Deep Learning Project 4

Devyani Lambhate

April 14, 2020

1 Deep Convolutional GAN

DC GAN is the most popular and simple model. In GAN we have two networks, the generator and the discriminator. The generator learns to generate better images and the discriminator learns to discriminate between real and fake images. It is an adversarial game between the generator and the discriminator. At nash equilibrium we expect the generator to generate images similar to the real training data.

In DC gan the generator first makes a 4x4 image, then 8x8, then 16x16 and at last 32x32 using the convolutional transpose operations. The discriminator also uses convolution layer that does the upsampling and finally classifies the image as real or fake

1.1 Conditional DC GAN

Convolutional GAN is a slight improvement in DC GAN. The aim here is to learn the conditional distribution p(x/y) where y are the classes and x are the images. I used an encoding layer to encode the class information. The architecture for Conditional GAN is here. Conditional GAN performs better than a vanilla GAN. Due to time constraint I did'nt trained a non conditional DC GAN completely.But I trained it for some iterations and it was clear that conditionals GANs are better.

Figure 1: Images generated by conditional Deep convolutional GAN

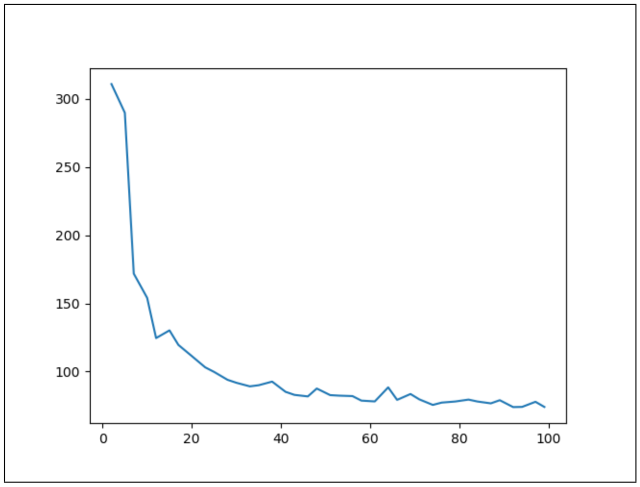

Figure 2: fID score vs 1000 iterations

1.2 Hyperparameters

For training DC GAN I tried different values of hyperparametes. For loss of discriminator and generator I tried RMSProp and Adam.In RMSProp afer some iterations the generator and discriminator was saturating without leaning anything. So I used Adam with lr= 0.0002 and beta1= 0.5

The number of batches for training was kept to be 64-128. More number of batches were not giving good results.

For embedding layer I used 50 neurons. As I found it sufficient to encode the class conditional information.

1.3 Spectral Normalization

Spectral normalization is a way of restricting the weights. This is done to stabilize the training of the discriminator. Spectral normalization is added in each convolutional layer.

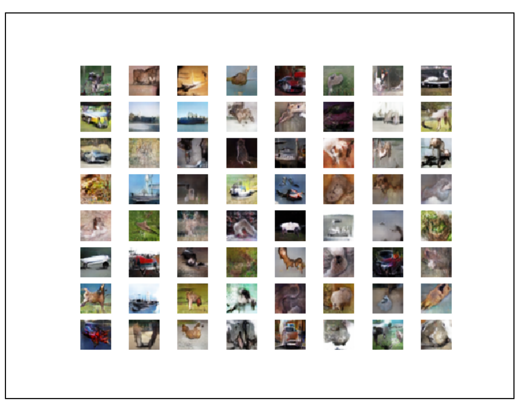

Figure 3: Images generated by spectral normalization

By comparing the FID score curve for simple DC GAN and DC GAN with spectral normlaization we can see that the FID curve is much smoother for the GAN with spectral normalzation.

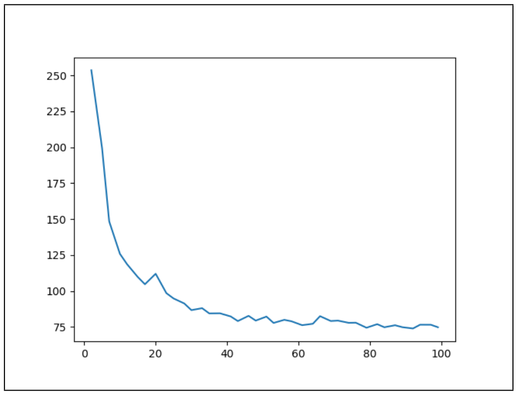

Figure 4: FID score of spectral normalization

2 Self Attention

DC GAN uses convolution operations in generating the images. DC GAN works good for features like sky, water where the texture is smooth. But it is not good at generating good geometry like it can generate a dog with more than four legs.

Figure 5: images generated by DC GAN

These are the images generated by simple DC GAN. As you can see in the second image the GAN very well copies the texture of the skin of an animal but it fails to generate proper geometry as in the second image the head of the animal is missing and in the first image there is a car with missing wheels. In Fact there is something like a leg attached to the car.

To better learn geometry of images we can increase the size of convolutional filters but doing so will make our GAN harder to train.

Therefore we use the attention mechanism in GAN. Attention mechanisms focus on particular parts of images to replicate the proper geometry of images.As you can see in the figure below that the images generated by Self Attention GANs have better geometry.

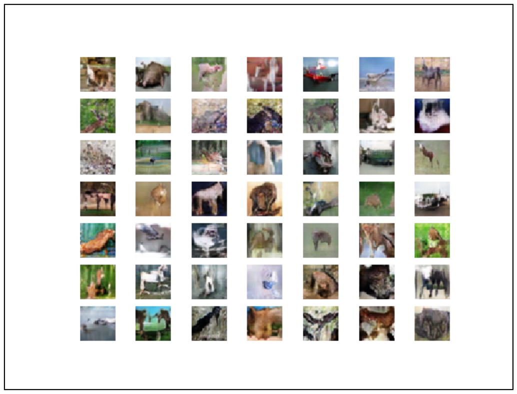

Figure 6: images generated by Self Attention GAN

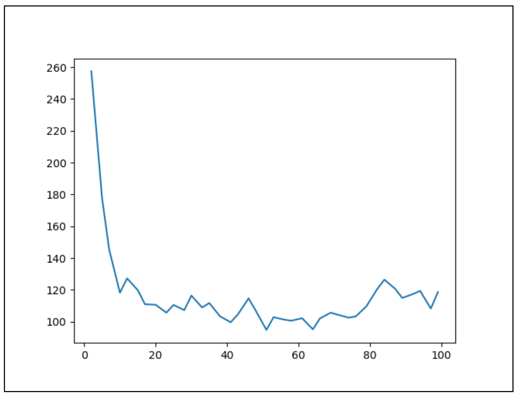

Figure 7: FID score per thousand iteration

3 Conclusion

I tried wasserstein loss function also but the loss was not converging due to some error in the code I guess. I will send the code with wassertein loss function once it is completed.

The images generated by DC GAN are good but adding spectral normalization gives better images.Images generated by Self attention GAN are very accurate in geometry.

Right now I am getting FID score in the range 70-75. By further training the GAN for more number of epochs the FID score was decreasing but due to time constraint, I was only able to train GAN till 100 epochs.

4 Refrences

https://machinelearningmastery.com/

https://github.com/IShengFang/SpectralNormalizationKeras

https://github.com/kiyohiro8/SelfAttentionGAN/blob/master/SelfAttentionLayer.py