Deformation of images using Generative Adversarial Networks: a study to improve their performance on neural activity data of the worm C. elegans

Roberto Ceraolo, Roberto Minini, Joel Rudsberg
Machine Learning CS-433, EPFL, Switzerland
Project hosted by Prof. Sahand Rahi, LPBS, Institute of Physics, EPFL

***Abstract*—LPBS at EPFL has designed a GAN intending to create realistic deformations of neuronal images of worms. The model is not functioning under some circumstances, and the goal of this paper is to find out when and why the GAN does not work and provide suggestions to fix the problems. Synthetic training data sets that portrayed specific characteristics of the worm images were created and used to train and test the GAN. Evaluation of the generated GAN images was made by manual inspection. These tests found that the GAN works with small deformations but breaks under large deformations; these include rotation, swirl with noise, dilation, and contraction. Increasing the number of data points was found to solve the issue for rotation, but the issue remained for swirl with noise. Dilation and contraction could not be tested with more data points due to how they were implemented. These findings suggest (i) increasing the number of worm images or (ii) preprocessing the training data to avoid large deformations could help solve the issues of the GAN.**

Introduction

The Laboratory of the Physics of Biological Systems (LPBS) at EPFL desires to track neurons of deforming brains of the worm C. elegans. Tracking is done by capturing a series of microscopic images of the worm while undergoing some deformation and analyzing how the neurons are affected in terms of change in position and shape. However, the images must first be annotated to identify the neurons, and only then can the images be compared to see the deformation. The issue is that only one data point is labeled, and manually annotating all images is not feasible since it is a manual, time-consuming, and, therefore, too expensive process.

To address this issue, LPBS has designed a generative adversarial network (GAN) that aims to take the annotated image and produce images of realistic deformations while preserving the annotations. With such a model, the generated images could be used to train a classifier that handles the image segmentation by annotating the real images; thus, solving the issue of tracking. However, the model cannot produce satisfactory results since it breaks under some scenarios.

Our project aims to produce insights into the GAN, identify when and why the GAN breaks, and suggest ways to resolve the identified issues.

Outline

The rest of the paper is organized as follows; first, section I describes the key characteristics of worm images and what realistic deformations are. Section II explains how GANs work and the variation used in this project. Then, section III demonstrates the issues of the current model. After that, the method used to find when and why the GAN breaks is described in section IV, which is followed by section V that shows the results of the experiments. Finally, a discussion and conclusion is given in section VI and VII, respectively.

I. Worm Image Characteristics

This section presents what the worm images look like and what it means for an image to be considered a realistic deformation.

A. The Data Set

Worm images consist of two major components: the cell bodies (somas) and a projection that originates from the cell body (neurites). To exemplify, consider figure 1 that shows three perspectives of the same worm image: the raw image as seen in the data set, an augmentation of it that changes the colormap to more clearly show the different parts, and the annotated neurons.

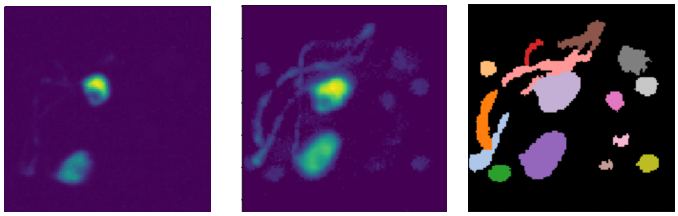

Fig. 1: The raw image, a contrast-enhanced version of it, and its annotation.

There are a couple of noteworthy observations to be made here. First, the round blobs are the somas, and the intricate lines which overlap are the neurites. Second, it is hard to see the individual neurons by only looking at the raw image. Third, all the 20 different parts shown in the annotated image are of interest.

The data set consists of 572 images, where one is annotated (the one shown above). The mission of the GAN is to create realistic deformations of this particular image (the left-most image). Once the deformation is found, it can also be applied to the annotation.

B. Realistic Deformations

A realistic deformation is characterized as

i focusing on the content (somas and neurites) and not the background;
ii not distorting the content (e.g., separation of somas into smaller parts, creating holes in the soma, or change in contrast);
iii somas and neurites are slightly rearranged or deformed;
iv unique, in the sense that it differs from other deformations (if all generated images are identical, it is called mode collapse) and from the image to deform;
v not only focusing on a subset of the components of the worm, for instance, only the two somas.

The performance of the GAN is evaluated by analyzing the produced outputs in respect to how well these characteristics are obeyed.

II. The Model

This section gives an overview of GANs and the specific variation used in this project.

A. Classical GANs

Generative Adversarial Networks is an unsupervised machine learning technique that involves discovering and learning the prominent patterns and regularities of data and then using that knowledge to generate new, never-before-seen data [1]. Two networks are trained in a zero-sum game to achieve this objective. One of the networks, called the discriminator, tries to differentiate between real and fake data points created by the second network, the generator, which goal is to fool the discriminator. The variance of the generator's output is possible by giving it random noise as input. If training proceeds successfully, the generator will convert the noise to new data points that model the real data distribution and its statistics.

B. Variation of GAN used

The variant of GAN used is a Deep Convolutional Generative Adversarial Network (DCGAN) [5]. The discriminator uses convolutional layers, and the discriminator strided convolutional layers. The strided convolutional layers allow the up-sampling to occur. Another noteworthy aspect of the model is the absence of fully connected and pooling layers.

A key difference of the GAN in this project is that it takes noise *and* the image to deform as input. As a result, the generator can learn a deformation that may be applied to both the image and the annotation. In the classical approach, new data instances are created but without any labels. Moreover, in classical GANs, the generator is fed only with noise and not the image to deform. In other words, it generates completely new data, and it does not deform one data point.

Several penalizing parameters have been introduced to guide the deformations and to achieve a balance between the generator and discriminator.

III. How the model breaks

To demonstrate how the model is not performing as intended, consider figure 2 which shows three outputs after 3000 epochs.

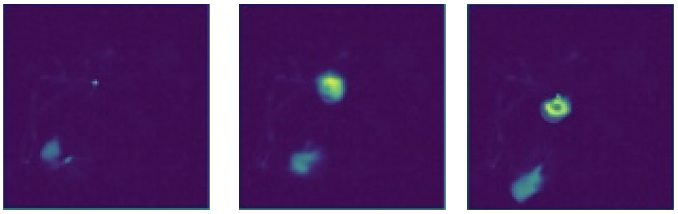

Fig. 2: Three example outputs from the GAN which does not satisfy the requirements of a realistic deformation.

The top and bottom somas have been completely and partially removed in the left image (violating requirement ii). The middle image is identical to the original image (violating requirement iv). The image to the right has created a distinct hole in the top soma (violating requirement ii). When analyzing the horizontal and vertical displacement of these images (not shown here), it is clear that the six parts to the right are ignored (these parts can more easily be seen in the annotated image of figure 1); thus, violating requirement v.

IV. Method

We created synthetic training data sets that resemble the worm images but only focused on a particular feature to precisely find when and why the GAN breaks. Starting from easier conditions, e.g., well-defined shapes, high contrast, no noise, only applying rotation, and gradually increasing the complexity after each condition was verified to satisfy the characteristics of the realistic deformations. The tests are grouped into two kinds depending on how the training data was generated: *de novo* and *non de novo*.

The de novo training set was independently generated de novo by drawing the ellipsoids in stochastically determined positions. In the non de novo case, the images were generated by applying a specific deformation (in various degrees) to the initial image. After having tested the de novo, we also considered the non de novo case because the generator produces outputs by deforming the original images, so deformation is the only tool the GAN has to produce good outputs. However, when feeding the GAN de novo images (produced by generation, not by deformation), the images might be too hard to reproduce by a deformation. Therefore, with the non de novo training sample, the GAN is fed images produced via a deformation since they were generated that way. In other words, it is easier for the GAN to learn to deform if the only difference between training datapoints is the actual deformation, and there is no stochastic generation, which can act as a confounder.

The test cases used the same hyperparameters as the original model and 3000 epochs. Each test starts with 500 data points to simulate the environment of the real data-set (except for erosion since it was not possible to achieve it). The number of data points was then increased to 5000 for rotation and swirl under particular circumstances.

The experiments were run on Izar, which is an Intel Xeon-Gold based cluster, available to the EPFL research community. More specifically, it was used with 1 node, 1 task, 1 gpu and 1 cpu, 4096MB of memory.

A. De Novo training data

Among the de novo test cases, the easiest one was first tested – only having the two ellipses representing the two somas. The library opencv [4] was used to create the two ellipses. We always draw two ellipsoids to model the somas. To draw them, we have to pass the following drawing parameters: center of the ellipsoid, sizes of the two axes, angle of rotation of the ellipsoid. A range of possible values that make the image realistic is found for each drawing parameter to add stochasticity and generate random synthetic images. For example, we made sure that one ellipsoid is always above the other, with no overlapping, and that their size is similar to the sizes of the somas in the real data. Then, when generating each image, we sample each drawing parameter from the respective range. Three example images can be seen in figure 3.

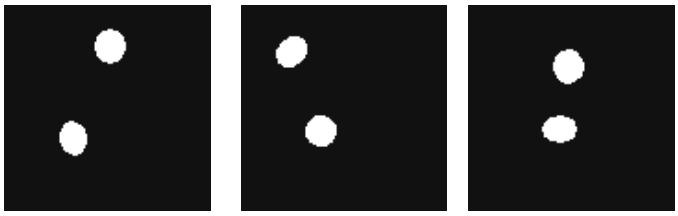

Fig. 3: Three random sets of ellipsoids generated by our code.

These are simple representations of the worm, and they leave out a critical part of the worm pictures: the neurites. To simulate the neurites, we decided to opt to use Bezier curves. The library Wand [3] was used to implement them.

In order to test a case that is even more difficult, we have also decided to add three different types of noises to our images: Gaussian Noise, Speckle Noise, and Poisson Noise.

B. Non De Novo training data

We tested four types of non de novo transformations: swirl, rotation, dilation, and erosion. A function from skimage [6] was used to generate swirl. Figure 4 shows two examples of this type of deformation.

Fig. 4: Examples of swirls. The original image is shown to the left and two example deformations following it.

A function from *skimage* was once again used to generate rotations. We produced rotations at various degrees thanks to the *angle* parameter of the skimage function. Specifically, the rotational values used were 20, 40, 100, 130, 160, and 180 degrees.

A kernel-based method was used to generate erosion and dilation. Similar to convolution, the kernel slides through the image. For erosion, a pixel in the original image (either 1 or 0, where 1 is non-background) will be considered 1 only if all the pixels under the kernel are 1. Otherwise, it is eroded (made to zero). This means that all the pixels near the boundaries of the blobs will be discarded depending upon the size of the kernel. Hence, the size of the blobs decreases. For dilation, it is the opposite. Here, a pixel element is 1 if at least one pixel under the kernel is 1. Effectively, this increases the size of the blobs. This method for dilation and erosion has a limitation. In general, each datapoint generated needs to be unique. The degrees of freedom in order to generate them here are: the size and the shape of the kernel. All the possible shapes and sizes of the kernel (without creating extreme transformations) were used and they are not able to produce as many datapoints as in the other experiments. In figure 5 we can see two examples of dilation and erosion.

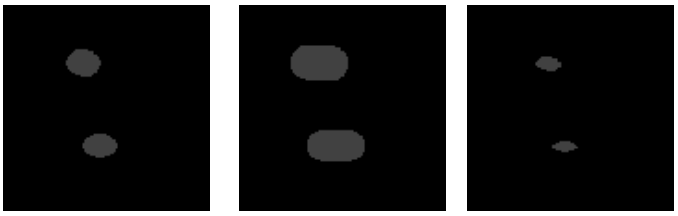

Fig. 5: The left image shows the original image, the middle a dilation, and the right an erosion.

V. Results

The GAN does not work for any experiment using the De Novo training data. For Non De Novo, when using 500 data points, the GAN works with small rotations and swirl without noise but breaks with more significant rotations (100 degrees or more), swirl with Gaussian noise, and dilation. It also breaks with erosion using 242 data points. Increasing the number of data points to 5000 solves the issues for rotation, but it remains for swirl with Gaussian noise. The results for Non De Novo are summarized in Table I.

A. De novo Results

Figure 6 shows the result from the first de novo test case of two stochastically generated ellipses.

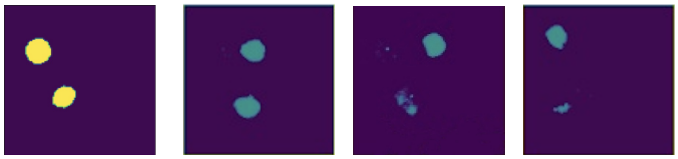

Fig. 6: The image to deform to the left and three outputs.

The first image, in which the somas are highlighted in yellow, is the image to deform and the following three are representative samples of the outputs. The following figures will have the same structure.

The first output (second image from the left) shows that the generated image is realistic. However, in many cases, the blobs are deleted, or the content is distorted, leading to unacceptable results. Similarly, the GAN did not work for the more challenging cases with neurites (which led to mode collapse) and with the addition of Gaussian, Speckle, and Poisson noise.

B. Non De Novo Results

All the results of the non de novo experiments can be seen in Table I. The non de novo experiments were started trying two different deformations: the swirl and rotation with a high degree span (15 to 180 degrees). In both cases, 500 data points was used. The GAN was able to learn the swirl deformation and generate unseen swirled images. However, rotation broke the GAN, and it experienced mode collapse. Several more experiments were therefore carried out in order to investigate further. Since it showed positive results, the swirl case was repeated with Gaussian noise, and the GAN proved not to be resilient to noise in this case. For rotation, three possible hypotheses were considered as plausible for causing the GAN not to work properly:

1) High degrees of rotation breaks the GAN, as they are easily recognizable by the discriminator.
2) Rotation must be performed through a deformation field (instead of an algorithmic solution), as the final transformation the GAN learns is through a deformation field.
3) Finally, another possible cause could be the lack of training data points.

Experiments were done with several different degree spans in the training data to verify or discredit the first two points. For 500 data points, it was observed that the GAN could deform the image when the range of degrees was kept under a certain threshold. Rotating with degrees spanning from 1 to 40 would allow the GAN to work, but results deteriorate when using 1-100 rotations or more. After seeing this, it was confirmed that the span of degrees is influential; thus, confirming hypothesis 1.

Moreover, we inferred that the non-deformation field transformation per se is not an issue for the GAN. We managed to train on smaller rotations (not reliant on deformation field), which did not break the GAN. If the deformation made without deformation field were to introduce artifacts, we would expect it to do so every time, not just with large rotations. Hence we concluded that not using deformation fields is not why the GAN breaks, thus confirming hypothesis 2.

Finally, the third hypothesis led to the primary discovery: the number of data points is a key factor influencing the success of the GAN. The initial rotation experiment, which broke the GAN with 500

Experiment	Noise	Number of Data Points	Results
Rotation 2 to 20, 1 to 40 degrees	-	500	Successful
Rotation 1 to 100, 130, and 160 degrees	-	500	Failed – content increasingly distorted
Rotation 15 to 180 degrees	-	500	Failed – mode collapse
Rotation 15 to 180 degrees	-	5000	Successful
Rotation 15 to 180 degrees	Gaussian	5000	Successful
Swirl	-	500	Successful
Swirl	Gaussian	500	Failed – mode collapse
Swirl	Gaussian	5000	Failed – mode collapse
Dilation	-	500	Failed – no deformation, only shifted
Erosion	-	242	Failed – no deformation, only shifted

TABLE I: Table shows all the *Non De Novo* results.

data points, was re-tried with 5000 datapoints, which caused the GAN to learn the deformation perfectly (as can be seen in figure 7). As a robustness double-check, the experiment was run with Gaussian noise added to the training data, and the GAN was again able to recognize the deformation and apply it. However, increasing the number of data points to 5000 did not solve the issue of swirl with noise (mode collapse persisted).

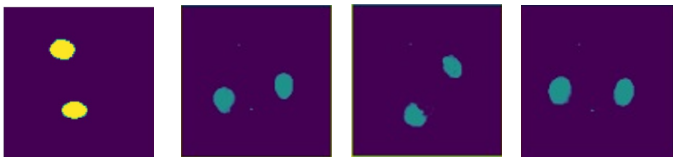

Fig. 7: The image to deform (left) and three generated images for rotation 15 to 180 degrees using 5000 data points.

Dilation and erosion broke the GAN for 500 and 242 data points. For these cases, the GAN was unable to learn the deformation. The generated images were not dilated/eroded but merely shifted.

VI. Discussion

The results show that the GAN breaks with de novo images. The GAN must learn the deformation per se and also deal with the difficulty arising from the fact that the images are produced independently and stochastically. This challenge may prove to be too hard. Moreover, this explains why the performances improve when fed with deformations of a single image. Therefore, future work should use non de novo images instead of de novo.

For what concerns dilation and contraction, it would be interesting to find out whether kernel methods, as presented here, introduce artifacts and whether creating more datapoints can help solve the issue.

Another suggestion of future work would include determining the threshold of data points necessary for good deformations, both for the simple synthetic images and the real worm images. Moreover, it would be interesting to address the following questions: does combining two or more deformations change the threshold? And if so, by how much?

Finding a metric that can describe the quality of the deformations produced by the GAN would be beneficial. We used visual inspection for the results; however, we believe that having an automatically computed metric could make large-scale experimentation faster and more trustworthy. For example, a metric that could help is the Fréchet inception distance [2].

Given our considerations about large deformations breaking the GAN, another possible fix could be to preprocess the training data to make them closer to each other. This could be done, for instance, by rotating the worm images to avoid large differences in the positioning of the somas.

Finally, a limitation of the study, which could be covered in future work, is the robustness of the results. Because of the stochastic nature of GANs, a single-run good result may not prove or disprove a thesis. To achieve reliable estimates, all experiments should be run several times and with several different starting training images. This was possible only for the main experiments, which showed significant variability across runs.

VII. Conclusion

The main finding of this project is that the GAN breaks when it is trying to learn too strong deformations, but increasing the number of data points helps it learn deformations even for more complex scenarios. This suggests a relationship between the difficulty of deformations and the number of data points. The GAN was also shown to handle noisy images if more data points were added. Moreover, de novo deformations were found not to be applicable to evaluate the GAN, as the latter is confounded by the stochastic generation of data.

References

[1] Ian J. Goodfellow et al. “Generative Adversarial Networks”. In: (2014). URL: https://arxiv.org/abs/1406.2661.

[2] Martin Heusel et al. “GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium”. In: (2017). URL: https://arxiv.org/abs/1706.08500.

[3] Hong Minhee. *Wand - Wand 0.6.7*. URL: https://docs.wand-py.org/en/0.6.7/.

[4] PyPI. *opencv-python*. URL: https://pypi.org/project/opencv-python/.

[5] Alec Radford, Luke Metz, and Soumith Chintala. “Unsupervised representation learning with deep convolutional generative adversarial networks”. In: (2015). URL: https://arxiv.org/pdf/1511.06434.pdf.

[6] Stéfan van der Walt. *skimage - skimage 0.19.0*. URL: https://scikit-image.org/docs/stable/.