Auxiliary Classifier with BigGAN Model Architechture (AC-BigGAN)

Author : Wong Zhao Wu

Abstract

In attempt to generate realistic and diverse image across multiple categories of CIFAR10,I have make use of ACGAN as the primary framework with improvements from modern GAN network architecture like BIGGAN backbone with Conditional Batch-Norm and improvements in training techniques such as implementation of hinge loss, Orthogonal Initialization and label-smoothing for auxillary loss.

For evaluation, both the Inseption Score (IS) and Frechet Inception Distance (FID) are employed to provide quantifiable indication of the quality and diversity of image generated. I managed to achieve FID of 45.04(**the lower the better**) with IS of 7.19(**the higher the better**) with 80 epoch and 8 hours of GPU-training-time on Colab's P-100 GPU.

TABLE OF CONTENTS

Data Preparation

Preprocessing CIFAR -10 Pixel Values

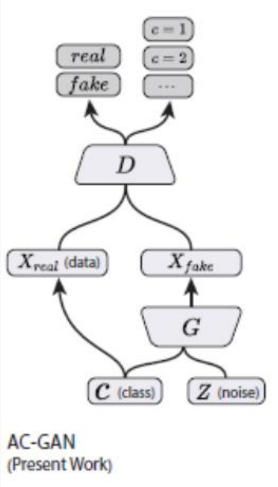
In the original CIFAR-10 dataset, the pixel range is between 0-1. However, since `tanh` activation is used for our generator which map the output range from -1 to 1, it is logical that similar normalization is needed for our real image such that it has similar data scale distribution to the generated image.

Concatenating Training and Testing Dataset

Since for GAN, there would not be a specific need of an independent testing-set for evaluation, I have concatenated both the training and testing set of CIFAR-10 which results to grand total of 60 thousands images across 10 classes with the classname defined as follows.

AC-BigGAN Model Architecture

AC-GAN



BigGAN Architechture

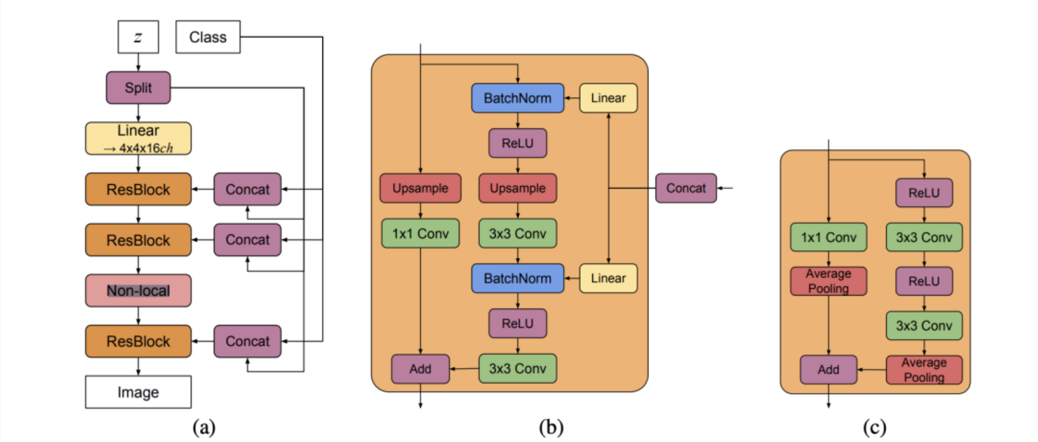


Figure 15: (a) A typical architectural layout for BigGAN's **G**; details are in the following tables. (b) A Residual Block (*ResBlock up*) in BigGAN's **G**. (c) A Residual Block (*ResBlock down*) in BigGAN's **D**.

Residual Blocks

Conditional Batch-Norm

Shared Embedding and Skipped Noise Vector

AC-BigGAN Model Architecture

Residual Blocks

- Increase the model capacity without suffering from increase difficulties in training a larger networks.
- Ensures that there is away for lower- level features and class information to be propagated from earlier layers to deeper layers.

The same residual connection is applied for both Generator and Discriminator with minor difference such as the use of average-pooling in discriminator for downsampling while the use of upsampling layer followed by convolution layer for upsampling.

Conditional Batch-Norm

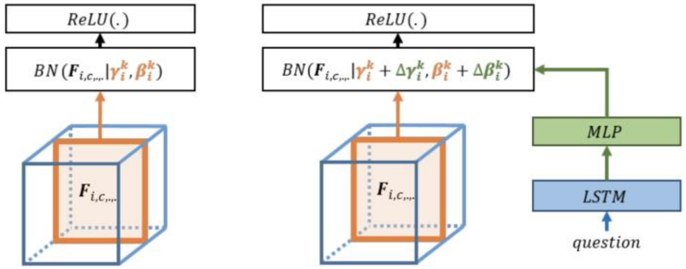


Figure 2: An overview of the computation graph of batch normalization (left) and conditional batch normalization (right). Best viewed in color.

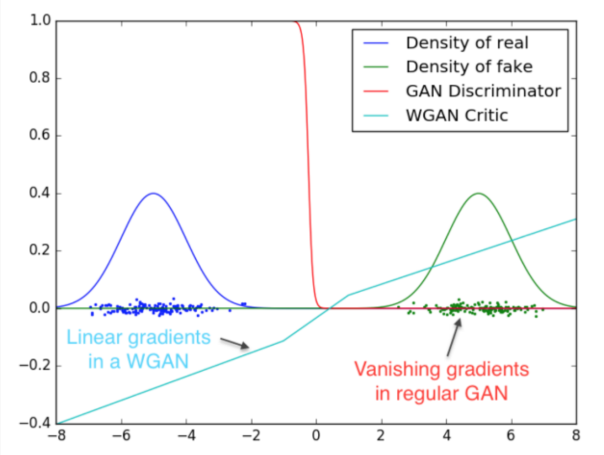
Shared Embedding and Skipped Noise Vector

- It is redundant to have independent embedding layer to transform the categorical label into categorical feature vectors that stores the semantic information of a class.
- This, however, could lead to higher chances of mode collapse as all of the cBN are essentially relying on a single interpretation of the categorical label.
- The skip connection for latent vector ensures that stochasticity are introduced not only to earlier layers where the general shape of the image and context of the image is formed but also to the later layers where more finer details are produced.

Failed Experiments

- 1. Differentiable Augmentation form [DiffAugment]()
- 2. Spectral Normalization from [SN-GAN]()
- 3. Data to Data Cross-Entropy from [ReACGAN]()

Training an AC-BigGAN



Categorical Cross Entropy with Label Smoothing for Categorical Loss

categorical loss is also introduced to motivates the model to produce image that is consistent with the input category.

To compare between the predicted class and actual class, categorical cross entropy is used with label smoothing applied for the actual label as a regularization techniques to make the discriminator loss confidence in the categorical prediction.

Hinge Loss for Adversarial Loss function

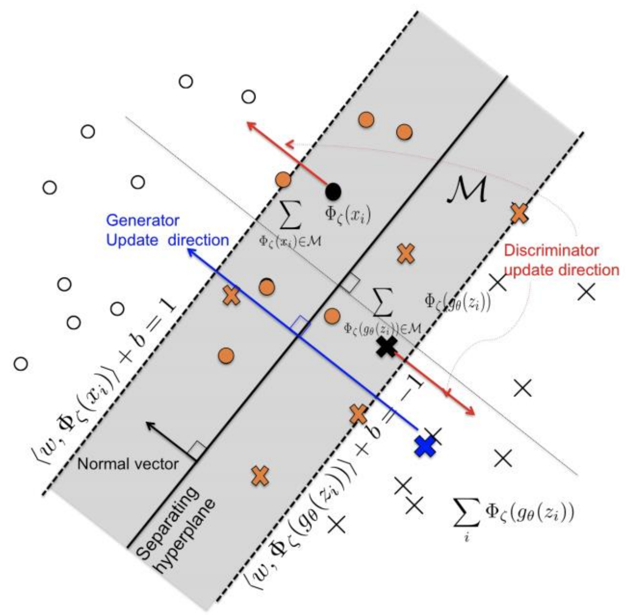
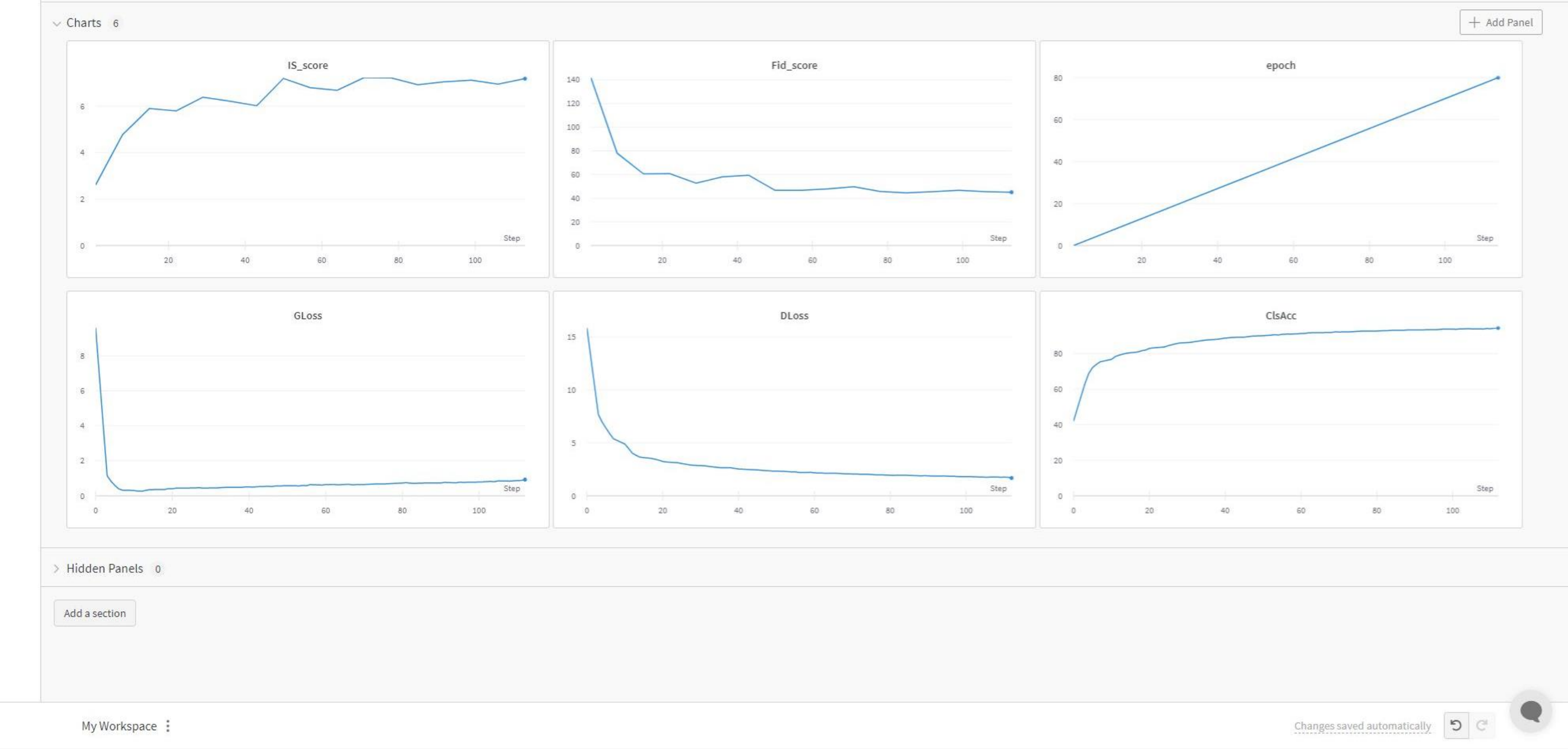


Figure 3: Geometric GAN using SVM hyperplane. Discriminator and generator update directions are shown.

Training an AC-BigGAN



Evaluation of GAN

Finally, when it comes to evaluating a GAN, common techniques like visualising the generated image is only helpful to a certain extend for small images especially so when our biological neural networks is struggling from guessing the content of the image from the real dataset.

Thus, common GAN evaluation metrics that with goal of generating high fidelity, distinct and diverse set of image is employed as a quantifiable statistics of the evaluation. Both IS and FID score will be using a pretrained Inception Network to provide us the feature statistics for computation of the metrics.

Eyepower Evaluation

Generate 1000 images for all 10 classes every 10 epoch,

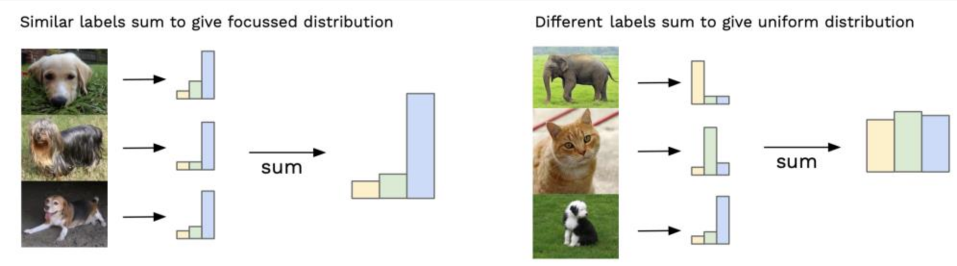
Plot grid of images every 5 epoch

Inseption Score (IS)

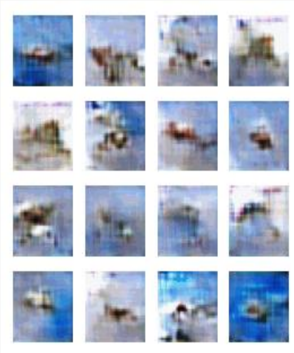
The goal of IS is to generate distinct image with high confidence from the pretrained InseptionV3 model across variety of image categories.

Frechet Inseption Distance (FID)

FID is another metrics available to calculate the distance of features maps generated from InseptionV3 model between the real and generated image and penalize the model if the images generated looks the same.

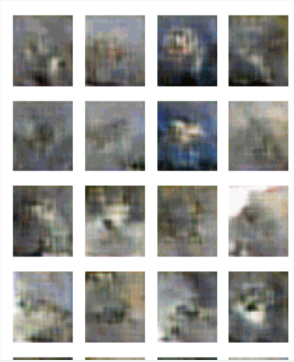


Evaluation of GAN

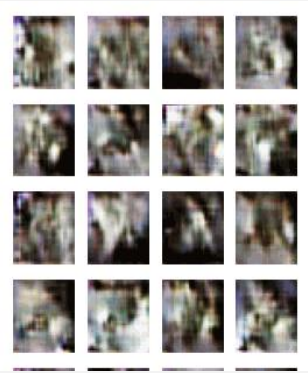


airplane

automobile



bird



cat

deer

dog

frog

horse

ship

truck

1. Generator has succeeded in understanding of content of an image given a specific labels

From all the gifs of generated images above, we can clearly observe that the generated images does represents the object that the labels it represents. (i.e. an automobile looks like a automobile, a ship looks like a ship etc.) This is a good sign as with just a training dataset, the generator is able to find out that it takes 4 legs and a head to form a horse just based on the information from the discriminator.

2. Subtle Mode-Collapsing Occured

By observing the gifs generated, at later epochs I have observed that although the image are significantly different from one and another in terms of pixel values, the specific kind of object that the image represents are very similar. For instance, when it comes to horse, at the start not all the horse are horse with the same breed. However as training progress, it seems that only a specific breed of horse have survived the natural selection with only the background changing from one horse to another.

This observation can be interpreted from two perspective. On the bright sight, it is consistent with the first observation that the generator has learned about the content of the image and which part of the image is more important than another an thus focuses on that while the background is just a result of the stochasticity introduced to the generator. On the other hand, this observation tell us that although by observing the FID-score, the model does not suffers from heavy mode collapse but subtle collapsing still occurs where the model keep generating the object of the same kind/breed/made.

3. GAN is having harder time Generating Image for Animals than Objects

Across all the classes, the model is doing a good job to generating image related to objects like Airplanes, Automobiles, Trucks, Ships and even Horse. However, when it comes to animals like Dogs, Cats and Frog, the quality of image generated aren't as high as the object available.

A quick explaination would be the complexity of task involved for animal images which can come from variety of angles and generally each animal breed looks significanly different than others. Another explaination could also be because of the "dirty" dataset whereby for instance, in bird category, there are mixes of chickens, ostriches, ducks which all are significantly different from one and another in terms of appearance. Thus, this might lead to inconsistent gradients which could be the explaination on why the model does badly on bird images.

Thank You

Personal Learning Reflection

Training a GAN is a great challenge due to the instability and the amount of time it consumes which limits the possibility of multiple experiments and fine-tuning in the context of an assignment.

Despite the improvement mentioned above, I have also attempted several model architecture techniques that is promised to improve the GAN's performance. However, due to lack of time and resource, I am unable to enjoy the performance boost as most of the training tends to collapse.

Nevertheless, the zero-sum training techniques is revolutionary, and I am excited to see how the same idea can be applied across multiple subfields of Deep Learning.