DEEP LEARNING LAB(EEE4423-01) Week 14 - Conditional Generative Adversarial Nets, Image-to-Image Translation with Conditional Adversarial Networks

Hyeon Si Eun[1], 2019142214
[1] Department of Electric and Electronic Engineering, Yonsei University

Conditional Generative Adversarial Nets

"Conditional Generative Adversarial Nets"[2] introduces a conditional version of Generative Adversarial Nets (GANs) that includes additional conditions. Traditional GANs cannot generate data with specific conditions because they learn from data distributions. In this paper, we incorporate labeling (additional information) into the original GANs to create a model that we can control as desired. By adding conditions to the Generator and Discriminator, we demonstrate that this model can generate MNIST numbers conditioned on class labels. We also explain how to train a multi-modal model using this model. Furthermore, we provide an example of image tagging by generating descriptive tags through this model.

The advantages of GANs can be summarized as follows: they eliminate the need for Markov chains, secure gradients through backpropagation, and allow for the simple combination of various factors and models. In addition, they can generate log-likelihood estimates and realistic samples. In an unconditioned generative model, there is no control over the data that is generated. However, through conditioning, the data generation process can be adjusted according to additional information. Here, some data or data with different modalities can be used for conditioning. We explain how to construct such a conditional GAN and conducted two experiments with the MNIST dataset adjusted according to class labels and for multi-modal learning.

To briefly explain multi-modal learning for image labeling, while there have been many successful supervised neural networks recently, there exist two major issues.

Problem 1: Expanding to a model that can accommodate a large number of predicted output categories.

Problem 2: Dealing with one-to-many mapping, like image labeling, rather than one-to-one mapping.

A solution to problem 1 is to utilize additional information from other modalities. To address problem 2, a conditional probabilistic generative model is used. In this case, the input is considered a conditional variable, and one-to-many mapping is achieved by predicting the conditional distribution.

GAN is a method for training data generation models, consisting of a Generative model G that captures the real data distribution, and a Discriminative model D that estimates the probability that samples come from the real training data, not G.

- To learn the Generator distribution p_g over data x, the generator constructs $G(z;\theta_g)$, a mapping function from the prior noise distribution $p_z(z)$ to the data space.

- The Discriminator D outputs a single scalar value between 0 and 1, representing the probability that a sample comes from the real training data rather than p_g.

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

However, this GAN has a disadvantage that it cannot control the generated data. The Conditional GAN extends the original GAN to a conditional model by including additional information y. This y could be a class label or data from another modality. This additional information y is input to both the Generator and Discriminator.

- In the Generator, the input noise $p_z(z)$ and y are combined into a joint hidden representation, and the adversarial training framework with D provides considerable flexibility in constructing such hidden representations.

- In the Discriminator, x and y are represented as inputs and a discriminative function.

This is an extended abstract. The full paper is available at this paper and this paper

$$\min_G \max_D V(D,G) = E_{x\sim p_{data}(x)}[\log D(x|y)] + E_{z\sim p_z(z)}[\log(1-D(G(z|y)))] \quad (2)$$

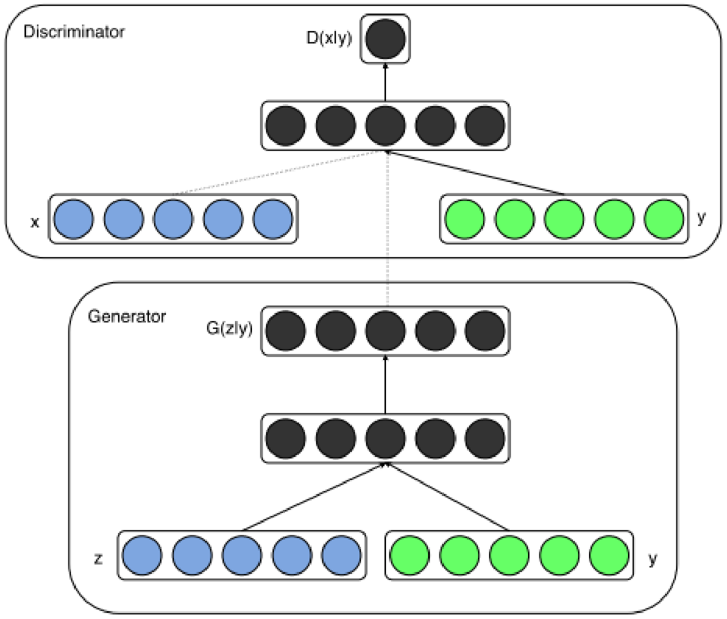

Figure 1: Conditional adversarial net

In this paper, two experiments were conducted. First, CGAN was trained on MNIST images conditioned on one-hot encoded class labels. Next, an experiment was performed to create a tag-vector by inputting image features as additional information into the noise variable. It can be seen as multimodal learning in that the input information is an image domain, but the output is a word token. If you want to see detailed experimental results, I recommend you to read this paper.

Image-to-Image Translation with Conditional Adversarial Networks[1]

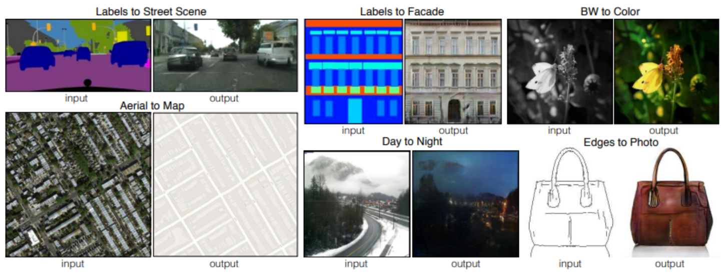

Figure 2: Many problems in image processing, graphics, and vision involve translating an input image into a corresponding output image. These problems are often treated with application-specific algorithms, even though the setting is always the same: map pixels to pixels. Conditional adversarial nets are a general-purpose solution that appears to work well on a wide variety of these problems. Here we show results of the method on several. In each case we use the same architecture and objective, and simply train on different data.

They proposed a framework that uses cGAN to transform one type of image into another type. Because we used GAN, we were able to obtain clearer images than when using L1 or L2 loss. In addition, since we used L1 loss, the concept of structured loss considering the relationship between pixels

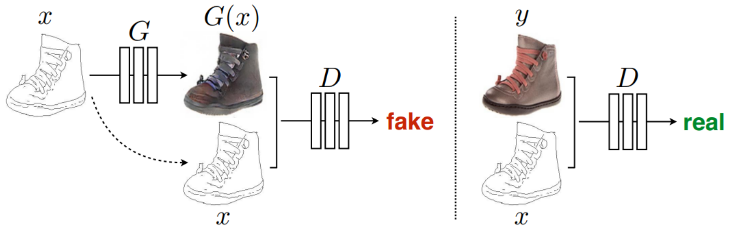

Figure 3: This figure is cGAN. Unlike conventional GANs, G is generated by receiving condition x, and D is also determined by receiving condition x. The model in this figure trains a Conditional GAN to map Edge → photo. Unlike Unconditional GAN, both the generator and discriminator look at the edge map.

was also applied. In order to plausibly fill in the empty areas of an image, one must understand the semantics of the target image accurately. However, using L1/L2 loss alone cannot avoid the issue of the restored image becoming blurry. To resolve this issue, They utilized Adversarial Loss.
Pix2Pix has an Adversarial Loss, which indicates whether an image is a real image, and a Reconstruction Loss, which shows the degree of similarity between the generated image and the ground truth.

$$\mathcal{L}_{GAN}(G,D) = \mathbb{E}_y[\log D(y)] + \mathbb{E}_{x,z}[\log(1-D(G(z)))] \tag{3}$$

$$\mathcal{L}_{cGAN}(G,D) = \mathbb{E}_y[\log D(x,y)] + \mathbb{E}_{x,z}[\log(1-D(G(x,z)))] \tag{4}$$

Adversarial Loss is very similar to the loss of cGAN. In the Generator, when the input image x is inserted, the output image y is generated. Unlike the traditional GAN, noise z is not inserted. In the Discriminator, both the input image x and the generated image G(x) or original image y are input, allowing it to distinguish whether the image is real or fake.

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y,z}[\|y-G(x,z)\|_1] \tag{5}$$

Reconstruction Loss is the L1 loss, representing the degree of similarity between the generated image and the ground truth. It is used to allow the Generator to produce images that are more similar to reality, while leaving the role of the Discriminator to discern fake images intact. Unlike the Context Encoder, only the L1 loss is used because the images were less blurry than when the L2 loss was used.

$$G^* = arg\min_G \max_D \mathcal{L}_{cGAN}(G,D) + \lambda \mathcal{L}_{L1}(G) \tag{6}$$

The final loss function of Pix2Pix is the sum of these two losses.
Both the Generator and Discriminator employ slightly modified structures based on the architecture of DCGAN. Both the Generator and Discriminator use modules in the form of convolution-BatchNorm-ReLU.

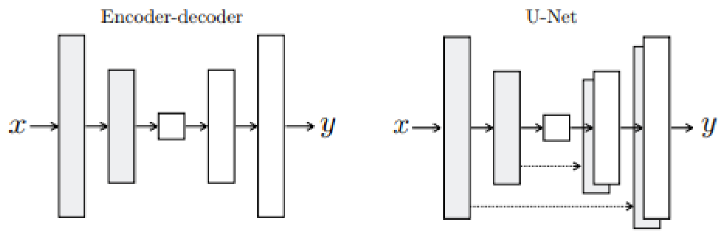

Figure 4: Two choices for the architecture of the generator. The"U-Net"[3] is an encoder-decoder with skip connections between mirrored layers in the encoder and decoder stacks.

Examining Image-to-Image translation problems, the structure is maintained while the surface appearance changes. In other words, the structure of the input is largely utilized in the output. Based on this, the paper uses residual connections in the Encoder-decoder network to share input information and aggregates this with the decoder's features through simple concatenation.
When you combine the encoder and decoder of the Generator, you can see it as a model with a bottleneck structure. Given that Pix2Pix transforms high-resolution images into high-resolution images, it is necessary to preserve low-level information in the input image, such as color or edge locations. Therefore, skip connections are added between the encoder and decoder layers at the same level. In other words, if there are n layers, there is a skip connection between the i-th layer and the n-i-th layer.
It has been reported that using L1 or L2 loss can blur the image. To solve this problem, the Discriminator in GAN is restricted to model only the high-frequency structure, while the L1 loss is responsible for the low-frequency correctness.
L2 loss and L1 loss fail to adequately preserve high-frequency information, resulting in blurry images, but they effectively capture low-frequency information. Hence, the authors constrain the GAN discriminator to model high-frequency structure, while the L1 term is assigned to capture low-frequency information.
When attempting to generate high-frequency information accurately, it would be more efficient to view the image locally rather than in its entirety. This is where PatchGAN comes into play. The method is straightforward. Instead of viewing the entire image when differentiating between real and fake images, the discriminator makes predictions based on NxN patches. Since the patch size N is significantly smaller than the total image size, it has fewer parameters and faster running time.
In Pix2Pix, after concatenating the 256x256 ground truth and the generated fake image, they pass through a convolution layer as depicted in the figure, yielding a final feature map of size 30x30x1. The receptive field of a single pixel in the output feature map corresponds to a 70x70 area in the input image.
The training and testing processes feature a few peculiarities. Firstly, when training the Generator G, instead of minimizing $\log(1-D(x,G(x,z)))$, they modify it to maximize $\log(D(x,G(x,z)))$. Secondly, when training the Discriminator D, they divide the objective by 2 to make D learn more slowly than G. This is likely to prevent a problem where the learning does not progress because the task of the Discriminator D is relatively easy due to the Generator G's inability to create convincing fake images at the beginning of the training. Unique points during test time include the continued use of Dropout, and the use of the test batch's statistics for batch normalization instead of the train batch's statistics.
To explore the generality of conditional GANs, They test the method on a variety of tasks and datasets, including both graphics tasks, like photo generation, and vision tasks, like semantic segmentation. And they test the effect of varying the patch size N of our discriminator receptive fields, from a 1 × 1 "PixelGAN" to a full 286 x 286 "ImageGAN". If you want to see detailed experimental results, I recommend you to read this paper.

[1] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[2] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[3] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.