CSE 676 Project 1: Generative Adversarial Networks

Shishir Suvarna
Graduate Student
Department of Computer Science and Engineering
University at Buffalo – The State University of New York
UB Person Number - 50290573
shishirs@buffalo.edu

Abstract

Generative Adversarial Networks (GANs) are being used to generate state of the art results by being able to generate realistic sample images over a variety of different classes. The objective of this project is to understand the working of GANs and to explore the latest developments like Self Attention mechanism

1 Introduction

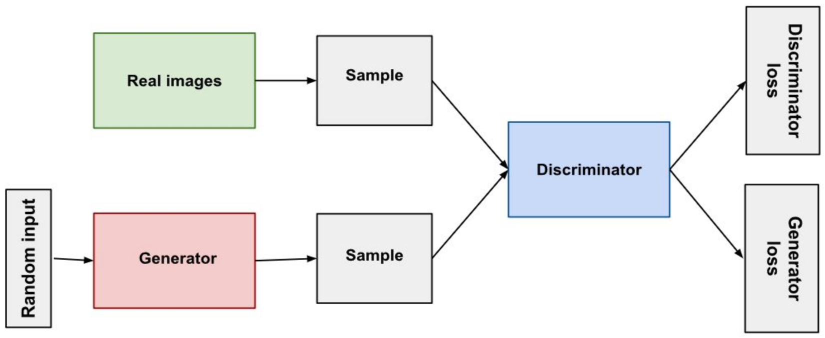
The field of Machine Learning and Deep Learning is advancing rapidly and one such breakthrough were GANs.They stand for Generative Adversarial Networks and were introduced in 2014 by eminent experts such as Ian Goodfello, Yoshua Bengio and Aaron Courville. They represent a new branch of Neural Networks which can have interesting applications including novel image generation, super resolution, domain transfer speech synthesis etc..

2 The GAN Model

As the name suggests, a GAN consists of two main components namely the Generator and the Discriminator. The generator is a network that produces output such as speech or images. The discriminator is a network that tries to distinguish whether a given input is a real image or whether it has been generated artificially.

Both the Generator and Discriminator are trained "adversarially" against each other in this Zero Sum form of training. As this training progresses, the discriminator becomes better at distinguishing between real and fake images and the generator becomes better at generating fake real looking images.

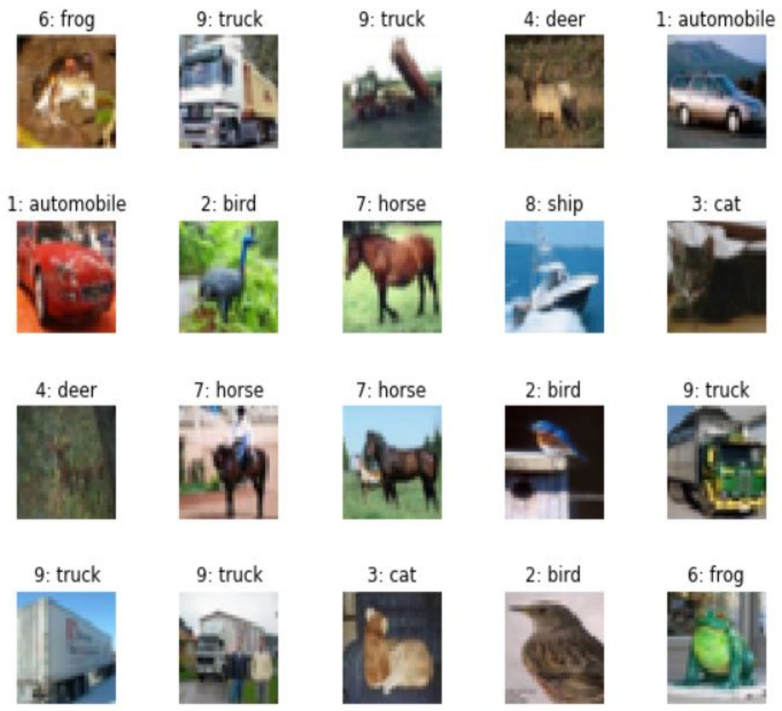
Training is said to be complete when the discriminator is not able to distinguish an image produced by a generator as fake.



Both the components are Neural Networks which are trained to get better at their tasks over time. The output of the Generator is directly connected to the input of the Discriminator. Based on the prediction of the discriminator, the weights of the generator are updated through back-propagation.

3 The Dataset (CIFAR10)

The dataset used in both our models is CIFAR10. This dataset consists of 60000 images divided into 5 batches for training and one batch for testing, each consisting of 10000 32x32 color images. These images belong to one of 10 classes namely airplane, cat, dog, frog, truck, automobile, bird, deer, horse, ship and bird. There are exactly 5000 images from each class in the training sets.



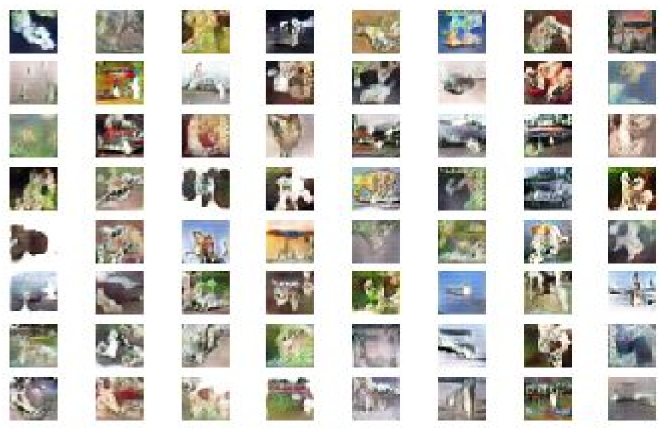
4 Deep Convolutional GAN

The first of the two models that we look at are Deep Convolutional GANs or DCGANS for short. In this model, the Generator model is a network which takes in input as noise from a latent space and produces an output image of 32x32x3 image. When it’s a part of a GAN, the output of the generator goes to the Discriminator.

The label for this output is set to be real. When the discriminator makes it’s prediction, the loss obtained is then used to update the Generator.

The Discriminator on the other hand, is a Convolutional Neural Net which uses downsampling of the input (unlike the generator which uses upsampling to increase dimensionality) to make a prediction. We must note that in order to prevent over-training of Discriminator, it isn’t traininable as part of the combined GAN

5 Output images of DCGAN



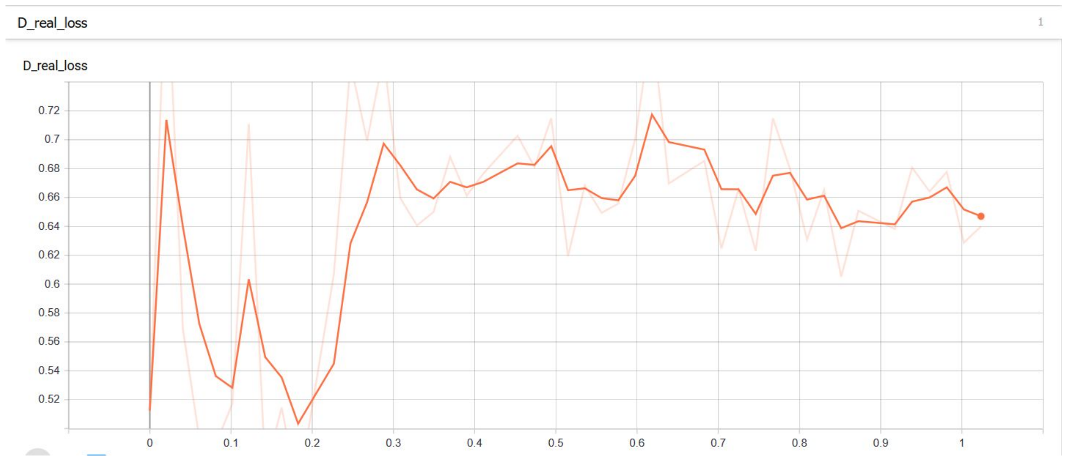
8x8 plot after 10 epochs

8x8 plot after 20 epochs

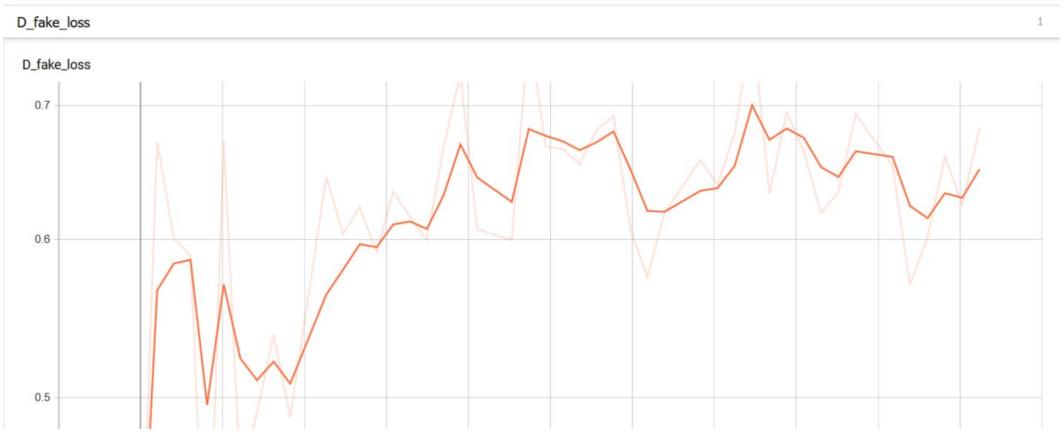
8x8 plot after 40 epochs

8x8 plot after 50 epochs

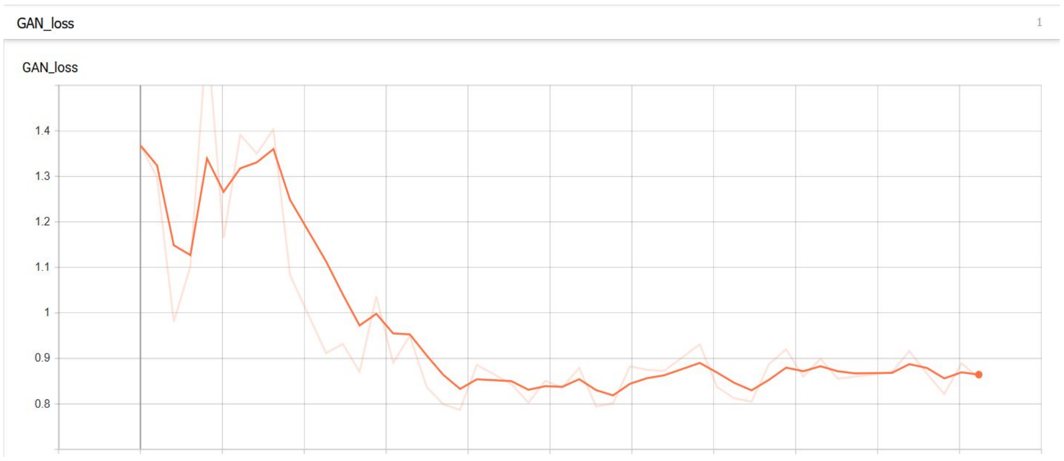
5 graphs for DCGAN using TensorBoard



real_loss after 50 epochs



fake_loss after 50 epochs



decreasing GAN loss

epoch 10:	FID 179.240	real_accuracy:76%	fake_accuracy:87%
epoch 20:	FID 134.985	real_accuracy:43%	fake_accuracy:85%
epoch 30:	FID 126.902	real_accuracy:52%	fake_accuracy:85%
epoch 40:	FID 104.808	real_accuracy:60%	fake_accuracy:77%

5 Limitations of the DCGAN

Even though DCGANS can produce new sample images of high resolution across many classes, they do so typically from spatially local points. For example, A DCGAN can be used to generate a high resolution image of a dog's face, but may not be able to generate a well defined set of legs. This is because it's a convolutional model which relies on the convolution operator. The convolution operator relies on a local receptive field.

6 Advantages of Self Attention GAN

DCGANs suffer another issue that could be limiting when dealing with multi class problems. They can succumb to mode collapse which basically means that the GAN produces a limited variety of images.

7 Self Attention GAN

Self Attention GANs mitigate the issues of normal DCGANs by introducing a non local model. This allows the generator and discriminator to model relations in an image which may be separated by a wide region. They help keep computational efficiency and have not just a local but a large receptive field. This helps establish relations between "long range dependencies".

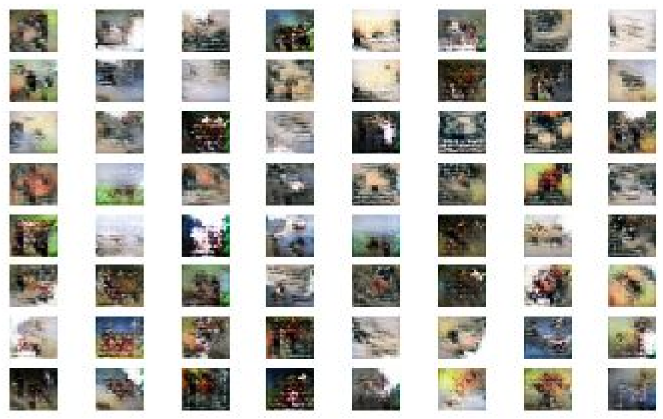
8 Attention mechanism

We deal with three vectors, our query, key and value(denoted by f, g,h in the implementation). The matrix multiplication of the query and key generate a third vector which decides how much of a value is actually passed on to the next layer. Post matrix multiplication, the output is passed through softmax(That's how we generate our probabilities).

In Self-Attention, the query, key and value are the same. In practice, these values are passed through a "convolution feature map". We then transpose the query and matrix-multiply it by the key and take the softmax on all the rows. Let's say the x was the image and o is the output, we multiply o by a parameter y. The final output O then becomes, O=y*0+x.

This whole process allows fine details to be captured even across a distance.

9 SAGAN Output



Output after 50 epochs

10 Conclusion

It is observed that SAGANs are better at recognizing long-term dependencies and hence generate more realistic images.

11 References

https://developers.google.com/machine-learning/gan/gan_structure
https://www.tensorflow.org/tutorials/generative/dcgan
https://machinelearningmastery.com/how-to-develop-a-conditional-generative-adversarial-network-from-scratch/
https://medium.com/towards-artificial-intelligence/techniques-in-self-attention-generative-adversarial-networks-22f735b22dfb
https://arxiv.org/abs/1805.08318
https://machinelearningmastery.com/how-to-implement-the-frechet-inception-distance-fid-from-scratch/
https://machinelearningmastery.com/how-to-develop-a-generative-adversarial-network-for-a-cifar-10-small-object-photographs-from-scratch/
https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks-Part-2/
https://sthalles.github.io/advanced_gans/
https://github.com/IShengFang/SpectralNormalizationKeras