Lab Pre-Report 14: Conditional Generative Adversarial Nets

Jiho Park, 2019142056
School of Electrical and Electronic Engineering, Yonsei University

Abstract

This paper is proposal of conditional Generatvive Adversarial Nets(GAN). Mehdi et al simply feeds the condition as input, both on generator and discriminator. Class-conditioned generation was experimented with MNIST digit dataset. Also, they've shown that this concept is also available on multi-modal tasks(e.g. image to tag generation).

1 Conditional Adversarial Nets

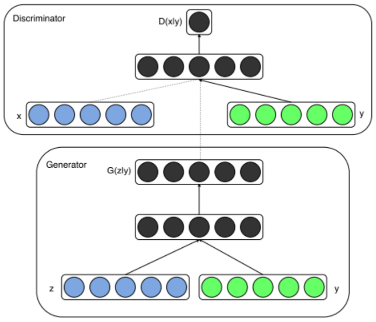


Figure 1: Conditional Adversarial Nets' Architecture

The main idea of conditioning on Generative Adversarial Nets(GAN) is to feed the condition y into both the generator and discriminator as additional input layer. y could be any additional information including the data from other modality. First in generator, the input noise $z \sim p_z(z)$ and y jointly forms hidden representation. In discriminator, data x and y are presents as inputs. This architecture is visualized in Fig 1. The objective function is as Eq 1.

$$\min_G \max_D V(D,G) = E_{x\sim p_{data}(x)}[D(x|y)] + E_{z\sim p_z(z)}[log(1-D(G(z|y)))] \quad (1)$$

2 Experiments

2.1 Unimodal: Class-conditioned Image Generation

Mehdi et al [1] trained their conditional GAN on MNIST data with one-hot class conditions.
Generator: Dimensionality of 100 with uniform distribution was used for the noise z. z and y are mapped with layer sizes of 200 and 1000 each, before concatenated to 1200 dimension. ReLU was applied for each layers, and sigmoid unit layer was used for final output, which is 784-dimensional generation.
Discriminator: Maxout layers map the each input in discriminator. For the data x, it is mapped to 240 units and 5 pieces of maxout layer. And for the condition y, it is mapped to 50 units and 5 pieces of maxout layer. Then, both layers are jointly mapped to another maxout layer with 240 units and 4 pieces. Finally, the sigmoid layer outputs the probability of the data with condition.

2.1.1 Results of Unimodal Experiment

As a evaluation result, they found out that their parzen window-based likelihood score wasn't good enough as other method, but was generating the class-conditioned data(Fig 2). Mehdi et al [1] have emphasized that this results is the proof-of-concept.

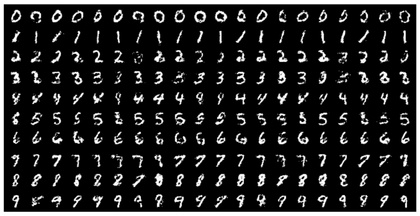


Figure 2: Class-Conditioned MNIST Digit Generation

2.2 Multimodal: Image-conditioned Tag Generation

The task of the second experiment is generation of tag-vectors from images. For data, they used MIR Flickr 25,000 dataset. And for feature extraction, the convolutional model pre-trained on ImageNet and the skip-gram model trained with YFCC100M were used.
Generator: Like in previous experiment, dimensionality of 100 was used for the noise, and mapped to 500 dimension ReLU layer. For 4096 dimension image(condition) input, it is mapped to 2000 dimension ReLU layer. Then finally, two layers are jointly mapped to 200 dimension word-vector.
Discriminator: Discriminator uses 500 and 1200 dimensions for hidden ReLU layer of each tag and image inputs. Then two layers are jointly mapped to 1000 units and 3 pieces of maxout layer. Finally, this layer is mapped to single sigmoid unit.

2.2.1 Results of Multimodal Experiment

For evaluation, they generated 100 samples for each image and find top 20 closest words using cosine similarity [1]. The qualitative results is shown in Fig 3. As the table shows, their tag is not accurate enough. However, Mehdi et al [1] said that their result demonstrates the potential of multimodal utility, and emphasized their conditiong idea.

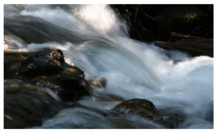
	User tags + annotations	Generated tags
	montanha, trem, inverno, frio, people, male, plant life, tree, structures, transport, car	taxi, passenger, line, transportation, railway station, passengers, railways, signals, rail, rails
	food, raspberry, delicious, homemade	chicken, fattening, cooked, peanut, cream, cookie, house made, bread, biscuit, bakes
	water, river	creek, lake, along, near, river, rocky, treeline, valley, woods, waters
	people, portrait, female, baby, indoor	love, people, posing, girl, young, strangers, pretty, women, happy, life

Figure 3: Image-conditioned Tag Generation Result

Reference

[1] Osindero S Mirza M. Conditional generative adversarial nets.

Lab14 Pre-Report:
Image-to-Image Translation with Conditional Adversarial Networks

Jiho Park, 2019142056
School of Electrical and Electronic Engineering, Yonsei University

Abstract

Pix2pix is conditional adversarial nets for image-to-image translation problems. Phillip et al [2] reveals that they are not the first to propose their idea. However this paper is focused on the investigation of conditional adversarial nets for image-to-image translation tasks. They demonstrates that this approach has wide applicability(e.g. colorizing images, synthesizing photos from label maps, reconstructing objects from edge maps). Also, they presents the simple framework for both good results and analysis. *Since there are not enough spaces, no figures and tables are attached on this report. You may check those on [2].*

1 Method

1.1 Objective Function

The objective function of the conditional Generative Adversarial Nets(GAN) is as Eq 1, where x is the input image(which is the condition), y is the ground truth of the translated image, and z is the input prior noise of the GAN. Generator G minimizes the equation and discriminator D maximizes the equation. (Phillip et al [2] also tested the loss with unconditional discriminator to examine the importance of conditioning the discriminator.)

$$L_{cGAN}(G,D) = E_{x,y}[D(x,y)] + E_{x,z}[log(1-D(x,G(x,z)))] \tag{1}$$

The researches before this, had revealed the benefit of the mix of traditional loss(e.g. L2 loss). In this case, L1 loss is used and the effect of it was analyzed and experimented. As a result, the final objective function is as Eq 3.

$$L_{L1}(G) = E_{x,y,z}[||y-G(x,z)||_1] \tag{2}$$

$$G^* = \arg\min_G \max_D L_{cGAN}(G,D) + \lambda L_{L1}(G) \tag{3}$$

They also investigated the effect of the noise. Without z, the network will be deterministic and won't produce the distribution. However, they've empirically found out that the generator learns to ignore the noise. So, they provided the noise by "dropout". But still, they've found out that only minor stochasticity remained and leaved it as an open problem.

1.2 Network Architecture

Both generator and discriminator uses the module of convolution-BatchNorm-ReLu form.

1.2.1 Generator: U-Net

The generator should be the mapping function between high resolution input to high resolution output. Therefore, U-Net architecture is used, which has the concatenate skip connection between layer i and layer $n-i$.

1.2.2 Discriminator: PatchGAN

In many cases, L1 loss mainly captures low frequencies. This motivates restricting the discriminator to only model high frequency structure [2]. Therefore, PatchGAN discriminator is used, which captures the probability of the local image patches. PatchGAN discriminator classifies each $N \times N$ patch and then averages all responses.

1.3 Optimization and Inference

As in [1], they also trained G to maximize $\log D(x,G(x,z))$, instead of minimizing $\log(1-D(x,(G(x,z))))$. In addition, they slowed down the D to learn, by dividing the objective by 2. At inference, the dropout and batchnorm was applied identically as in training. When the batch size is 1, this approach is called "instance normalization" and has shown effectiveness on image generation.

2 Experiments

2.1 Evaluation Methods

Evaluation of the generative model's output is challenging task, so two evaluation methods were adopted. First is the human based perceptual study, 'Amazon Mechanical Turk(AMT)'. Second is the "FCN-score", which is using the classification accuracy of the FCN-8s model.

2.2 Analysis of the Objective Function

The effect of the each loss term was investigated. L1 loss alone led to blurry results. cGAN loss alone ($\lambda = 0$ in Eq 3) led to sharper results but with visual artifacts, and using both terms reduced this effect. Additionally, loss term removing the conditioning of discriminator were also tested, and have shown realistic but mismatching results. In aspect of color, unlike cGAN loss, L1 loss led to narrower distribution, encouraging more grayish colors. As a results, they found out that L1+GAN and L1+cGAN performed best.

2.3 Analysis of the Patch Size N

The discriminator receptive field sizes of $1\times 1, 16\times 16, 70\times 70, 286\times 286$ (full) were tested. As expected, the result with 70×70 PatchGAN performed best. This fixed-size patch discriminator has advantage of that it can be applied to arbitrary input size. The test of 512×512 maptranslation with 256×256 trained network, shows how effective it is.

2.4 Perceptual Evaluation

For the result of the AMT experiment, mentioned on Sec 2.1, their method fooled participants 18.9% on map to photo task and 6.1% on photo to map task. It outperformed the L1 loss network especially on map to photo task. In colorization evaluation, their model fooled 22.5% which outperformed L2 regression loss network, but lost to Zhang et al's network.

2.5 Semantic Segmentation

As a results of the loss comparison on task of semantic segmentation, L1 loss outperformed other losses(e.g. cGAN, L1+cGAN). Phillip et al [2] argued that vision problems has less ambiguous goals than graphics tasks. And due to this, reconstruction loss like L1 performs better on those tasks.

3 Conclusion

Empirically, we can confirm that conditional adversarial networks is an general and sufficient approach for image-to-image translation tasks. Also, it learns a task-adapted loss which makes it applicable in a wide variety.

Reference

[1] Pouget-Abadie J. Mirza M. Xu B. Warde-Farley D. Ozair S. Courville A. Bengio Y Goodfellow, I. J. Generative adversarial nets. *NIPS*.

[2] T. Zhou P. Isola, J.-Y. Zhu and A. A. Efros. Image-to-image translation with conditional adversarial networks. *CVPR*.