Cycle-consistency for Normalizing Flow Models

Jen Weng (zzweng@stanford.edu)

Summary

Unsupervisesd Image-to-image translation is a task in which we teach a model to translate from one style to the other from two unpaired datasets.

CycleGan is GAN network that learns image-to-image translation. It augments the GAN loss with another term (*cycle loss*) that ensures that translating from one domain to another is reversible. Normalizing flow models automatically guarantee such consistency

In this project, we leveraged a flow model as a replacement to the cycle loss term in the CycleGAN objective. We implemented a flow model to learn the translation from style A to style B. Since flow models are invertible, by passing the image of style B through the inverse of the flow, we will get the translated image of style A. The flow model can then be trained adversarially using a GAN. Real NVP is used here because it is fast to compute the inverse.

Background/Problem Setup

We will denote the domain of the images of style A as X, and style B, Y. The algorithm will take images of style A in batches, and pass it through the Real NVP flow model to get the (fake) image of style B. Vice versa.
The GAN has two discriminators (Dx and Dy) and one generator (F). F translates image A to image B. F inverse goes the other direction. Dx aims to distinguish between real images of style A and fake images of style A generated by the inverse of F. Dy aims to distinguish between real images of style B and fake images of style B generated by F.

In the loss function, we will include the loss function for the generator (the flow model) as well as the identity loss. The intuition for identity loss is that F should do nothing for images of style B.

Technical Methods

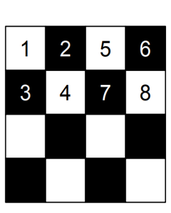
Real NVP

Coupling layer

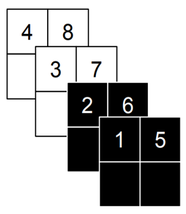
$$y = b \odot x + (1 - b) \odot \Big(x \odot \exp\big(s(b \odot x)\big) + t(b \odot x) \Big). \quad (9)$$

$$\begin{cases} y_{1:d} &= x_{1:d} \\ y_{d+1:D} &= x_{d+1:D} \odot \exp\big(s(x_{1:d})\big) + t(x_{1:d}) \end{cases} \quad (7)$$

Mask types



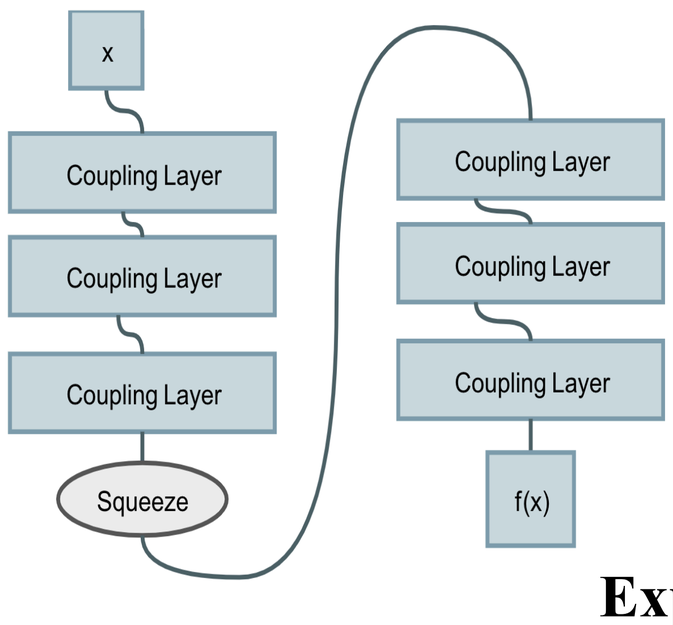
Checkerboard (before the squeeze operation)



Channel-wise (after the squeeze operation)

Multi-scale architecture

- Fig 1: the first scale in a multi-scale architecture
- The final scale only contains 4 coupling layers.
- Coupling layers in each scale use masks with alternating patterns
- Squeeze operation transforms an (s, s, c) tensor into an (s/2, s/2, 4c) tensor.
- To reduce the computational and memory cost, we factor out half of the dimensions after each scale



GAN Setup

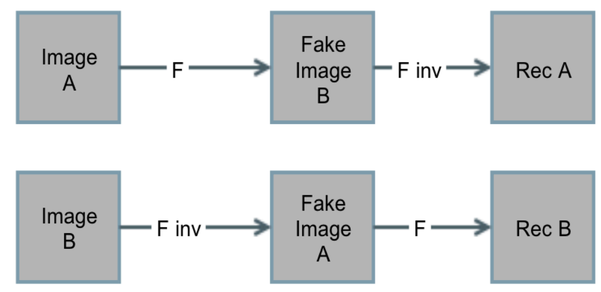


Fig 2: F is a flow model using Real NVP. F inv is inverse of the model.

$$\mathcal{L}_{D_Y} = \frac{1}{2}(\mathbb{E}_y[(D_Y(y) - 1)^2] + \mathbb{E}_x[(D_Y(F(x)))^2])$$

$$\mathcal{L}_{D_X} = \frac{1}{2}(\mathbb{E}_x[(D_X(x) - 1)^2] + \mathbb{E}_y[(D_X(F^{-1}(y)))^2])$$

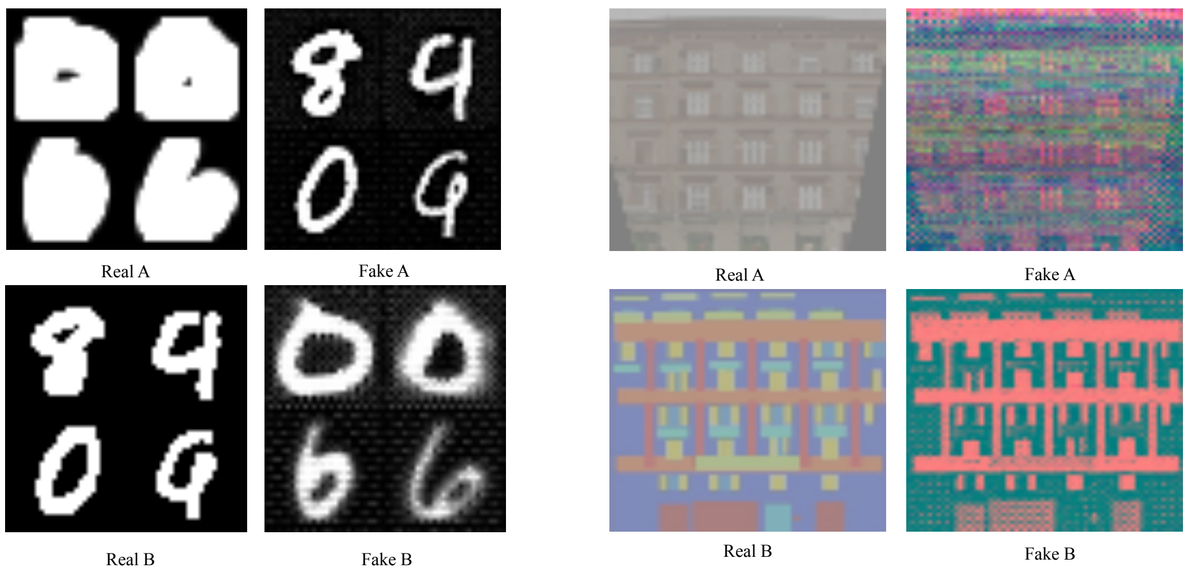
$$\mathcal{L}_{G_X} = \mathbb{E}_x[(D_X(F(x)) - 1)^2] + \lambda_X \lambda_{idt} \mathbb{E}_y[||F(y) - y||_1]$$
$$\mathcal{L}_{G_Y} = \mathbb{E}_y[(D_Y(F^{-1}(y)) - 1)^2] + \lambda_Y \lambda_{idt} \mathbb{E}_x[||F^{-1}(x) - x||_1]$$

- No need to include the cycle loss, because we are explicitly using the inverse of F for the reverse translation.
- Replace the negative log likelihood objective by a least-squares loss for stabilzing the model training procedure.

Experiments

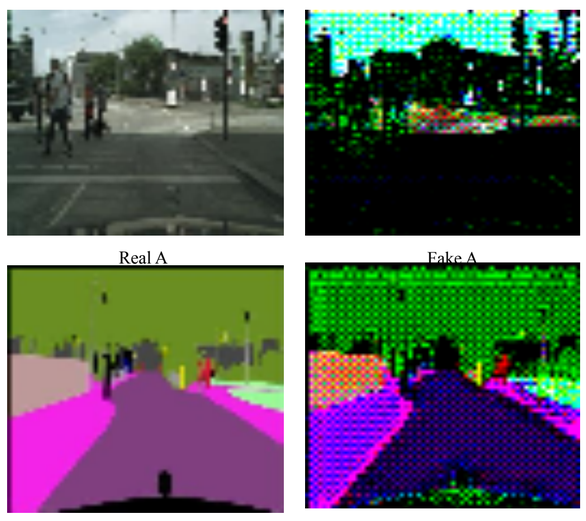
We trained the model on three datasets, MNIST/USPS digits dataset, facades dataset and cityscapes dataset. The first dataset is 28 by 28 pixels and is trained with batch size 32. The other two datasets are downsampled to 64 by 64 pixels and trained with batch size 1 (because of their high computational cost). The outputs are fake images produced by F and F inv. (See some examples to the right)
The parameters used are as following:

- $\lambda_X, \lambda_Y, \lambda_{idt}$: 10, 10, 0.5
- Learning rate: 0.005
- s and t in the coupling layers are learned by a ResNet with 8 residual blocks and 64 feature maps
- ADAM optimizer with default hyperparameters and weight



USPS Digits / MNIST
Epoch = 1, iteration = 14

Facades
Epoch = 1, iteration = 10



Cityscapes
Epoch = 1, iteration = 7

References

Dinh, L., Sohl-Dickstein, J., Bengio, S. (2016). Density estimation using Real NVP. arXiv preprint arXiv:1605.08803.
X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. P. Smolley. Least squares generative adversarial networks. In CVPR. IEEE, 2017.
Zhu, J. Y., Park, T., Isola, P., Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. arXiv:1703.10593.