Unsupervised Distribution Alignment / The Revolution will be Unsupervised

Presenters: *Hu Hengchang*
Authors: Xin Jie, ***Please put your names here.***

Image to image distribution alignment problem

- Align distribution of semantic image to distribution of regular images
- Align distribution of day images to night images
- Align distribution of black and white images to colour images

Application examples:

- Image to Image: pix2pix
- Text to text: machine translation
- Image to text: captioning
- Text to Image, voice to text, text to voice

Marginal Matching

- p(b) is the ideal distribution of b, q(b) is the approximated distribution of b
- Same for q(a) and p(a)

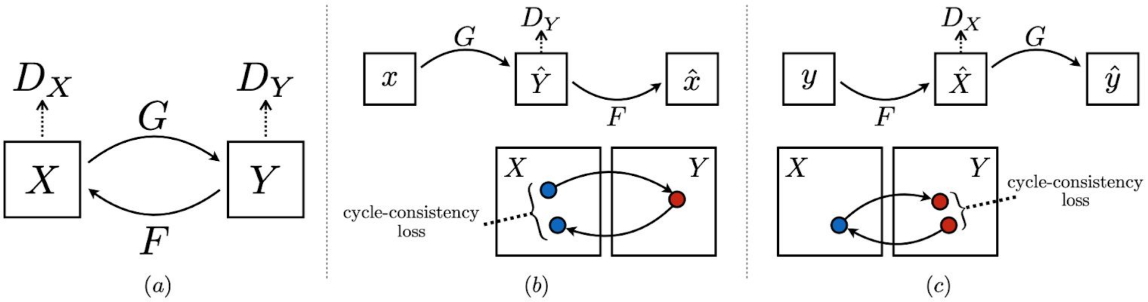

Figure 3: (a) Our model contains two mapping functions $G : X \to Y$ and $F : Y \to X$, and associated adversarial discriminators D_Y and D_X. D_Y encourages G to translate X into outputs indistinguishable from domain Y, and vice versa for D_X and F. To further regularize the mappings, we introduce two *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: $y \to F(y) \to G(F(y)) \approx y$

From the CycleGAN paper.

Cycle Consistency

- Deterministic mapping from a to b

From
http://www.cs.toronto.edu/~rgrosse/courses/csc321_2018/assignments/a4-handout.pdf:

-

Recently, Generative Adversarial Networks have been successfully applied to image translation, and have sparked a resurgence of interest in the topic. The basic idea behind the GAN-based approaches is to use a conditional GAN to learn a mapping from input to output images. The loss functions of these approaches generally include extra terms (in addition to the standard GAN loss), to express constraints on the types of images that are generated. A recently-introduced method for image-to-image translation called CycleGAN is particularly interesting because it allows us to use un-paired training data. This means that in order to train it to translate images from domain X to domain Y , we do not have to have exact correspondences between individual images in those domains. For example, in the paper that introduced CycleGANs, the authors are able to translate between images of horses and zebras, even though there are no images of a zebra in exactly the same position as a horse, and with exactly the same background, etc. Thus, CycleGANs enable learning a mapping from one domain X (say, images of horses) to another domain Y (images of zebras) without having to find perfectly matched training pairs.

To summarize the differences between paired and un-paired data, we have:

- Paired training data: {(x (i) , y(i))} N i=1
- Un-paired training data:
  - Source set: {x (i)} N i=1 with each x (i) ∈ X
  - Target set: {y (j)}M j=1 with each y (j) ∈ Y
  - For example, X is the set of horse pictures, and Y is the set of zebra pictures, where there are no direct correspondences between images in X and Y

Cycle GAN paper

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

Jun-Yan Zhu, Taesung Park, Phillip Isola, Alexei A. Efros

https://arxiv.org/abs/1703.10593

Objective function:

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

* L_GAN and L_cyc are as following:

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

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

Augmented CycleGAN paper https://arxiv.org/abs/1802.10151

-

A version of CycleGAN that allows it to capture a distribution and non-deterministic model of the X and Y data. Does it address mode collapse when the data has multiple modes?

WORD TRANSLATION WITHOUT PARALLEL DATA paper https://arxiv.org/abs/1710.04087
NLP version of word alignment as mentioned in the last part of the lecture.

Part 2

Classifiers at present aren't particularly good at identifying things outside a nicely constrained dataset because they still depend on categories.

- The idea of categories are problematic because it's hard to clearly define what belongs in a category. It's always hard to define a rule that applies to every single member of category. (Is a tomato a fruit? Is a cucumber a fruit?)
    - Even for cases where humans define both the category and the object, it's still difficult! (Is Pluto a planet? What defines a country?)
- The context in which the object is in can also change the categorization. Efros uses the example of a stick figure car and road
    - (UwU) (OwO) - On the left, we know I'm typing an emoji because of how it's arranged, and we know that the 'U' and the 'O' are eyes. But on their own, 'U' and 'O' are just alphabets if they were shown alone without additional context.
- Models are relying on having large amounts of data to overcome this disadvantage
    - As long as you have near-infinite amounts of data, K-Nearest Neighbours can achieve "unreasonably effective" results even if it's not very elaborate.
- More data is better than more complex models. cf Fred Jelinek quote from NLP: "Every time I fire a linguist, the performance of the speech recognizer goes up". ⇒ *The less complex the model, the more data I can feed through, the better the performance.*

-

- The “autoencoder” within the human brain is somehow able to encode tremendous amounts of detail, even for exemplar and state differences (only a loss of about 5% on a much harder task)
- If you can structure your problem in a way that’s conducive to self-supervision, you can get exponentially more data since you no longer need to label anything
    - In the case of videos, you already have two sets of correlated data: the audio bit, and the visual bit. You can then train a model to learn what action onscreen is making the noise, and you need not annotate anything!