AFEKA TEL-AVIV ACADEMIC COLLEGE OF ENGINEERING

Self-Attention in Deep Convolutional Generative Adversarial Networks

Uria Levkovich | Chen Avraham | Rotem Bar | Maor Bachar

Afeka College of Engineering,
Department Of Intelligent Systems
Computer Vision Course

Abstract This article will cover an approach for image generation task called self-attention GAN. This approach used long-range dependency modeling for the image generation task. The SAGANs rely on the regular or traditional convolutional GANs. Those traditional GANs can generate high-resolution details as a function of only spatially local points in a lower-resolution feature map. By using SAGANs, additional details can be generated by focusing on the location of those features. We will demonstrate whether a self-attention mechanism can improve the quality and visibility of generated images and the FID score. We performed several experiments by using the Stanford Dogs Dataset.

Keywords – Generative Adversarial Network; self-attention; Frechet Inception Dist; image generation task.

1. Introduction

Image generation or image synthesis is a major problem in computer vision. There has been remarkable progress in this task. As a result, a growing number of generative models have been developed, including the Varinonal AutoEncoder (VAE) [1][2], Diffusion [3][4], and Generative Adversarial Networks (GANs) [5. There are many advantages and disadvantages to those models, but over time, the GAN model has become one of the most popular and most widely used models.

In most GAN-based image generation models, convolution layers are used. However, by examining the generated samples from these models, we can observe that convolutional GANs have more difficulty modeling some images classes than others when trained on multi-class datasets such as ImageNet. One possible explanation for this is that these models rely highly on convolution to model the dependencies across different image regions. The convolution operator has a local receptive field which leads to long-range dependencies that can be processed after passing through several convolution layers. This prevents learning about long-term dependencies. This article discusses Self-Attention GANs, which introduce a self-attention mechanism to convolutional GANs. By using Self-Attention, we can achieve both computational efficiency and a large receptive field simultaneously. This helps to model long-range, multi-level dependencies across image regions. A generator with self-attention can generate images with fine details at every location with distant portions of the image. Also, discriminators with self-attention could more accurately enforce the complicated constraint on the image structure.

2. Knowledge Background

This section will present the GAN model and how the attention layers in the GAN model work.

2.1. GAN

Generative Adversarial Networks or GAN is a neural network architecture for generative modeling proposed in 2014 by [5]. GANs networks contain two main models: generator and discriminator.

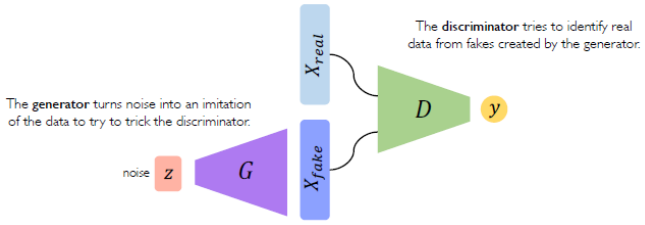


Fig 1: the two models which are learned during the training process for a GAN-the discriminator (D) and the generator (G)

Given a set of target samples, the generator generates relevant persuasive data from a random input. These generated samples become training examples for the discriminator. On the other hand, the discriminator compares the generator's output with the real data. The discriminators learn to differentiate the true image data from the generator's fake data. These two neural networks duel each other in the training process, the generator generating fake data and the discriminator learning to predict if data is authentic or fake. As training progresses, the generating image produced by the generator becomes more similar to the real data. As a result, the discriminator starts failing in differentiating between real data and generated data.

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

Equation 1 describes the two-player min-max game value function V (G, D) played by the discriminator (D) and the generator (G). We train D to maximize log(D(x)), meaning maximize the probability of assigning the correct label to real training data and samples from G. We simultaneously train G to minimize log(1-D(G(z))), meaning that the G generate data that trikes the D. In practice, equation one did not provide sufficient gradient for G to learn well so rather than training G to minimize log(1-D(G(z))) we train G to maximize log(D(G(z))), meaning we want to maximize the probability of D to classify G samples as real data. This objective function provides much stronger gradients early in learning. The training process dived into two parts – First, we sample m noise samples

$\{z^{(1)},..,z^{(m)}\}$ from noise prior p(z) and m examples $\{x^{(1)},..,x^{(m)}\}$ from the real data distribution, then we update the discriminator by ascending its stochastics gradient:

$$(2) \quad \nabla\theta_d \frac{1}{m}\sum_{i=1}^{m}\left[logD_{\theta_d}(x^{(i)}) + log\left(1 - D_{\theta_d}\left(G_{\theta_g}(z^{(i)})\right)\right)\right]$$

Seconds, we sample different m noise samples $\{z^{(1)},..,z^{(m)}\}$ from noise prior p(z) and update the generator by ascending its stochastic gradient:

$$(3) \quad \nabla\theta_g \frac{1}{m}\sum_{i=1}^{m}\left[log\left(D_{\theta_d}\left(G_{\theta_g}(z^{(i)})\right)\right)\right]$$

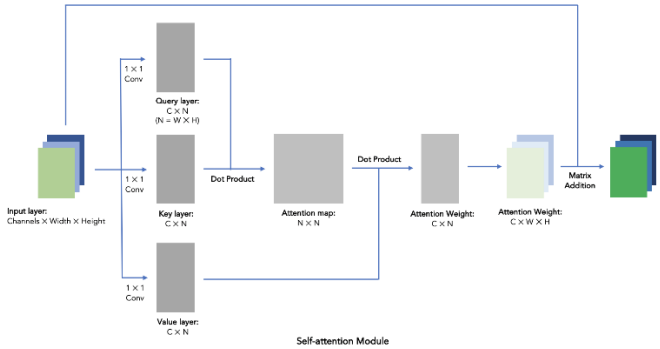
We repeat those two steps until the model converges.

2.2. Self Attention in GAN

First, we want to introduce the motivation using attention layers in GAN. GAN's models are good at classes with a lot of texture (landscape, sky) but perform much worse for structure. For example, GAN may render a dog's fur nicely but fail badly for the dog's legs. While convolutional filters are good at exploring spatial locality information, the receptive fields may not be large enough to cover larger structures. We can increase the filter size or the depth of the deep network, making GANs harder to train. Alternatively, we can apply the attention concept. For example, to refine the image quality of the eye region (the red dot on the left figure), SAGAN only uses the feature map region on the highlighted area in the middle figure. As shown below, this region has a larger receptive field, and the context is more focused and more relevant. The right figure shows another example on the mouth area (the green dot).

Fig 2: Example attention layers in CNN models

Implementing attention layers in GAN [6] will be as follow:



1. Using a kernel size one convo to generate Query, Key, and Value layers, with the shape of (Channels * N), where N = Width * Height.
2. Generate attention map by the matrix dot product of Query and Key, with the shape of (N * N). Pixels here mean data points in input matrices. The N * N attention map describes each pixel's attention score on every other pixel.
3. Get attention weights by the matrix dot product of Value and attention map, with the shape of (C * N). We then reshape the attention weights into (C * W * H). The attention weights describe each pixel's total attention score throughout all pixels.
4. Add the attention weights back onto the input layer with Gamma's weight, a learning parameter initializing at 0 means that the self-attention module does not do anything initially.

3. Experiments

This section will describe the experiment goals and research questions, the dataset used in our experiments, the evaluation metrics and motivation, and the experiment process.

3.1. Research questions

In our research, we will examine a few leading questions that derive from one another:

1. How to measure GAN performance for image generation – while in many fields, the loss functions are well described, measuring GAN performance is not easy. The difference between generating a good and bad quality image is very subtle and is combined by many features (ratios, color, position, etc.). The GAN's discriminator and generator loss functions can provide us with insights into the ability of the model to learn but provide us with little understanding of how good the image generated are (in a human perspective). FID was used to answer this question and will be further explained in the following sections. However, in a nutshell, this approach uses a trained CNN model to extract the strong feature from the real and generated images – and compare them. This process leads us to the next question.
2. As FID is based on feature extraction rather than spatial context, will the score be affected in cases where the spatial relationship is not maintained? This question provides another challenge – how can we create datasets that would

have different spatial noise? One way we can manage this issue is by leveraging the concept of attention. As stated previously in the article, using attention will help improve large structures in terms of locality and dimensions and improve the image's context. So, it is implied that removing attention would do the opposite. Using this assumption, we will remove the attention mechanism from different parts of the architecture and test for differences in the scores

3. The last remaining question is for a sanity check and intuition gaining - Does attention impact the generated images from a human perspective? Although not measurable (as implied from the human perspective), we needed to test the different results generated while removing attention. We first want to make sure that there are differences in the results as a check for the process development (the development was largely independent, so mistakes are possible). Second, if there are differences, what are they? How do they look? Are they crucial? We provided generated image comparisons for each N iteration to answer this question.

Fig 3: Will the score be affected in cases where the spatial relationship is not maintained? Example spatially inaccurate image

3.2. Dataset Description

The Stanford Dogs dataset contains images of 120 breeds of dogs from around the world. This dataset has been built using images and annotation contents the following:

- Number of categories: 120
- Number of images: 20,580
- Annotations: Class labels, Bounding boxes

3.3. Data Preprocessing

We will crop the images according to the annotations in the preprocessing phase. Since there are images that contain landscape objects other than dogs, such as humans and objects, the annotations give us information about the dog's location. Thus we can cut out the images to contain only dogs. These images will be resized to 64 * 64 pixels.

3.4. FID Score for Evaluation

The Frechet Inception Distance (FID) [7] is a metric that calculates the distance between feature vectors calculated for real and generated images. The score summarizes how similar the two groups are in terms of statistics on computer vision features of the raw images calculated using the inception v3 model used for image classification. Lower scores indicate the two groups of images are more similar or have more similar statistics, with a perfect score being 0.0, indicating that the two groups of images are identical. The FID score is used to evaluate the quality of images generated by generative adversarial networks, and lower scores have been shown to correlate well with higher-quality images.

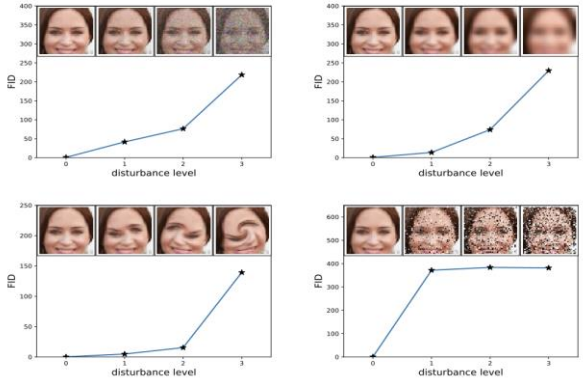


Fig 3: Example of how the increased distortion of an Image correlates with a high FID Score

The FID score is calculated by first loading a pre-trained Inception v3 model. The output layer of the model is removed, and the output is taken as the activations from the last pooling layer, a global spatial pooling layer. This output layer has 2,048 activations; therefore, each image is predicted as 2,048 activation features. This is called the coding vector or feature vector for the image. A 2,048 feature vector is then predicted for a collection of real images from the problem domain to reference how real images are represented. Feature vectors can then be calculated for synthetic images. The result will be two collections of 2,048 feature vectors for real and generated images. The FID score is then calculated using the following equation:

$$d^2((\boldsymbol{m}, \boldsymbol{C}), (\boldsymbol{m}_w, \boldsymbol{C}_w)) = \|\boldsymbol{m} - \boldsymbol{m}_w\|_2^2 + \mathrm{Tr}\left(\boldsymbol{C} + \boldsymbol{C}_w - 2\left(\boldsymbol{C}\boldsymbol{C}_w\right)^{1/2}\right)$$

The score is referred to as d^2, showing that it is a distance and has squared units. The and m_w are the covariance matrices for the real and generated feature vectors, often called sigma. The / and / refer to the feature-wise mean of the real and generated images, e.g., 2,048 element vectors where each element is the mean feature observed across the images. The $\left|m - m_w\right|^2$ refers to the sum squared difference between the two mean vectors. Tr refers to the trace linear algebra operation, e.g., the sum of the elements along the main diagonal of the square matrix.

4.1. Results and Analysis

4.1.1. Analysis step one – comparison of generated samples at 500 iterations

The first experiment tests the generated images after 500 iterations, while attention is turned off at different parts.

The differences are subtle but revealing.
With no attention to all parts of the architecture, the results focus on the image's center part. There is a lot of coloring generated (better than the other parts), and the is a diagonal structure of attempt to draw the dogs - with no particular "strategy".

Fig 4: Dogs images generation at 500 iterations – No attention

At attention in the generator is turned off, we expect the generator to have a disadvantage in the spatial context area. As a result, we see that the generated images are poorly formed, in a very unstructured way, as if the generator is trying to find its way to learn through an unstructured path.

Fig 5: Dogs images generation at 500 iterations – No attention to generator

With attention to the discriminator turned off, we expect the generator to have a spatial advantage. Indeed, we see that the generation is formed in a visual path, almost as the illustration of the attention mechanism suggests (Fig 2).

Fig 6: Dogs images generation at 500 iterations – No attention to the discriminator

When attention is turned on at all parts, we expect to get the best results, but when? The worst generated images, in comparison, are the result of this stage. The reason is that there is no advantage between the parts and more data to consider (as attention adds more context). Therefore, the model is expected to take more time to converge.

Fig 7: Dogs images generation at 500 iterations with full attention

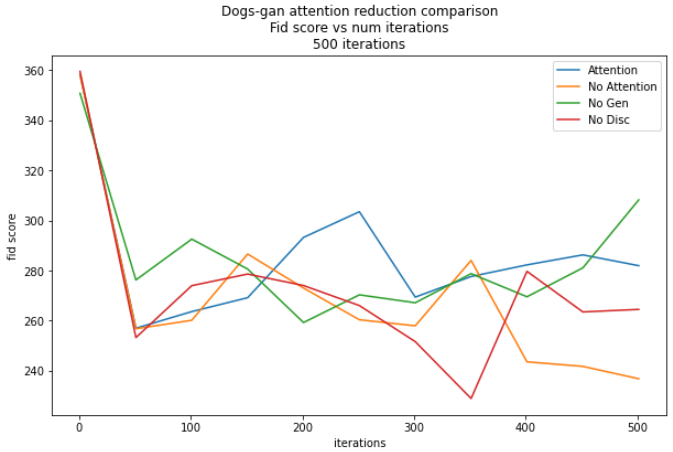


Fig 8: Dogs images generation at 500 iterations – FID score comparison

When comparing the FID score at this point, we do see that the full attention model is indeed getting the worst results, as visually detected.

4.1.2. Analysis step two – comparison of generated samples at 5000 iterations

From the human analysis, the best generating images are generated using full attention, and the worst results are generated from No attention, where the images are smeared and unstructured. However, the results are not dramatically different.

Fig 8: differences in results for 5000 iterations. From the top-down: Full attention, No attention, No generator, No discriminator. Images are smeared and unstructured, mostly in No attention.

The FID comparison shows pretty similar behavior, where the most significant difference is around 2500 iterations, and full attention achieves a much higher score for this point. However, at around 5000 iterations, the rest of the models close the gap, and the score is almost the same.

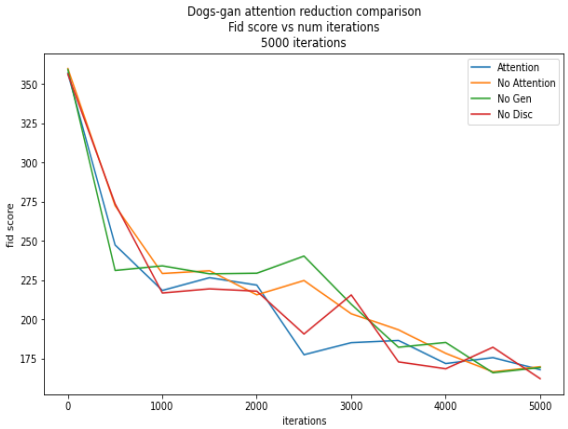


Fig 9: Dogs images generation at 5000 iterations – FID scores comparison. Full attention gains a higher score of around 2500 iterations before the gap closes.

4.1.3. Analysis step three – comparison of generated samples at 10000 iterations

At 10,000 iterations, the generated images produce better-defined dogs. There aren't significant differences in FID, and the behavior is roughly the same. However, one interesting difference separates the full attention model from the rest. The dataset contains 120 categories of different dogs. One common issue of generated images with gans is Mode collapse. This problem occurs when the generator learns to map several different input z values to the same output point. Partial mode collapse refers to scenarios in which the generator makes multiple images containing the same color or texture themes or images containing different views of the same dog [8]. In our case, the generator produced a small number of dog breeds, specifically in small sizes. The only model that could also produce bug breeds was attention one. This may indicate another advantage of the attention.

Fig 10: Dogs images generation at 10,000 iterations with no attention – only small bread dogs are generated

Fig 11: Dogs images generation at 10,000 iterations with full attention – different sized dogs breeds are generated

4.1.4. Analysis step four – comparison of generated samples at 20,000 iterations

At this point, the digs generated are mostly well-defined. The No attention model generates the worst result, as it produces extra features (double heads) and bad ratios. The FID score shows no significant difference.

Fig 12: differences in results for 20,000 iterations. From the top-down: Full attention, No attention, No generator, No discriminator. No attention generating double heads and bad feature ratios.

4.1.5. Analysis five – comparison of generated samples for full attention and loss score

The final analysis aims to answer whether the FID score can provide an insight of a good enough result to stop training. We compared the loss and FID score while checking the results every few thousands of iterations to check if the results are correlated with human analysis.

The FID score is at his lowest, around 15,000 iterations; we see almost no change there. However, the discriminator loss keeps improving until 24,000 iterations – indicating that the model keeps improving.

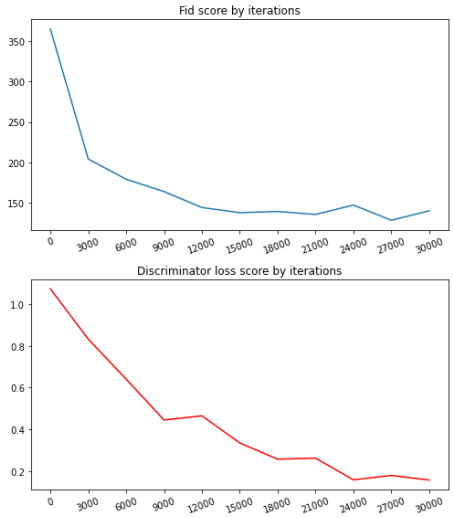


Fig 13: comparison of FID and loss discriminator scores

After 15,000, 24,000, and 30,000 iterations (fig 14). Each to understand the level of quality provided for the level. At 15,000 iterations, where the FID score indicated no more improvement, we see mostly well-defined dogs with some noise. After 24,000 iterations, where the loss stops improving, the dogs are better generated, and the model provides good results. At 30,000 iterations, there is no more improvement, and the generated images are even less in quality. This suggests that the discriminator loss provides a better indication of improvement than the FID score.

Fig 14: generated images after 15,000, 24,000, and 30,000 number of iterations – for full attention model

4. Conclusions

In this work, We developed the SA-GAN model to generate different dogs. Our main question revolved around the technique that will allow us to understand the quality of images generated by the model. We decided to use the FID score, a technique that uses the inception model to extract features from images. As FID is spatial insensitive (check the existence of features rather than their location), we wanted to examine whether changing the feature ratios will affect the score. We used activation of attention at different parts of the architecture to produce a variance in the spatial quality of the generated images. As part of our testings, we conclude using human result analysis that attention does positively affect the overall structure and localization of the dogs created. However, the FID score does not reflect this effect, meaning it is not a suggested metric for spatial ratio - this may be a result of an inability of the inception activations to capture localization features. This also indicates that the FID score is not an appropriate metric to check the quality of images generated by GANS. In terms of early stopping, we showed that the model kept generating better quality images even after the FID score stopped improving. We also found that the discriminator loss score provided a much better indication for improvement, as it indicated no more improvement after 24,000 iterations - a result aligned with our analysis. In conclusion, we suggest using both the metrics (FID and discriminator loss score) to gain insights into the generated images in the following way – at the first step, use the FID score to make sure the model is done creating all the relevant features (when it is done improving), following by the discriminator loss improvement as a measure of overall quality.

References

1. Doersch, C. (2016). Tutorial on variational autoencoders. arXiv preprint arXiv:1606.05908.
2. Van Den Oord, A., & Vinyals, O. (2017). Neural discrete representation learning. Advances in neural information processing systems, 30.
3. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., & Ganguli, S. (2015, June). Deep unsupervised learning using nonequilibrium thermodynamics. In International Conference on Machine Learning (pp. 2256-2265). PMLR.
4. Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. Advances in Neural Information Processing Systems, 33, 6840-6851
5. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. Advances in neural information processing systems, 27.
6. Zhang, H., Goodfellow, I., Metaxas, D. and Odena, A., 2019, May. Self-attention generative adversarial networks. In International conference on machine learning (pp. 7354-7363). PMLR.
7. Obukhov, A. and Krasnyanskiy, M., 2020, October. Quality Assessment Method for GAN Based on Modified Metrics Inception Score and Fréchet Inception Distance. In Proceedings of the Computational Methods in Systems and Software (pp. 102-114). Springer, Cham.
8. Ian Goodfellow (2017). NIPS 2016 Tutorial: Generative Adversarial Networks. arXiv:1701.00160v3.