Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks

Sebastian Rietsch

May 24, 2019

Approach for learning to translate an image from a source domain X to a target domain Y in the absence of paired examples. The goal is to learn a mapping $G : X \rightarrow Y$ such that the distribution of images from $G(X)$ is indistinguishable from the distribution of Y using adversarial loss. Because the mapping is highly under-constrained it gets coupled with an inverse mapping $F : Y \rightarrow X$ and an cycle consistent loss gets introduced to enforce $F(G(X)) \approx X$.

Links:

- `https://junyanz.github.io/CycleGAN/`
- `https://hardikbansal.github.io/CycleGANBlog/`

Introduction

Given: one set of images from domain X and a different set from domain Y. One way to achieve this goal is to train a mapping $G : X \rightarrow Y$ such that the output $\hat{y} = G(x), x \in X$, is indistinguishable from images $y \in Y$ by an adversary trained to classify $\hat{y}$ apart from y. The optimal G thereby translates the domain X to a domain $\hat{Y}$ distributed identically to Y. However, such a translation does not guarantee that an individual input x and output y are paired up in a meaningful way - there are infinitely many mappings G that will induce the same distribution over $\hat{y}$. Also: it is really hard to optimize such an adversarial objective (mode collapse).

These issues call for adding more structure to our objective. Translations should be cycle-consistent, in the sense that if we translate e.g. a sentence from English to French, and then translate it back from French to English, we should arrive back at the original sentence. Mathematically: two translators $G : X \rightarrow Y$ and $F : Y \rightarrow X$, G and F should be inverse of each other and both should be bijections.

This structural assumptions gets applied by training both mappings G and F simultaneously and adding a *cycle consistency loss* that encourages $F(G(x)) \approx x$ and $G(F(y)) \approx y$.

Combining this loss with adversarial losses on domain X and Y yields the full objective for unpaired image-to-image translation.

Formulation

The goal is to learn mapping functions between two domains X and Y given training examples $\{x_i\}_{i=1}^N$ where $x_i \in X$ and $\{y_j\}_{j=1}^M$ where $y_j \in Y$. We denote the data distribution as $x \sim p_{data}(x)$ and $y \sim p_{data}(y)$. The model includes two mappings $G : X \rightarrow Y$ and $F : Y \rightarrow X$. In addition, two adversarial discriminators D_X and D_Y get introduced, where D_X aims to distinguish between images $\{x\}$ and translated images $\{F(y)\}$; in the same way, D_Y aims to discriminate between $\{y\}$ and $\{G(x)\}$. the objective contains two types of terms: 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 mapping G and F from contradicting each other.

Adversarial Loss

For the mapping function $G : X \rightarrow Y$ and its discriminator D_Y, the loss is:

$$\begin{aligned}\mathcal{L}_{GAN}(G, D_Y, X, Y) = \mathbb{E}_{y\sim p_{data}(y)}[log(D_Y(y))] \\ + \mathbb{E}_{x\sim p_{data}(x)}[log(1 - D_Y(G(x))],\end{aligned} \tag{1}$$

where G tires to generate images $G(x)$ that look similar to images from domain Y, while D_Y aims to distinguish between translated samples $G(x)$ and real samples y. G aims to minimize this objective against an adversary D that tries to maximize it, i.e. $min_G max_{D_Y} \mathcal{L}_{GAN}(G, D_Y, X, Y)$. A similar adversarial loss for the mapping function $F : Y \rightarrow X$ and its discriminator D_X gets introduced as well: i.e. $min_F max_{D_X} \mathcal{L}_{GAN}(F, D_X, Y, X)$.

Cycle Consistency Loss

As already discussed, adversarial loss alone cannot guarantee that the learned function can map an individual input x_i to a desired output y_i. To further reduce the space of possible mapping functions, we argue that the learned mapping functions should be cycle-consistent. Two types:

- The image translation cycle should be able to bring x back to the original image, i.e. $x \rightarrow G(x) \rightarrow F(G(x)) \approx x$. We call this *forward cycle.*
- For each image y from domain Y, G and F should also satisfy *backward cycle consistency*: $y \rightarrow F(y) \rightarrow G(F(y)) \approx y$.

From this follows the *cycle consistency loss*:

$$\begin{aligned}\mathcal{L}_{cyc}(G, F) = \mathbb{E}_{x\sim p_{data}(x)}[\|F(G(x)) - x\|_1] \\ + \mathbb{E}_{y\sim p_{data}(y)}[\|F(G(y)) - y\|_1]\end{aligned} \tag{2}$$

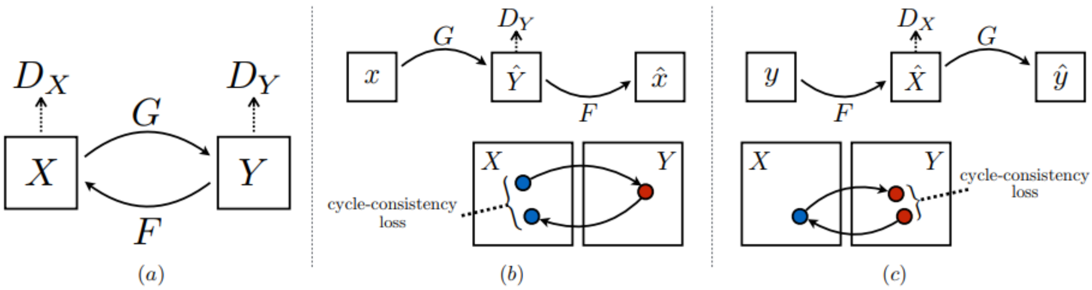

Figure 3: (a) Our model contains two mapping functions $G : X \to Y$ and $F : Y \to X$, and associated adversarial discriminators D_Y and D_X. D_Y encourages G to translate X into outputs indistinguishable from domain Y, and vice versa for D_X and F. To further regularize the mappings, we introduce two *cycle consistency losses* that capture the intuition that if we translate from one domain to the other and back again we should arrive at where we started: (b) forward cycle-consistency loss: $x \to G(x) \to F(G(x)) \approx x$, and (c) backward cycle-consistency loss: $y \to F(y) \to G(F(y)) \approx y$

Replacing the L1 norm in this loss with an adversarial loss between $F(G(x))$ and x and vice verse did not induce improved performance.

Full objective

The full objective is

$$\begin{aligned}\mathcal{L}(G, F, D_X, D_Y) = &\mathcal{L}_{GAN}(G, D_Y, X, Y)\\ &+ \mathcal{L}_{GAN}(F, D_X, Y, X)\\ &+ \lambda\mathcal{L}_{cyc}(G, F),\end{aligned} \tag{3}$$

where λ controls the relative importance of the two objectives. We aim to solve:

$$G^*, F^* = argmin_{G,F} max_{D_X,D_Y} \mathcal{L}(G, F, D_X, D_Y) \tag{4}$$

The model can be viewed as training two autoencoders: we learn on autoencoder $F \circ G : X \to X$ jointly with another $G \circ F : Y \to Y$. These autoencoders each have special internal structured: they map an image to itself via an intermediate representation that is a translation of the image into another domain.

Implementation

Network Architecture

- Generator: Adopted from Johnson et al. [1]
  - Two stride-2 convolutions, several residual blocks and two fractionally-strided convolutions with stride $\frac{1}{2}$

  - 6 blocks for 128×128 images and 9 blocks for 256×256 and higher-resolution training images
  - Similar to Johnson et al. instance normalization gets used
- Discriminator: 70×70 PatchGANs [2] which aim to classify whether 70×70 overlapping image patches are real or fake. Such a patch-level discriminator architecture has fewer parameters than a full-image discriminator and can work on arbitrarily-sized images in a fully convolutional fashion

Training details

- $\mathcal{L}_{GAN}$: replace negative log likelihood objective by a least squares loss [3]. This loss is more stable during training and generates higher quality results. In particular, for GAN loss we train G to minimize $\mathbb{E}_{x \sim p_{data}(x)}[(D(G(X)) - 1)^2]$ and train the D to minimize $\mathbb{E}_{y \sim p_{data}(y)}[(D(y) - 1)^2] + \mathbb{E}_{x \sim p_{data}(x)}[D(G(x))^2]$
- To reduce model oscillation Shrivastava et al.'s [4] strategy gets applied, where the discriminator gets updated using a history of generated images rather than the ones produced by the latest generators. An image buffer that stores the 50 previously created images gets kept.
- $\lambda = 10$
- Adam solver with a batch size of 1, learning rate of 0.0002. Same learning rate for the first 100 epochs and linearly decay the rate to zero over the next 100 epochs.

What's next?

- Generator is inspired by Johnson et. al. $\rightarrow$ Need to understand and be able to implement
  - Maybe implement the style transfer algorithm?
  - Reflection padding
  - Residual blocks
- Discriminator uses PatchGAN $\rightarrow$ Need to understand and be able to implement
  - TODO

References

[1] J. Johnson, A. Alahi, and F. Li. Perceptual losses for real-time style transfer and super-resolution. *CoRR*, abs/1603.08155, 2016.

[2] C. Li and M. Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. *CoRR*, abs/1604.04382, 2016.

[3] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, and Z. Wang. Multi-class generative adversarial networks with the L2 loss function. *CoRR*, abs/1611.04076, 2016.

[4] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb. Learning from simulated and unsupervised images through adversarial training. *CoRR*, abs/1612.07828, 2016.