Deep Learning CPSC - 8430
Homework -3 Report
By

Chaitanya Mandru
(cmandru@g.clemson.edu)

Training a discriminator/generator pair on CIFAR10 dataset utilizing techniques from DCGAN, Wasserstein GANs, ACGAN.

Introduction:

In this report the performance comparison between DCGAN, WGAN and ACGAN was compared on CIFAR-10 dataset.

GAN (Generative Adversarial Network) are algorithmic structures that utilization two organizations restricting against one another to produce manufactured new occasions of information that will be comparable like the genuine information. The generative organization makes new information while the discriminative organization will assess the information. Here, the generative organization adjusts to plan from the idle space to the information circulation of interest, on the other hand, the discriminative organization separates the information delivered by the generative organization from the genuine information dispersion.

The discriminator takes all true and false pictures and returns odds of 0 or 1 with 1. The discriminator is with the fundamental reality of the images in the feedback loop.

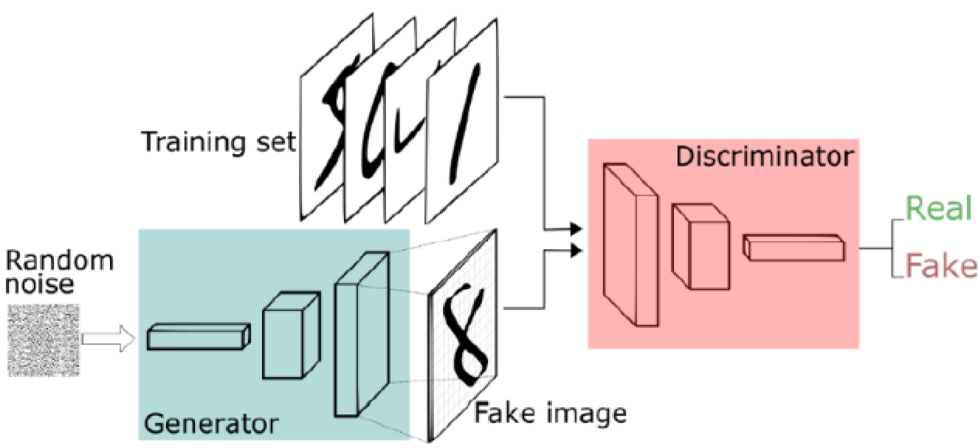

Figure 1: The process on images in the form of Generator and Discriminator

For example, a GAN trained on photographs can generate new photographs that look at least superficially authentic to human observers, having many realistic characteristics. However initially proposed as a structure of generative model for solo learning, GANs have likewise demonstrated valuable for semi-regulated learning, completely directed learning, and support learning.

Requirements:

- CIFAR-10 Dataset
- DCGAN Model
- WGAN model
- ACGAN model
- Python 3, PyTorch

About CIFAR-10 dataset:

The CIFAR-10 dataset comprises of 60000 32x32 variety pictures in 10 classes, with 6000 pictures for each class. There are 50000 training images and 10000 test images. The dataset is isolated into five preparation clusters and one test clump, each with 10000 pictures. The classes in the CIFAR-10 dataset are plane, vehicle, bird, feline, deer, canine, frog, horse, transport, truck.

DCGAN Model:

Deep Convolutional Generative Adversarial Network (DCGAN) composes of mainly convolution layers without max pooling and fully connected layers. It uses transposed convolution and convolutional stride for up sampling and down-sampling of the images. As DCGANs are hard to train, so there are architecture guidelines for stable training of the network.

- Use convolutional stride instead of max pooling
- For up-sampling use transposed convolution, eliminate fully connected layers,
- Use Batch normalization for all the layers except the input layer of the discriminator and the output layer for the generator
- Use Leaky-ReLU in the generator except for the output layer where tanh activation function is used and use Leaky-ReLU in the discriminator except for the output layer where sigmoid activation function is used.

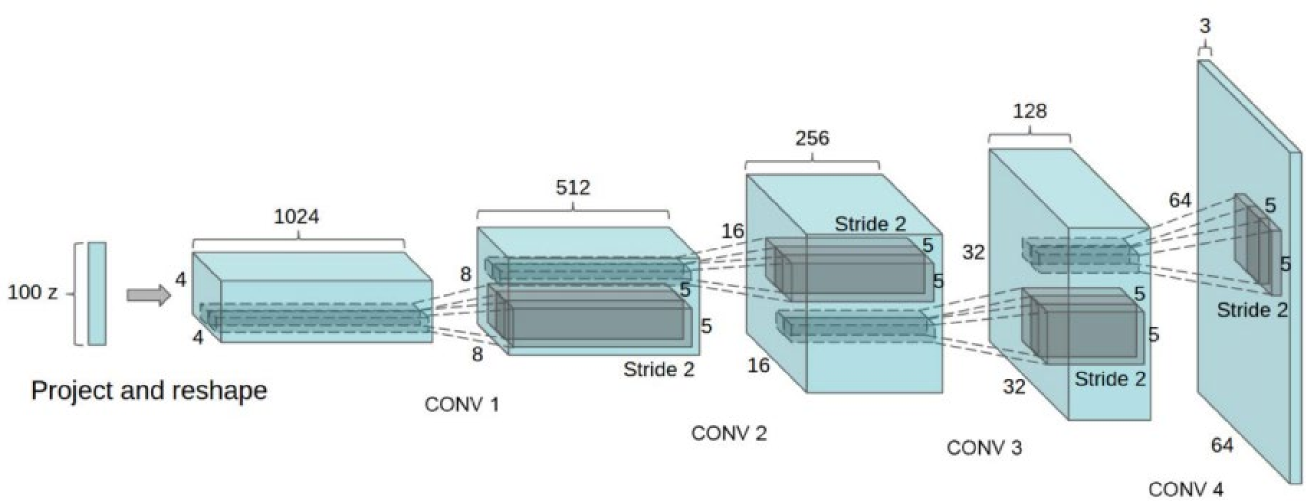

Figure -2: DCGAN generator working in a CNN

WGAN Model:

Wasserstein Generative Adversarial Network (WGAN) expands into the generative network, both improving reliability during model training and having a loss feature that is related to image quality. WGAN has a solid numerical inspiration, albeit a couple of minor changes to the norm, profound coevolutionary, generative rival organization or DCGAN expect practically speaking. Rather than sigmoid capacity, it utilizes the straight initiation highlight in the fundamental model's result sheet. It utilizes - 1 for real pictures and 1 for counterfeit pictures (rather than 1 and 0). Train the imperative and generator models with Wasserstein consumption. Limit imperative item weight for any little cluster move up to a little reach (for example [-0.01,0.01]). It additionally checks the fundamental model per emphasis more than the generator. It depends on the variation of RMSProp with low learning and no force (for instance 0.00005).

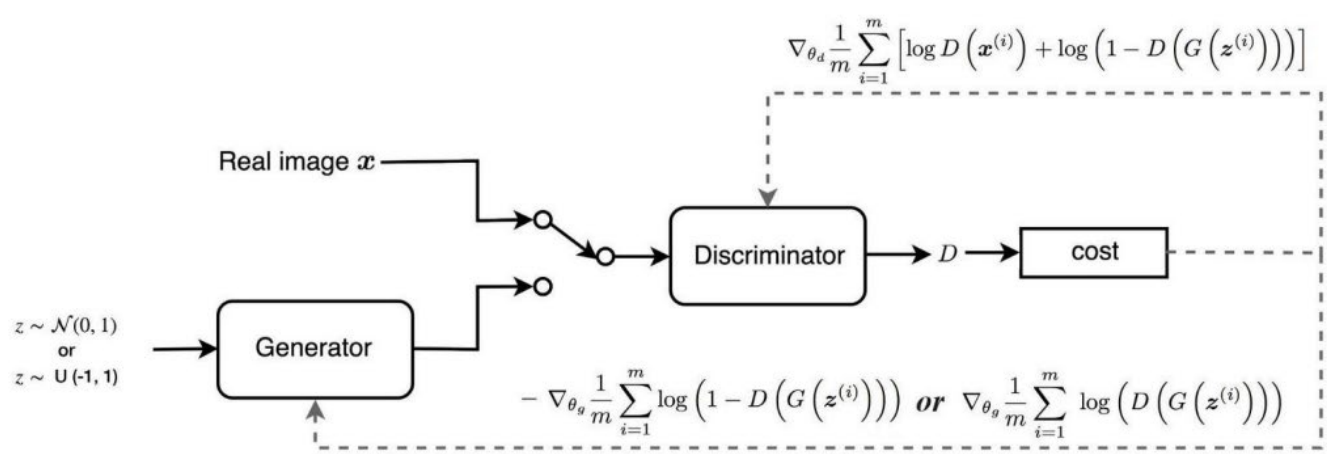

Figure -3: Working of WGAN with loss functions and Generator/Discriminator model

ACGAN model:

The ACGAN comprises of a generator and a discriminator, as any GAN network does. Notwithstanding, in the ACGAN, each produced test has a relating class mark c ~ C (available classes) notwithstanding the noise z. This class label assists the model with integrating the picture in view of the mark passed. The Generated pictures can be addressed as A translated convolutional layer is equivalent to a convolutional layer, yet with added cushioning to the first info. Therefore, when the convolution is applied with step 1 and no padding, the level and width of the result is more than that of the input. The translated convolutional layers of the generator is upheld with ReLU non-linearity.

The discriminator contains a bunch of 2d convolutional modules with broken ReLU non-linearity followed by direct layers and a softmax and sigmoid capacity for every one of its results — recognizing the class and the wellspring of the model. The loss function of the ACGAN is divided into two parts

- The log likelihood for the source being checked
- The log likelihood of the class being checked

As it is clear from the above loss function that the generator and the discriminator 'fight' about this loss function. The generator and the discriminator both attempt to amplify the class-loss. The source-loss is anyway a min-max issue. The generator attempts to limit the source-loss and fool the discriminator. The discriminator then again attempts to augment the source-misfortune and attempts to forestall the generator from acquiring an advantage.

Implementation:

In order to implement various networks and evaluate their performance on generating fake images based on CIFAR10 dataset, first we downloaded and extracted the CIFAR10 dataset using the download link provided in the project description.

Generator Model:

The generator worldview makes mistaken pictures with plausible blunders in a predetermined number of aspects. A square picture from the dormant area is utilized. The generator thought gives the idle space significance, and the inactive space is a smaller impression of the result space. This is finished by making a thick layer, like the principal secret layer, with the necessary hubs to mirror a low-goal picture. The following step is to convert the low-resolution image to a higher resolution image. It's used to quadric the field in the Conv2DTranspose row's input maps. Two more cycles are required to reach the 32x32 performance image.

Performance and Results:

DCGAN is mostly engaged with network plan upgrades, while WGAN is the misfortune include. The WGAN objective capacity can't upset the DCGAN design: everything brings down the normal disappointment of Wasserstein instead of the division of Jensen-Shannon with a particular organization engineering. The WGAN (and its branch-offs, like WGAN-GP) are compositionally skeptic. The main thing you'll need to stress over is utilizing the group standard; DCGAN suggests presenting to everything across, except those disorders with significant regularization figures.

Performance comparison between DCGAN, WGAN and ACGAN:

The below are the FID scores for all the models that tested on 50,000 CIFAR-10 samples trained for 500 iterations till 50 epochs.

Model	FID score
DCGAN	38.40562
WGAN	15.03176
ACGAN	13.74321

Table -1: Model comparison table

Below are the 10 best generated pictures for every model

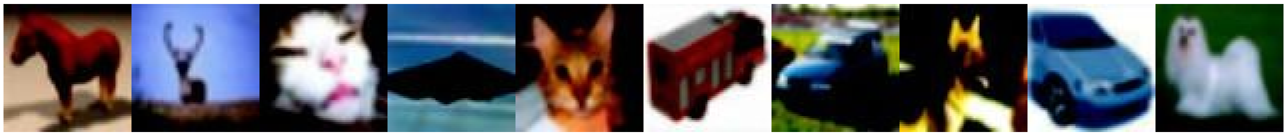

Figure- 4: 10 Best generated images by DCGAN model

Figure -5: 10 Best generated images by WGAN model

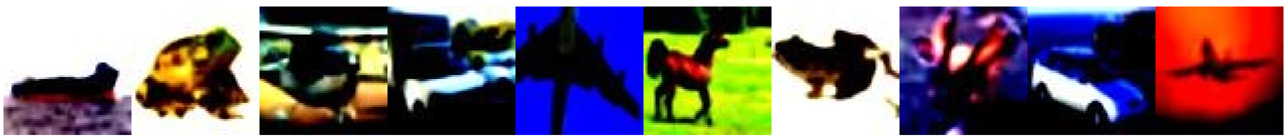

Figure - 6: 10 Best generated images by ACGAN Model

As we can see in the results the ACGAN model has the best PID score among all models.

GitHub Link: https://github.com/bittu426/CPSC-8430_HW_3