CS536 - Final Project
Image-to-Image Translation

February 23, 2022

1 Introduction of Generative Adversarial Networks (GAN)

1.1 Concept

Given a large dataset, without any labels, we want to learn a generative model that concisely captures the characteristics of this data. One of the generative models is the generative adversarial network which can synthesize photo-realistic images. The generative adversarial networks shown in Fig. 1 leverage discriminative models to get good generators. A generator is good if the discriminator cannot tell fake data apart from real data.

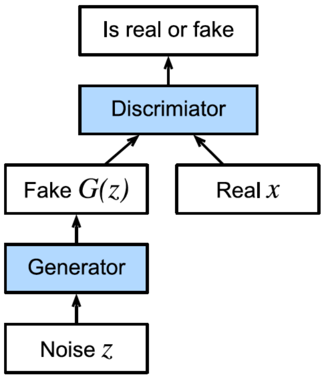

Figure 1: Generative Adversarial Networks

1.2 Task

In this task we build a simple GAN as shown in Fig. 1 in PyTorch and manual inspect of generated images (qualitatively evaluation).

1.2.1 Implementation details

We train the networks from scratch. Weights are initialized from a Gaussian distribution $N(0, 0.02)$.

Datasets. We train the models on two datasets: SyntheticPizza and RealPizza datasets [1]. The SyntheticPizza images and RealPizza images are scaled to 256×256 pixels. The datasets download from `https://drive.google.com/drive/folders/1jaag8cyb72bFHSA-V8ckieTL9uAOsK7r?usp=sharing`

Generator architectures. We adopt the architecture from [2]. Use 9 residual blocks for 256×256 training images. We follow the naming convention used in [2]'s Github repository.

Let $c7s1-k$ denote a 7×7 Convolution-InstanceNorm-ReLU layer with k filters and stride 1. dk denotes a 3×3 Convolution-InstanceNorm-ReLU layer with k filters and stride 2. Reflection padding was used to reduce artifacts. Rk denotes a residual block that contains two 3×3 convolutional layers with the same number of filters on both layer. uk denotes a 3×3 fractional-strided-ConvolutionInstanceNorm-ReLU layer with k filters and stride $1/2$.

The network with 9 residual blocks consists of:

$c7s1-64$,$d128$,$d256$,$R256$,$R256$,$R256$,$R256$,$R256$,$R256$,$R256$,$R256$,$R256$,$u128$,$u64$,$c7s1-3$.

Discriminator architectures. For discriminator networks, we use 70×70 PatchGAN [3].

Let Ck denote a 4×4 Convolution-InstanceNorm-LeakyReLU layer with k filters and stride 2. After the last layer, we apply a convolution to produce a 1-dimensional output. We do not use InstanceNorm for the first $C64$ layer. We use leaky ReLUs with a slope of 0.2.

The discriminator architecture is: $C64-C128-C256-C512$.

Qualitatively evaluation of GAN. The GAN model is trained iteratively over many training epochs. After several epochs use the current state of the model to generate some number of synthesized images. This allows for the post-hoc evaluation of each saved generator model via its synthesized images.

1.3 Due by 03/11/2022

2 Quantitative Evaluation of GAN

2.1 Concept

There are several metrics to assess the performance of a GAN model based on the quality and diversity of the generated images, such as the inception score (IS), the Frechet inception distance (FID), and reconstruction error (ER). They can be combined with qualitative assessment to provide a robust assessment of GAN models. The details of GAN evaluation metrics can be found here [4].

2.2 Task

Pick two evaluation metrics and evaluate your trained GAN models. For example you can compute FID scores of synthesized images. The FID measures the distance between the distributions of synthesized images and real images. A lower FID score means a better performance.

2.3 Due by 03/18/2022

3 Image-to-Image Translation Using Paired Images

3.1 Concept

Image-to-image translation (I2I) is the task of taking images from source domain and transforming them so they have the style of images from target domain while preserving the source content representations. We could classify I2I methods into four categories based on the different ways of leveraging various sources of information: supervised I2I, unsupervised I2I, semi-supervised I2I and few-shot I2I.

We first focus on supervised I2I methods. In supervised I2I settings, many aligned image pairs are as the source domain and target domain.

3.2 Task

In this task we train a widely-used pix2pix [3] model on Dayton dataset [5].

3.2.1 Implementation Details

We train the networks from scratch.

Dataset. We perform the experiments on the Dayton dataset which downloads from `https://www.mediafire.com/folder/f4gga3h86d659/GTCrossView`. We select 76,048 images and create a train/test split of 55,000/21,048 pairs following the same setting of [6]. The images in the original dataset have 354×354 resolution. We resize them to 256×256.

Evaluation.

- Qualitatively evaluation of trained model.
- Compute FCN-score described in [3]. You could also include other evaluation metrics.

Implementation Details. Follow the instructions of https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix and the original paper [3].

3.3 Due by 04/01/2022

4 Image-to-Image Translation Using Unpaired Images

4.1 Concept

Unsupervised I2I methods use two large but unpaired sets of training images to convert images between representations. One of widely-used unsupervised I2I methods is CycleGAN [7].

4.2 Task

In this task we train CycleGAN models to learn a translation map from synthetic pizza domain to real pizza domain, and vice versa. We provide two different synthetic pizza domain images – 'pre-recorded' synthetic pizza and 'live' synthetic pizza. In the 'pre-recorded' synthetic pizza domain, the distribution of ingredients are fixed. In the 'live' synthetic pizza domain, each image is created by yourself using our provided code. You could create your images by changing some factors, such as the density of each ingredient, the size of each ingredient, and the location of each ingredient. The number of images in two synthetic pizza domains are same. We share the same real pizza domain for two synthetic pizza domains. Compare the performance of CycleGAN models trained on two different synthetic pizza images, the students should investigate how these factors impact your results.

4.2.1 Implementation Details

Train CycleGAN models on 'pre-recorded' synthetic pizza domain with real pizza domain and 'live' synthetic pizza domain with real pizza domain from scratch.

Dataset. The 'pre-recorded' synthetic pizza and real pizza images download from https://drive.google.com/drive/folders/1jaag8cyb72bFHSA-V8ckieTL9uAOsK7r?usp=sharing.

The code for creating 'live' synthetic pizza images download from https://drive.google.com/drive/folders/1y2E-R-0nE2_HVC-gm-yiOJSVOpi-s9CT?usp=sharing.

Evaluation

- Qualitatively evaluation of trained models.
- Compute FID score of synthesized real pizza images and FID score of synthesized synthetic pizza images. You could also include other evaluation metrics.

Implementation Details. Follow the instructions of https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix and original paper [7].

4.3 Due by 04/15/2022

5 Enhancement of Unpaired Image-to-Image Translation Model

5.1 Task

- Option 1: Since the real pizza domain is far apart from the synthetic pizza domain, the performance of CycleGAN on pizza images is constrained by the large domain gap. This task is to propose a GAN-based unpaired image-to-image translation model which can compete with the baseline model(CycleGAN). You could read these related work [8; 9].

- Option 2: To understand which parts of that GANs model are responsible for answering paired and unpaired translation mapping through analysing and visualizing the latent space of models.

Implementation Details

Dataset. The 'pre-recorded' synthetic pizza and real pizza images download from https://drive.google.com/drive/folders/1jaag8cyb72bFHSA-V8ckieTL9uAOsK7r?usp=sharing.

The code for creating 'live' synthetic pizza images download from https://drive.google.com/drive/folders/1y2E-R-0nE2_HVC-gm-yiOJSVOpi-s9CT?usp=sharing.

Evaluation. Choose proper evaluation metrics.

References

[1] D. P. Papadopoulos, Y. Tamaazousti, F. Ofli, I. Weber, and A. Torralba, "How to make a pizza: Learning a compositional layer-based gan model," in *proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8002–8011, 2019.

[2] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European conference on computer vision*, pp. 694–711, Springer, 2016.

[3] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1125–1134, 2017.

[4] A. Borji, "Pros and cons of gan evaluation measures," *Computer Vision and Image Understanding*, vol. 179, pp. 41–65, 2019.

[5] N. N. Vo and J. Hays, "Localizing and orienting street views using overhead imagery," in *European conference on computer vision*, pp. 494–509, Springer, 2016.

[6] K. Regmi and A. Borji, "Cross-view image synthesis using conditional gans," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3501–3510, 2018.

[7] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232, 2017.

[8] T. Park, A. A. Efros, R. Zhang, and J.-Y. Zhu, "Contrastive learning for unpaired image-to-image translation," in *European Conference on Computer Vision*, pp. 319–345, Springer, 2020.

[9] Y. Zhao, R. Wu, and H. Dong, "Unpaired image-to-image translation using adversarial consistency loss," in *European Conference on Computer Vision*, pp. 800–815, Springer, 2020.