Machine Learning Project: Summary and Implementation
Conditional Image Synthesis with Auxiliary Classifier GANs

Hugo Da Costa, Adam Descarpentries, Tiago Fleury

1. Abstract

This document provides a summary of the paper by A. Odena, C. Olah and J. Shlens. on the subject of image synthesis. Their study aims at measuring the performances of the combination of conditional synthesis and auxiliary classifier GANs trained on the ImageNet dataset. These measures were performed using two new metrics of their making.

2. GANs

A generative adversarial network (GAN) consists of two neural networks trained in opposition. The generator G takes as input a random noise vector z and outputs an image $X_{fake} = G(z)$. The discriminator D receives as input either a training image or a synthesized image from the generator and outputs a probability distribution $P(S|X) = D(X)$ over possible image sources. The discriminator is trained to maximize the log-likelihood it assigns to the correct source:

$$L = E[logP(S = real|X_{real})] + E[logP(S = fake|X_{fake})] \quad (1)$$

The generator is trained to minimize the second term in Equation 1.

2.1. Conditional Image Synthesis

One way to improve the quality of image synthesis is to provide additional information to the training samples. Using the ImageNet dataset, the strategy is to supply the generator and the discriminator with the class label. Class conditional synthesis is known to improve the quality of generated samples.

2.2. AC-GANs

Auxiliary classifier GANs (AC-GANs) task the discriminator with reconstructing side information. Instead of feeding the class label to the discriminator, it is modified to contain an auxiliary decoder network tasked with outputing the class label.

The generator now uses both the noise and the label to produce samples and the discriminator outputs a probability distribution over sources and a probability distribution over the class labels. The objective function has two parts: the log-likelihood of the correct source, L_S, and the log-likelihood of the correct class, L_C. D is trained to maximize $L_S + L_C$ while G is trained to maximize $L_C - L_S$.

$$L_S = E[logP(S = real|X_{real})] + E[logP(S = fake|X_{fake})] \quad (2)$$

$$L_C = E[logP(C = c|X_{real})] + E[logP(C = c|X_{fake})] \quad (3)$$

To cover all 1000 classes from the ImageNet dataset, they made 100 AC-GANs, each one trained for 10 classes as to not crowd their models with tasks.

3. Resolution and Discriminability

To measure discriminability, they fed synthesized images to a pre-trained Inception network to compute the proportion of images that were correctly classified, they call this metric the Inception Accuracy. Discriminability supposedly measures the ability of the model to depict class-related features.

They trained two versions of their AC-GANs, one generating 128x128 images and a variant generating 64x64 images. They then created samples of all 1000 classes using both models. The samples were downsized to 64x64, 32x32 and 16x16, and also upsized to 128x128 and 256x256 using bilinear interpolation. Inception accuracy was computed for all samples and resized samples.

Results show a loss of 50% of accuracy when 128x128 are resized to 32x32, allegedly proving that samples are not "just naive resizings of low resolution samples". More importantly, results show that in all resolutions, the samples generated by the 128x128 model give a higher inception accuracy than samples of identical size generated by the 64x64 model, thus proving that high resolution AC-GANs provide better discriminability than lower resolution AC-GANs.

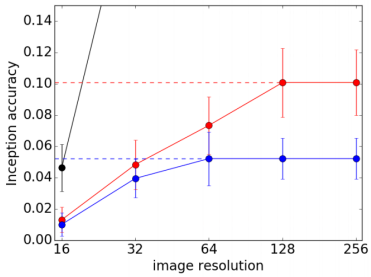

Figure 1. Inception accuracy for samples generated by a 128x128 architecture (red) and a 64x64 architecture (blue).

4. Measure of Diversity

One common problem with GANs is the "collapse" of the generator. This is when the generator finds a fault in the discriminator and always outputs the same image that fools it, resulting in a lack of diversity. To detect collapse, the authors suggest using the metric Multi-Scale Structural Similarity Index (MS-SSIM), usually used to measure the similarity between two images. From MS-SSIM they derived a method to gauge diversity among a set of images.

They computed the mean of the similarity over 100 random pairs of images of each class from the training set and from generated samples. Results show that for 847 classes the synthesized images show lower similarity, hence higher diversity, that the worst class of the ImageNet dataset. This proves that the majority of generated images present relevant results.

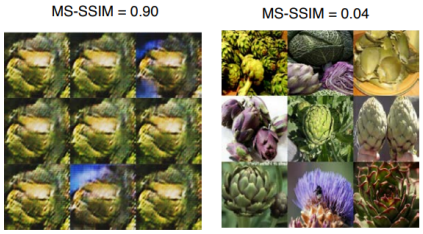

Figure 2. Example of the MS-SSIM score of the generated images from the artichoke class with little diversity (left) and of the ImageNet images from the same class with much better diversity (right).

4.1. Relation between Diversity and Discriminability

The authors find useful to emphasize on the relation between diversity and discriminability. They show that diversity is not obtained at the expense of discrminability by plotting classes according to their Inception Accuracy and MS-SSIM. In most cases, Inception Accuracy and MS-SSIM are anti-correlated, meaning that discriminabiliy and diversity are correlated.

4.2. Studying Overfitting

Another pitfall that must be avoided in GANs is overfitting. The network could simply memorize the training data and not really "generate" brand new samples. Three ways are proposed to verify whether or not a GAN is over-fitting.

1. *Nearest neighbors:* One can check the nearest neighbors of one generated sample in the training set and check the resemblance.

2. *Interpolation in the latent space:* For a chosen class, select two input noises z_1, z_2 and generate the corresponding images x_1, x_2. Then observe the transition from x_1 to x_2 by producing samples x_i from the interpolation points z_i. The transition must be smooth and each x_i sample must be meaningful. The type of interpolation is not specified in the paper.

3. *Exploit the structure of the model:* One can sample images from different classes but with the same input noise z. With AC-GANs, one can observe similarity in the composition of all images despite the fact that they all come from different classes. That proves that AC-GANs are able to "compose" the style of an image and that they are not just memorizing the training data.

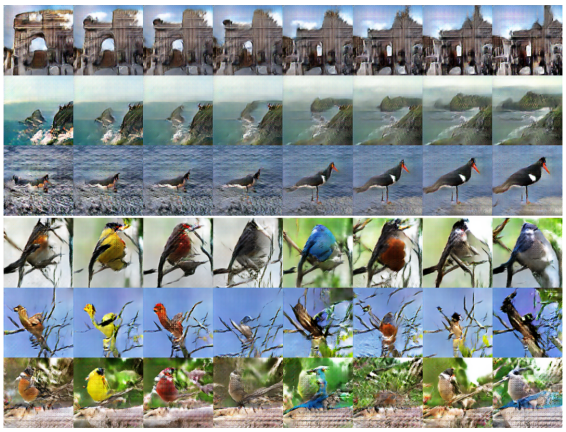

Figure 3. (Top) Interpolation in the latent space for 3 selected classes. (Bottom) Each column is a distinct bird class and each row is a distinct fixed latent code.

4.3. Benefits of Class Splitting

The last point of the paper is a highlight on the benefits of reducing the diversity of classes for training an AC-GAN. Indeed, it appears that training an AC-GAN on a splitted trainset containing only 10 classes yields more compelling samples with the full ImageNet set. However the authors could not conclude about how the choice of the classes affect the samples quality.

Implementation

5. Implementation

We tried to apply the concepts stated in the paper on datasets less ambitious than ImageNet because of our limited means and the impossibility to access the full ImageNet dataset. The following implementations were done using both Fashion MNIST and the CIFAR-10, as the paper gives an AC-GAN architecture for CIFAR-10 that we were able to adapt to Fashion MNIST.

Here are only displayed explaining and results on our CIFAR-10 application.

5.1. Architecture

The paper provides architectures for both Fashion MNIST and CIFAR-10. No experiments on the architectures were done as we did not want to diverge from their work. We copied them strictly and implemented them using Keras.

No template for AC-GANs for CIFAR-10 with lesser than 32x32 resolution was given. We tried training 16x16 and 24x24 models of our making but the results were inconclusive.

5.2. Training

We first started training our AC-GAN on epochs of a single batch composed of one half CIFAR images and the other half generated images. The training process was monitored using displayed generated samples every now and then. Samples started all gray before turning all black for hundreds of epochs. After a while, some color appeared without ever displaying logical shapes or patterns no matter the number of epochs. The size of the batch only delayed the apparition of colors. We chose a batch size of 1000 that we kept throughout the next steps of the training.

The authors referenced other papers, one of which gave us solutions to our training problem. One issue could have been the discriminator learning way faster than the generator, to solve this issue the trick was to add noise to the activation layers of the discriminator, rendering it less efficient.

The training was performed in 3 stages. At first, the addition of noisy activation helped shapes appear, without rogue colors this time. We then decreased the noise to decrease the static effect on the images, and in the 3 third stage we completely got rid of the noise to slowly transition out of it.

In figure 4, we can clearly see the affect of the noise on the samples. Further in the training, we managed to make it fade out while continuing learning. In the end, the synthesized images appear way clearer and the shape of the represented classes is recognized (for some).

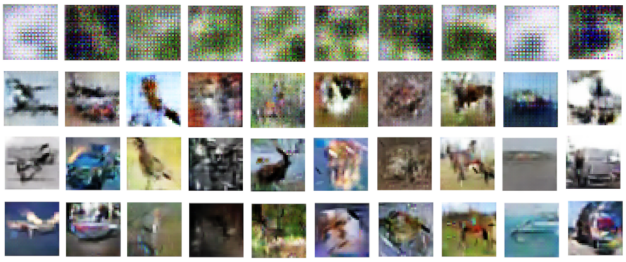

Figure 4. Generated samples of every class throughout different stages of the training.

6. Results Assessment

Once the model trained, we aimed at reproducing the experiments gauging our GAN quality. Discriminability and overfitting assessments were done but none giving a measured point of view of the diversity.

6.1. Inception Accuracy: CNN Accuracy

We tried reproducing the experiments seen in the paper using Inception Accuracy. It supposedly uses a pre-trained Inception network, but there was no mention on where to find it. Since none could be found on the internet for CIFAR-10, we designed a variant of the metric using CNNs.

In substitution to the Inception network, we trained 3 simple architecture of CNNs to classify the CIFAR dataset and 64x64 and 16x16 resized samples. Using basic training until convergence, we managed to get an accuracy of 0.73, 0.87 and 0.9 (respectively on the 16x16, 32x32 and 64x64 models). These networks were used to measure accuracy on the images generated by our GAN and their 16x16 and 64x64 resizings.

Since no other GAN architecture was trained, no comparison can be done. The goal here is to try to mimic the curves from figure 1. As seen in figure 5, results were non conclusive as we can see that the shapes are clearly different, the accuracy for our generated samples being so low. The probable cause for such results are the CNN themselves. being biased by the poor training only done on resizings of CIFAR-10 images.

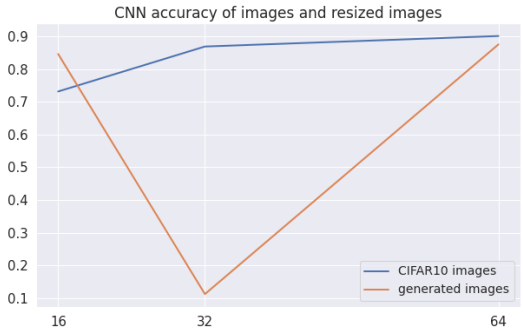

Figure 5. Plot of CNN Accuracy for 16x16, 32x32 and 64x64 images from CIFAR-10 (blue) and for 16x16, 32x32 and 64x64 images generated by our GAN (orange).

6.2. Overfitting: Nearest Neighbors, Interpolation, Style Application

To spot any kind of overfitting of our model, we tried implementing all 3 of the proposed methods: nearest neighbors, interpolation in the latent space and 'style' composition. Note that GANs are huge and complicated networks: just because an example shows no sign of overfitting doesn't mean there isn't any, as we will demonstrate.

6.2.1. Nearest Neighbors

The nearest neighbors method consists in generating an image and finding the most resembling samples in the training dataset. We generated multiple images from all classes and formed a set of well-generated samples, meaning that the object in the image is recognizable. Blurred samples are irrelevant as they can't be copies of existing samples from the dataset. Our results show that the tested samples are not copies of original images. We have yet to find examples of copying

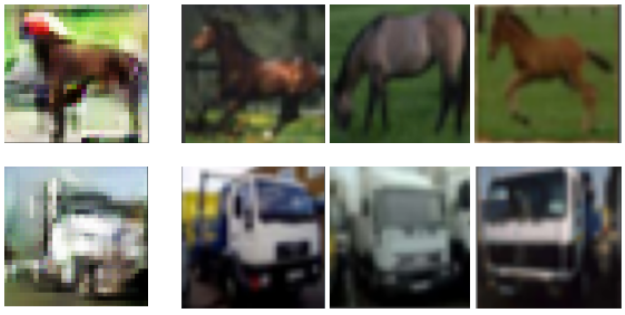

Figure 6. Two generated samples and their 3 closest neighbors. For those 2 examples, the closest neighbors are not copies signifying the model has not overfitted.

6.2.2. Interpolation in the Latent Space

We then tried reproducing the exploration of the latent space through linear interpolation. Once again we performed this experiment with a selection of good quality, relevant samples. For each pair of samples from the same class we derived 8 transitional samples by linear combination. We observed the two types of behaviours described in the paper. The expected behaviour which is a smooth transition with meaningful images and the undesired behaviour where the transition samples are not meaningful.

Figure 7. Transition from a generated sample to another through interpolation of in the noise input space. (Top) Expected behaviour. The transition is smooth. No overitting. (Bottom line) Undesired behaviour. The interpolated samples are not meaningful. That is a manifestation of overfitting.

6.2.3. Style Application

Finally, we tried replicating the impressive experiment of style application. Just like in the paper, we chose to compose style over classes that are the most similar in nature. The authors did it over different species of birds where we chose to do it over the CIFAR-10 *car* and *truck* classes. This experiment is limited in relevance as the classes of CIFAR-10 are not as meaningfully close as the example of the different bird species. That last experimentation did not yield clear conclusive results. Indeed, as we can see in figure 8 we observed *(car, truck)* pairs seeming to have similarities in colors or orientation as well as pairs without any resemblance in style.

Figure 8. Six *(car, truck)* pairs of sample. For a given pair, the same noise is used to generate both the car and the truck. (Left) One could possibly see some similarity in the shapes or the color. (Right) There is no similarity in style at all.

7. Conclusion

Conditional Image Synthesis with Auxiliary Classifier GANs is a very interesting paper about basic image synthesis and has been a great introduction to the field of GANs for us. After explaining GANs it has introduced a variant: AC-GANs. The goal was to prove the benefits of the variant through metrics of discriminability and diversity. Those metrics were introduced as the Inception Accuracy and the MS-SSIM Mean. While these methods seem relevant and the results were good, we can argue that said results are not very meaningful as the metrics are self imposed and not used by anyone else.

Implementing their work was a very enriching experience, quite satisfying as well. This subjects yields a lot of visual tests, which helps grasping the extent of our work. Even though not all of our experiments produced meaningful or satisfying results, we were pleased and happily surprised by the quality of our final model and of the generated images.

References

Odena, A., Olah, C., and Shlens, J. Conditional image synthesis with auxiliary classifier GANs.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. Improved techniques for training GANs.