Composable Unpaired Image to Image Translation

Laura Graesser, Anant Gupta
New York University

Problem Setting

Project objective: to learn a mapping from one image domain, X, to another domain Y when we do not have training examples from the domain Y.

Suppose we have images which can be described with two characteristics: x_1 or y_1, and x_2 or y_2. For example blond / brunette, or smiling / not smiling. Then let:

$$X = x_1 \ AND \ x_2$$

$$Y = y_1 \ AND \ y_2$$

So the mapping from X to Y can also be described as:

$$\begin{aligned} & f : X \Rightarrow Y \\ & \Leftrightarrow x_1 \ AND \ x_2 \Rightarrow y_1 \ AND \ y_2 \end{aligned} \tag{10}$$

Also suppose we only have access to images labeled with the marginal characteristics (e.g. x_1, y_2), instead of the joint characteristics (e.g. X, Y).

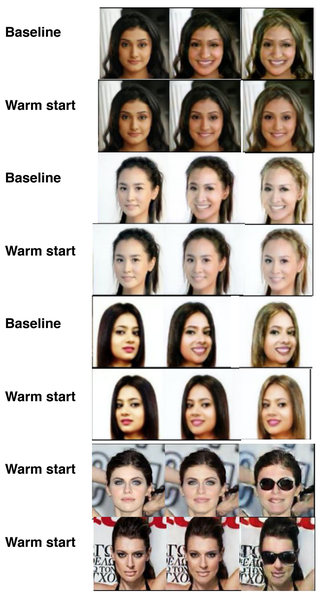
This project explores whether it is possible to just learn the marginal mappings $g : x_1 \rightarrow y_1$ and $h : x_2 \rightarrow y_2$, and compose them to yield f where,

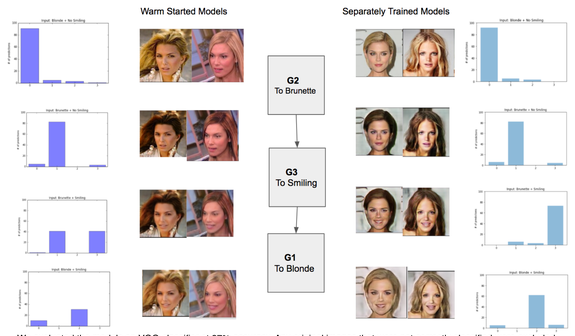
$$f = g \circ h$$

Potential benefits from this approach include

- Facilitating training on larger datasets since only data with the marginal and thus more general labels are required
- Making it possible to translate between joint combinations of the marginal distributions that never appeared in the training set.

Results





We evaluated the models on VGG-classifier at 87% accuracy. Any original images that were not correctly classified were excluded

Label Map

0: Blonde + No Smiling
1: Brunette + No Smiling
2: Blonde + Smiling
3: Brunette + Smiling

Loss function

$$\begin{aligned} \min_{E_1,E_2,E_3,E_4,G_1,G_2,G_3,G_4} \quad & \mathcal{L}_{\text{VAE}_1}(E_1, G_1) + \mathcal{L}_{\text{GAN}_1}(E_1, G_1, D_1) + \mathcal{L}_{\text{CC}_1}(E_1, G_1, E_2, G_2) \\ \max_{D_1,D_2,D_3,D_4} \quad & + \mathcal{L}_{\text{VAE}_2}(E_2, G_2) + \mathcal{L}_{\text{GAN}_2}(E_2, G_2, D_2) + \mathcal{L}_{\text{CC}_2}(E_2, G_2, E_1, G_1) \\ & + \mathcal{L}_{\text{VAE}_3}(E_3, G_3) + \mathcal{L}_{\text{GAN}_3}(E_3, G_3, D_3) + \mathcal{L}_{\text{CC}_3}(E_3, G_3, E_4, G_4) \\ & + \mathcal{L}_{\text{VAE}_4}(E_4, G_4) + \mathcal{L}_{\text{GAN}_4}(E_4, G_4, D_4) + \mathcal{L}_{\text{CC}_4}(E_4, G_4, E_3, G_3) \end{aligned}$$

The **VAE objectives** (given for just one pair of distributions) are:

$$\mathcal{L}_{\text{VAE}_1}(E_1, G_1) = \lambda_1 \text{KL}(q_1(z_1|x_1)||p_\eta(z)) - \lambda_2 \mathbb{E}_{z_1 \sim q_1(z_1|x_1)}[\log p_{G_1}(x_1|z_1)] \tag{4}$$

$$\mathcal{L}_{\text{VAE}_2}(E_2, G_2) = \lambda_1 \text{KL}(q_2(z_2|x_2)||p_\eta(z)) - \lambda_2 \mathbb{E}_{z_2 \sim q_2(z_2|x_2)}[\log p_{G_2}(x_2|z_2)]. \tag{5}$$

where the hyper-parameters λ_1 and λ_2 control the weights of the objective terms and the KL divergence terms penalize deviation of the distribution of the latent code from the prior distribution.

The **GAN objective** functions (given for just one pair of distributions) are:

$$\mathcal{L}_{\text{GAN}_1}(E_1, G_1, D_1) = \lambda_0 \mathbb{E}_{x_1 \sim P_{X_1}}[\log D_1(x_1)] + \lambda_0 \mathbb{E}_{z_2 \sim q_2(z_2|x_2)}[\log(1 - D_1(G_1(z_2)))] \tag{6}$$

$$\mathcal{L}_{\text{GAN}_2}(E_2, G_2, D_2) = \lambda_0 \mathbb{E}_{x_2 \sim P_{X_2}}[\log D_2(x_2)] + \lambda_0 \mathbb{E}_{z_1 \sim q_1(z_1|x_1)}[\log(1 - D_2(G_2(z_1)))]. \tag{7}$$

A **VAE-like objective function is used model the cycle-consistency constraint**, which is given (for just one pair of distributions) by,

$$\begin{aligned} \mathcal{L}_{\text{CC}_1}(E_1, G_1, E_2, G_2) = & \lambda_3 \text{KL}(q_1(z_1|x_1)||p_\eta(z)) + \lambda_3 \text{KL}(q_2(z_2|x_1^{1\rightarrow 2})||p_\eta(z)) - \\ & \lambda_4 \mathbb{E}_{z_2 \sim q_2(z_2|x_1^{1\rightarrow 2})}[\log p_{G_1}(x_1|z_2)] \end{aligned} \tag{8}$$

$$\begin{aligned} \mathcal{L}_{\text{CC}_2}(E_2, G_2, E_1, G_1) = & \lambda_3 \text{KL}(q_2(z_2|x_2)||p_\eta(z)) + \lambda_3 \text{KL}(q_1(z_1|x_2^{2\rightarrow 1})||p_\eta(z)) - \\ & \lambda_4 \mathbb{E}_{z_1 \sim q_1(z_1|x_2^{2\rightarrow 1})}[\log p_{G_2}(x_2|z_1)]. \end{aligned} \tag{9}$$

Note: The loss function is adapted from the function used in Unsupervised Image-to-Image Translation Networks, Ming-Yu Liu, Thomas Breuel, Jan Katz, Arxiv, October 2017

Related work

- Unsupervised Image-to-Image Translation Networks, Ming-Yu Liu, Thomas Breuel, Jan Katz, NVIDIA, arxiv, October 2017
  - Key ideas: shared latent space between image domains, cycle consistency

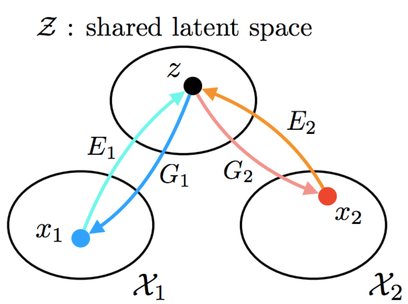
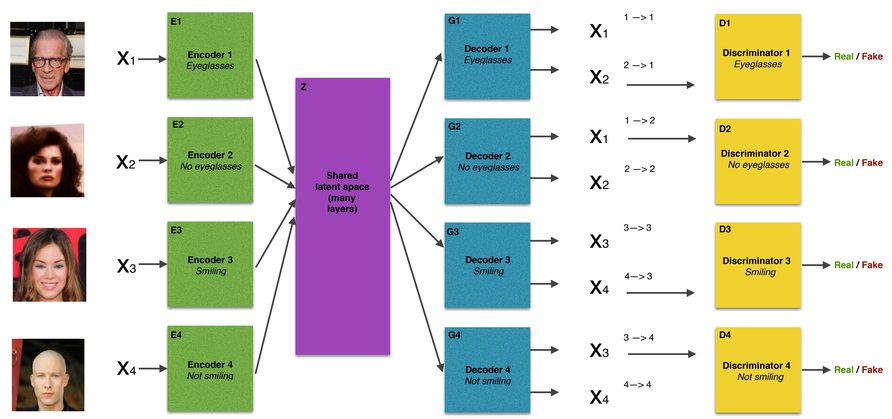


Image source: Unsupervised Image-to-Image Translation Networks, Ming-Yu Liu, Thomas Breuel, Jan Katz, NVIDIA, arxiv, October 2017

- Unpaired Image to Image Translation using Cycle-Consistent Adversarial Networks, Jun-Yan Zhu, Teasing Park, Phillip Isola, Alexei E. Efros, Berkeley AI Research, arxiv, 2017
- Coupled Generative Adversarial Networks, Ming-Yu Liu, Oncel Tuzel, NIPS, 2016

Model Architecture



Encoders

- Conv, leaky ReLU, stride 1
- 3 x Conv, leaky ReLU, stride 2
- 3 x residual block

Decoders

- 3 x residual block
- 2 x Conv Transpose, leaky ReLU
- Conv Transpose

Shared

- Residual Block
- Gaussian Noise
- Residual Block

Discriminators

Separate
- 3 x Conv, leaky ReLU, stride 2
- 1 x Conv

Shared
- 4 x Conv, leaky ReLU, stride 2
- 1 x Conv

Key

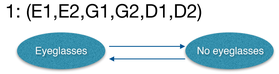
Discriminator
Generator - encoder
Generator - decoder
Generator - shared latent space

Note: We adapted the model architecture from Unsupervised Image-to-Image Translation Networks, Ming-Yu Liu, Thomas Breuel, Jan Katz, Arxiv, October 2017. We are very grateful to the authors for making all of their source code available and providing instructions for its use.

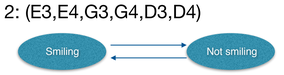
Training & Inference

Training

- The model is conceptually split into two, each part responsible for learning to translate between one pair of distributions
  - 1: (E1,E2,G1,G2,D1,D2)



  - 2: (E3,E4,G3,G4,D3,D4)



- The shared latent space is responsible for ensuring realistic translations between image pairs the model has never seen before
- Three models
  1. Baseline: Two separately trained models
  2. Jointly trained, from scratch
  3. Jointly trained, initialized with separately trained

Inference

No eyeglasses, not smiling X → x —> E2 —> G1 —> y → **No eyeglasses, smiling** Y → y —> E4 —> G3 —> z → **Eyeglasses, smiling** Z

No eyeglasses & smiling examples in training set