EC795P - Deep Learning and Computer Vision

Group F Critical Analysis Report
On
Generative Adversarial Nets

Drawbacks of deep generative model

- Difficult to approximate probabilistic computations arising in maximum likelihood estimation

- Difficult to use the benefits of piecewise linear units in the generative context

GAN Base Concept

- The generative model - a team of counterfeiters, trying to produce fake currency and use it without detection,
- Pitted against an adversary
- The discriminative model - the police, trying to detect the counterfeit currency

Algorithm

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

for number of training iterations **do**

  for k steps **do**

  - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  - Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
  - Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right)\right].$$

  end for

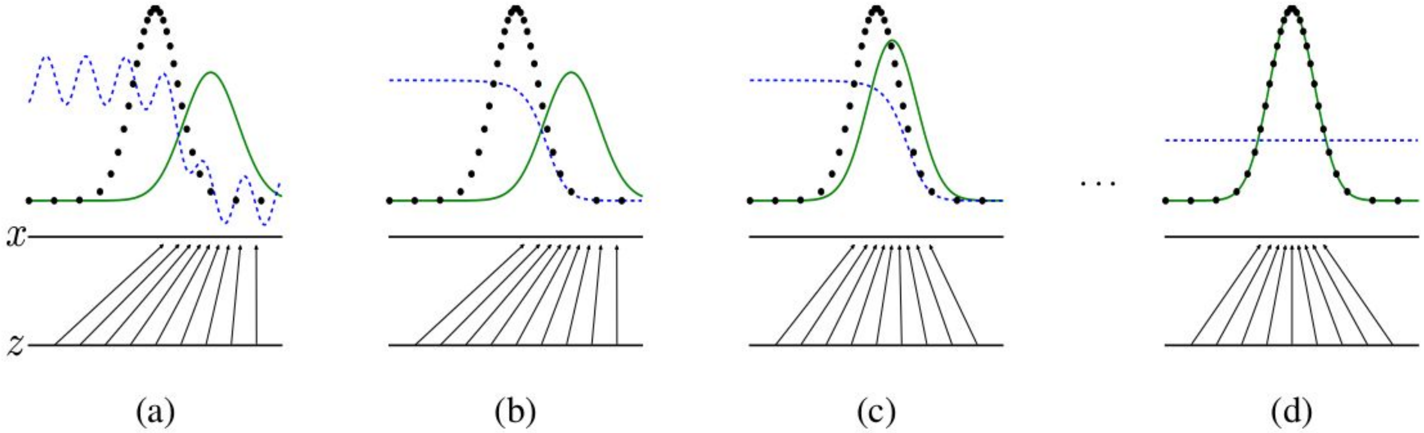
  - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  - Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

Working of Adversarial Net



Update discriminative distribution(D, blue, dashed line), so that it discriminates between samples from the data generating distribution (black, dotted line) from those of the generative distribution(G, green, solid line) and the generative distribution needs to fool the discriminator

GAN Training

- Because a GAN contains two separately trained networks, its training algorithm must address two complications:
    - GANs must juggle two different kinds of training (generator and discriminator).
    - GAN convergence is hard to identify.

GAN Training - Alternate Training

- The generator and the discriminator have different training processes. So how do we train the GAN as a whole?
- GAN training proceeds in alternating periods:
    - The discriminator trains for one or more epochs.
    - The generator trains for one or more epochs.
    - Repeat steps 1 and 2 to continue to train the generator and discriminator networks.

GAN Training - Alternating Training

- We keep the generator constant during the discriminator training phase. As discriminator training tries to figure out how to distinguish real data from fake, it has to learn how to recognize the generator's flaws. That's a different problem for a thoroughly trained generator than it is for an untrained generator that produces random output.
- Similarly, we keep the discriminator constant during the generator training phase. Otherwise the generator would be trying to hit a moving target and might never converge.
- It's this back and forth that allows GANs to tackle otherwise intractable generative problems. We get a toehold in the difficult generative problem by starting with a much simpler classification problem. Conversely, if you can't train a classifier to tell the difference between real and generated data even for the initial random generator output, you can't get the GAN training started.
-

Convergence

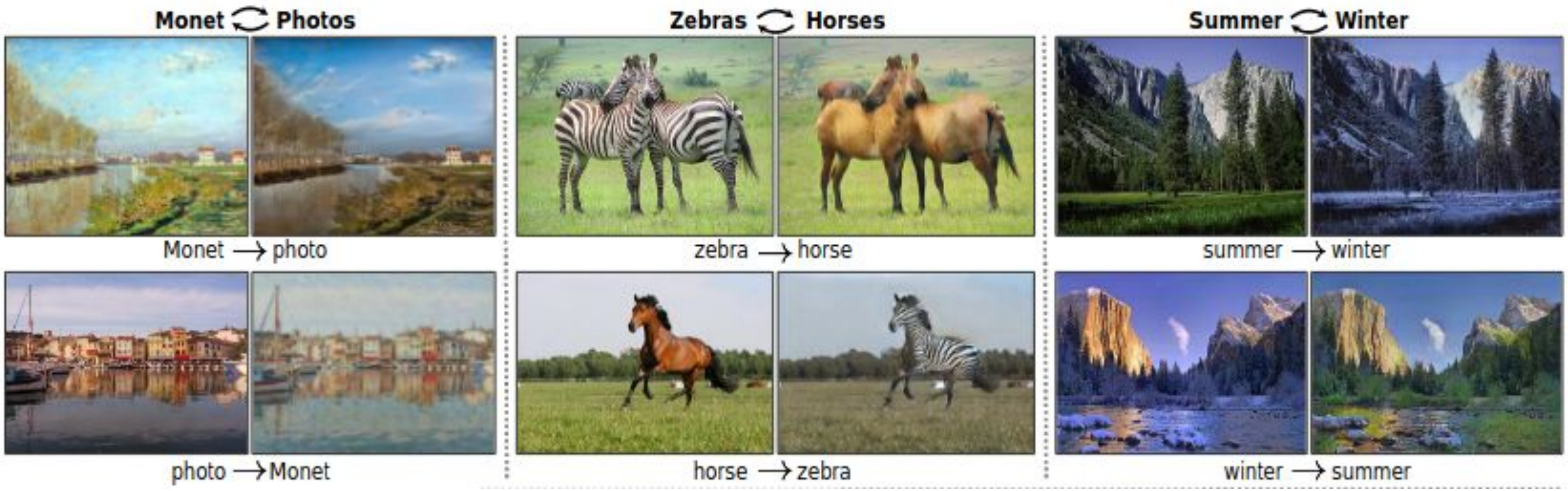
As the generator improves with training, the discriminator performance gets worse because the discriminator can't easily tell the difference between real and fake. If the generator succeeds perfectly, then the discriminator has a 50% accuracy. In effect, the discriminator flips a coin to make its prediction.

This progression poses a problem for convergence of the GAN as a whole: the discriminator feedback gets less meaningful over time. If the GAN continues training past the point when the discriminator is giving completely random feedback, then the generator starts to train on junk feedback, and its own quality may collapse.

For a GAN, convergence is often a fleeting, rather than stable, state.

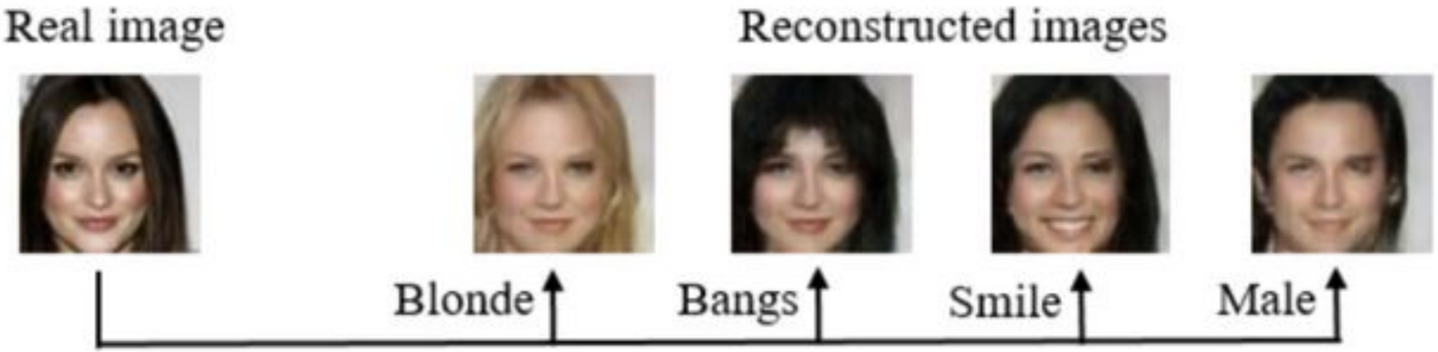
Applications of GAN

Image Processing and Computer Vision



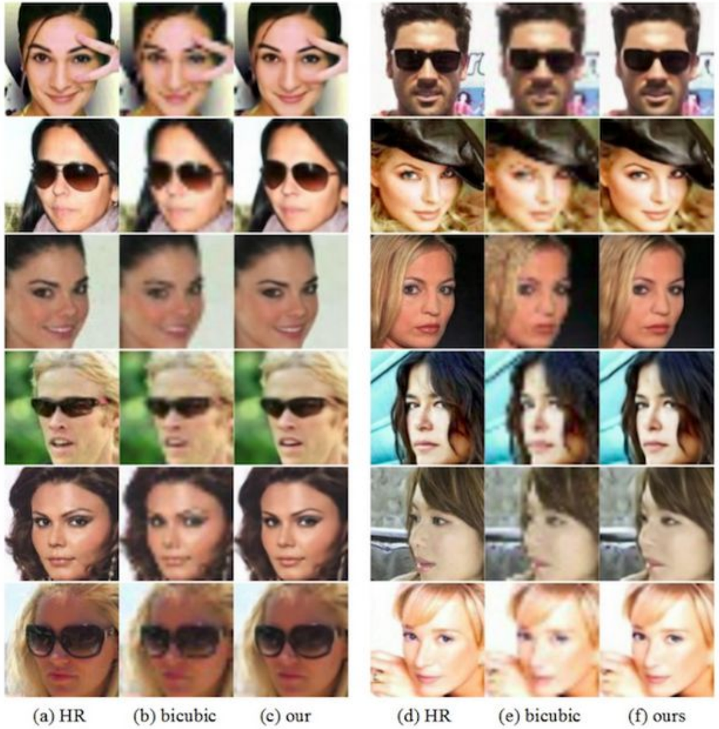
Text -to-Image

Photograph Editing



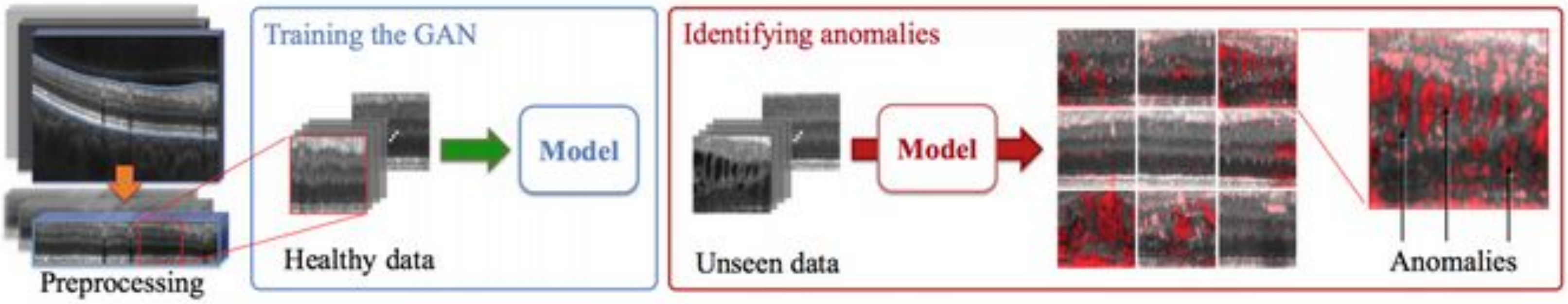
Example of Face Photo Editing with IcGAN.Taken from Invertible Conditional GANs For Image Editing, 2016.

Super Resolution



Example of High-Resolution Generated Human FacesTaken from High-Quality Face Image SR Using Conditional Generative Adversarial Networks, 2017.

Medical Anomaly Detection



AnoGAN

Problems of GAN

Vanishing Gradient

- If the discriminator is too good, then generator training can fail due to vanishing gradients
- Remedies:
  - Wasserstein loss
  - Modified minimax loss

Mode Collapse

- If a generator produces an especially plausible output, the generator may learn to produce only that output
- Remedies:
  - Wasserstein loss
  - Unrolled GANs

Failure to Converge

- If the generator keeps getting better, the performance of discriminator decreases.
- When GAN continues training past the point when the discriminator is giving completely random feedback, then the generator starts to train on junk feedback, and its own quality may collapse.
- Remedies:
    - Adding noise to discriminator inputs
    - Penalizing discriminator weights

“The most interesting idea in the last 10 years in ML”

-Yann LeCun

Thank You