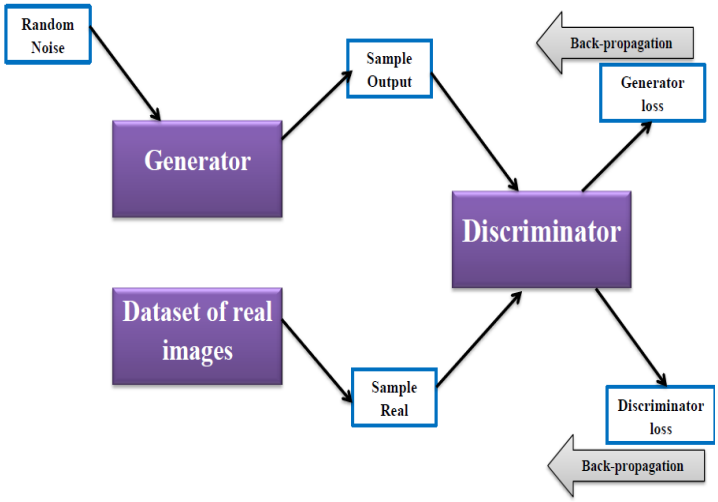

Fig -1: Working of GAN

Generator generates new persuasive data with the help of random input and this becomes a training example for discriminator. Discriminator on the other hand compares generator's output with the real data. The generator output is given as input to the discriminator. It learns to differentiate the generator's false data from real image data. During the training process, the generator keeps generating false image data and the discriminator quickly learns to predict that it's fake. As training progresses, the generator starts generating output that is very much similar to real image and the discriminator starts failing in differentiating between real and fake images. And finally, if generator training goes well, the discriminator starts failing at predicting the difference between real and fake. It begins to

classify fake data as real, and its accuracy decreases. Through back-propagation, generator and discriminator both update their weights from their respective losses.

3. SELF-ATTENTION GENERATIVE ADVERSARIAL NETWORK (SAGAN)

Self-Attention Generative Adversarial Network (SAGAN) is an attention-driven, long-range dependency modeling mechanism for image generation tasks. The existing convolutional GANs produces high-resolution details as a function of only spatially local points in lower-resolution feature maps.[1] Researchers observed that CGANs could easily generate images with a simpler geometry like Ocean, Sky etc. but failed on images which had some specific geometry . For example: CGAN was able to produce the texture of furs of dog but was not able to generate distinct legs. The problem arises because in a convolution mechanism, it is impossible for any output on the top-right pixel position of image to have any relation to the output at bottom-right pixel. Trying to solve this problem could lead towards the decrement of computational efficiency and would also make GAN training unstable. Hence SAGAN was introduced. In SAGAN, using cues from all feature locations, details can be generated. Moreover, the discriminator can figure out whether the highly detailed features in distant portions of the image data are consistent with each other or not [1]. It is a complementary to existing convolution GAN mechanism.

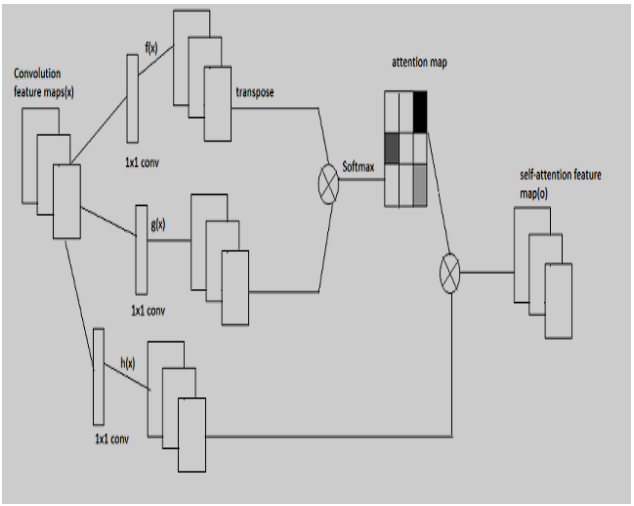

Fig -2: The proposed self-attention mechanism

In the proposed mechanism, the feature map obtained from previous convolution layer is passed through three 1x1 convolutions separately. After passing through them, three feature maps are obtained f, g and h. Now the self-attention is performed over it. Transpose of 'f' is calculated and matrix-multiply it by the 'g' and take the soft-max on all the rows. So we get an attention map as a result which is then multiplied with the 'h' vector and an output self-attention feature map is obtained. At last multiply the final output by a learnable scale parameter and add back the input as a residual connection. For controlling the gradients, spectral normalization is applied to the weights of both generator and discriminator. Two-timescale update rule (TTUR) was used which simply uses different learning rate for both discriminator and generator. Self-attention layer helps the network to capture every fine detail from even distant parts of image and overcome the limitations of existing convolution GAN method.

4. PROJECTS IMPLEMENTED USING SAGAN

4.1 Improving Human Pose Estimation using Self-Attention GAN

Human pose estimation is a technique used to locate human body joints like wrist, elbows, etc. from given input images or videos. Much research is done to improve the estimation. All the research included ambiguities as few complex joints are not visible. So, to remove these ambiguities and improve performance XIANGYANG WANG [2] has used SAGAN in their project. Generator and discriminator are the two main models used in the network. The generator is used to generate heat-maps which give the probability of the keypoint (human body joint) being present in the input images. The trained discriminator computes the loss between predicted heat-maps and the ground truth heat-maps to distinguish real images and fake images.

4.1.1 Generator

The generator consists of four stacked hourglass networks [3]. Each hourglass captures local and global information or features across all scales. The input image goes through repeated top-down and bottom-up processing along with intermediate supervisions. Intermediate supervision is performed at the end of each hourglass to improve performance. The network uses skip connections to preserve spatial information associated with keypoint resolution. The network begins with convolutional layers that are used to capture features. While moving down the layers the performance is degraded as the gradients vanish. To preserve these gradients residual model is used. In the residual step up, the output of previous layers is given to the next new layers thereby preserving the gradients. 256 features are given as output by the residual model. All these features even contain background information and this adds redundancy during estimation. So, to refine the features max pooling is used. Max pooling performs down sampling to reduce feature dimensions that are sufficient for accurately estimating keypoints. Resolution of output image is reduced to 64×64 after max pooling.

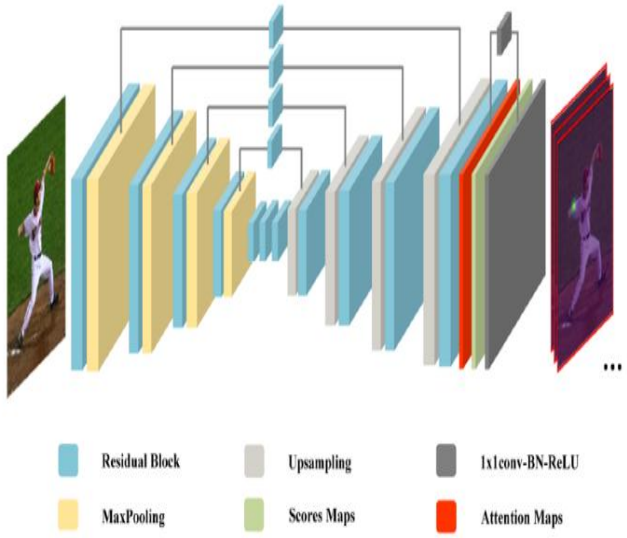

Fig -3: Overview of the Generator Framework [2]

4.1.2 Discriminator

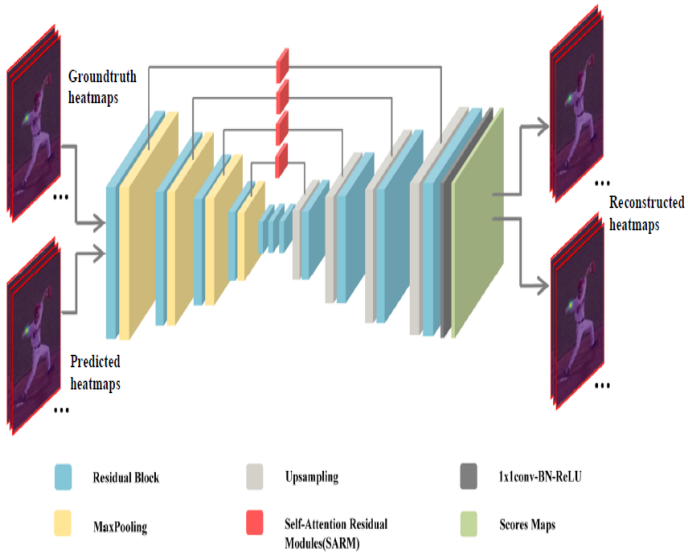

Fig -4: Overview of the Discriminator Framework [2]

One of the four hourglass networks is used as a discriminator. The discriminator is used to distinguish between the generator generated samples and real images. The hourglass thus includes attentional techniques of classifying real and fake images as opposed to the standard hourglass network that uses the residual model. The framework of the discriminator is shown in Fig.4.

4.1.3 Improvements using Self-Attention GAN

Since the convolution layers process only that part of the image that is visible to that particular unit of the network, hence several convolution layers are needed for processing. But the increased layers result in failing. So for modeling long-range dependencies, the self-attention mechanism was introduced.

4.2 Video Game Level Generation using conditional embedding SAGAN (CESGAN) [4]

CESGAN is used to design video game level generation which is unplayable or tough. And further used for General Video Game AI Framework. GAN based models for level generations were previously built using convolutional layers. A convolutional layer is a local operation and its correlation depends on the spatial size of the kernel. For example, it is difficult for an output on the top-left position to have any correlation to the output at bottom-right in a convolution operation for level generation. A deep convolution network with many layers would be required which will increase the large search space. This phenomenon has increased for video games level generation where tiles/pixels located at a distance away from each other (e.g. avatar-door-key) must be correlated to produce a playable level. An impulsive solution to this problem could be reducing the kernel's sizes and layers located deeper in the network to bring ease to capture this relationship later. However, this approach would result in more number of layers of the deep neural network and thus make the GAN training more unstable.

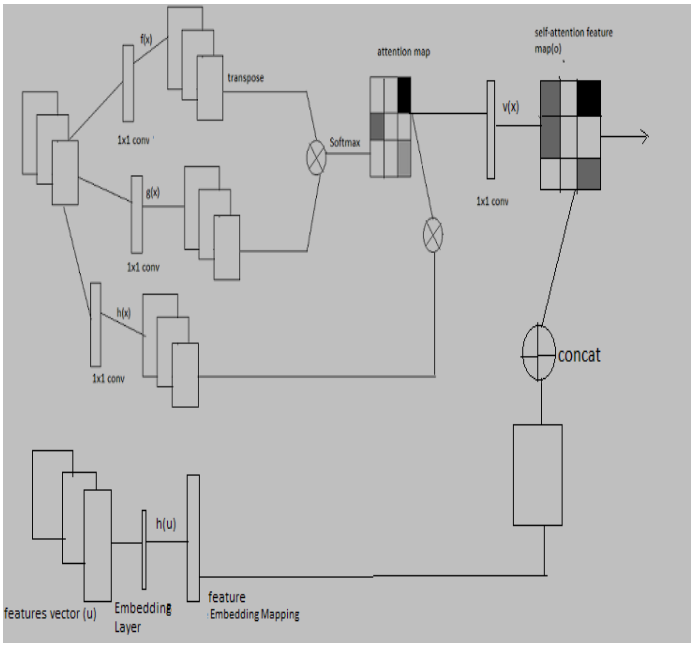

Fig -5: Architecture for Conditional Embedding Self-Attention GAN (CESAGAN)

The CESAGAN network uses 1×1 convolutions in the discriminator and 1×1 de-convolutions in the generator. We use a simple fully connected layer that is concatenated with the self-attention feature map for conditional embedding layer. Bootstrapping is a technique which can be used along

with CSEGAN. This mechanism raises the number of training examples after clearing a playability and diversity test. It enhances the quality of the GAN's discriminator.

5. CONCLUSION

The emerging idea of SAGAN is rapidly moving towards the ladder of success, aiming to ease image composition like never before. With its ability to overcome drawbacks of traditional convolution GAN, this mechanism has started gaining momentum in the domain of Computer Vision. Our paper presented an overview of GAN, Self-GAN mechanism and two of its real-life implementations.

REFERENCES

[1] Zhang, Ian Goodfellow, Dimitris Metaxas and Augustus Oden, "Self-Attention Generative Adversarial Networks", arXiv: 1805.08318v1 [stat.ML], 21 May 2018.

[2] Xiangyang Wang, Zhongzheng Cao, Rui Wang, Zhi Liu and Xiaoqiang Zhu, "Improving Human Pose Estimation with Self-Attention Generative Adversarial Networks", IEEE international conference on Multimedia and Expo Workshop, 2019

[3] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation", In ECCV, 2016.

[4] Ruben Rodriguez Torrado, Ahmed Khalifa, Michael Cerny Green, Niels Justesen, Sebastian Risi and Julian Togelius, "Bootstrapping Conditional GANs for Video Game Level Generation", arXiv:1910.01603, 3 October 2019.