Reading Report based on "Conditional Generative Adversarial Nets" by M. Mirza et al. and "Unpaired Image-to-Image Translation using Cycle Consistent Adversarial Networks" by J. Zhu et al.

Author: Golobokov Mikhail

0. Introduction:

What is GAN? We have 2 neural networks. The first one is generator, let's mark it as function $y_g=G(z)$. The second one is discriminator, we mark it as $y_d = D(x)$.Input is x, and output is y. The main idea is that G generate such images, so they look like they are real. The main task of D is to understand, that the input image is fake. If it's true, G will make better fake-image the next time. So, the other name for this type of neural networks is adversarial neural networks.

1. Conditional Adversarial Nets:

There are two methods to understand the world:

- To understand is to recognize
- To understand is to repeat

If we speak about first method, let's start with examples – here is a chair and we are sitting on it, here is an apple, we are eating it. So, by consistently observing the right number of chairs and apples, we learn to distinguish them from each other according to the instructions of the teacher, and thus discover some heterogeneity and structure in the world. The right word for this method is Discriminator. There main aim - assign the observed data the correct label, and they all answer the question "what does what I see look like?».

The second method is based on the following idea - if you watched the world for a while, and then were able to reconstruct part of it (fold a paper airplane, for example), then you understand something about how it works. The models, which make it are named as Generator. They don't needed in teachers.

Speaking about experimental results: we can use GAN's to generate new datasets as MNIST digits, for example. Absolutely incredible results are shown in this project: *Generating Faces with Torch* - http://torch.ch/blog/2015/11/13/gan.html

2. Cycle Consistent Adversarial Networks

The main problem of the image-to-image translation is that we usually have pairs of objects as input-output images pair. But what we should do if we don't have pairs? We should use Cycle Consistent Adversarial Networks. Their main

idea is we should use pairs of Discriminator and Generator: the one for S2T (source to target conversion) and another for T2S (target to source conversion).

Our two main objectives to control are: adversarial losses for matching the distribution of generated images to the data distribution in the target domain; and cycle consistency losses to prevent the learned mappings G and F from contradicting each other.

Adversarial Loss belong to both mapping functions (F and G). So, as I have already said, the main idea is that G generate such images, so they look like they are real. The main task of D is to understand, that the input image is fake. If it's true, G will make better fake-image the next time.

But there is a problem, that we should have some information about the output image: it might be cycle-consistent. Cycle Consistency Loss give us the guarantee that the new image has something inside like the same.

Examples:

Turning Fortnite into PUBG with Deep Learning (CycleGANs) - https://youtu.be/xkLtgwWxrec

Understanding and Implementing CycleGAN in TensorFlow - https://hardikbansal.github.io/CycleGANBlog/

3. How do we generate?

Let's speak about it using such example as monets.

The first variant is to use the Bayes ' theorem: we try to model coin tosses, and assume that they correspond to a binomial distribution. For really interesting objects like pictures or music, this doesn't work very well — the curse of dimensionality makes itself felt when there are too many features (and they depend on each other, so you have to build a joint distribution...).

The second variant is more abstract. The idea is to suppose that the thing we model can be represented as a small number of hidden variables or «factors». If we speak about our example, the figure on the bill can be decomposed into components in the form of denomination, serial number, beautiful pictures, inscriptions. This is how auto-encoders work.

The last variant suggested by Ian Goodfellow of Google:

a. We create 2 models, Generator(counterfeiter) and Discriminator(banker);
b. Counterfeiter tries to build on the output of a fake for real money, and the banker-to distinguish the fake from the original;

c. The aim of Counterfeiter is to make such coin, that the banker could not distinguish from the present;
d. The aim of Banker is to recognize the fake coin.

The article argues that the game eventually converges to the victory of the counterfeiter and, accordingly, the defeat of the banker.

4. Conclusion

GANs show incredible result. We can use them as graphic mods (the example with fortnite video), create new datasets. Also, some words about negative side of GAN:

- It’s harder to train face generator. Usually faces turn into an incomprehensible random mess.
- If we train colorful image or object, there three times more features.
- Sometimes we need to correct the behavior of our neural networks, for example, to freeze some constants and not allow to change them. It comes from the situation, when Generator or Discriminator is better than another one. So he can «see» more features, while another one not enough trained for it.

To sum up, Adversarial networks, cycle-GAN are awesome and give as opportunities to create new content and update older one.