CS 446/ECE 449: Machine Learning

A. G. Schwing

University of Illinois at Urbana-Champaign, 2020

L26: Generative Adversarial Nets

Goals of this lecture

- Getting to know Generative Adversarial Nets
- Understanding generative methods
- Differentiating between discriminative and generative methods

Reading Material:

- I. Goodfellow et al.; Generative Adversarial Networks; arxiv.org/abs/1406.2661
- M. Arjovsky et al.; Wasserstein GAN; arxiv.org/abs/1701.07875

Recap: Maximum likelihood so far?

Model:

$$p(\boldsymbol{y}|x) = \frac{\exp F(\boldsymbol{y}, x, \boldsymbol{w})/\epsilon}{\sum_{\hat{\boldsymbol{y}}} \exp F(\hat{\boldsymbol{y}}, x, \boldsymbol{w})/\epsilon}$$

- $\boldsymbol{y}$: discrete output space
- x: input data

Now:

How about modeling a distribution $p(x)$ for the data?

Why modeling a distribution $p(x)$?

- Synthesis of objects (images, text)
- Environment simulator (reinforcement learning, planning)
- Leveraging unlabeled data

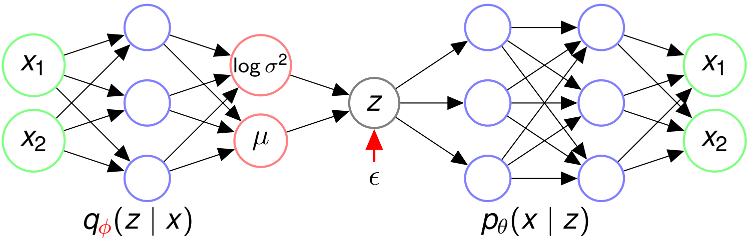
Given data points x, how can we model $p(x)$?

- Maximum likelihood approach:

$$\theta^* = \max_\theta \sum_i \log p(x^{(i)}; \theta)$$

  - Fit mean and variance (= parameters θ) of a distribution (e.g., Gaussian)
  - Fit parameters θ of a mixture distribution (e.g., mixture of Gaussian, k-means)
  - Use a variational auto-encoder

Variational auto-encoder:



Loss function:

$$\mathcal{L}(p_\theta, q_\phi) \approx -D_{KL}(q_\phi, p) + \frac{1}{N}\sum_{i=1}^{N} \log p_\theta(x|z^i)$$

Issue that we had to address:

Computing a normalization constant

Another approach:

Generative Adversarial Nets

Main idea:

Don't write a formula for $p(x)$, just learn to sample directly

Advantage:

No summation over large probability spaces

Formulate the problem as a game between two players:

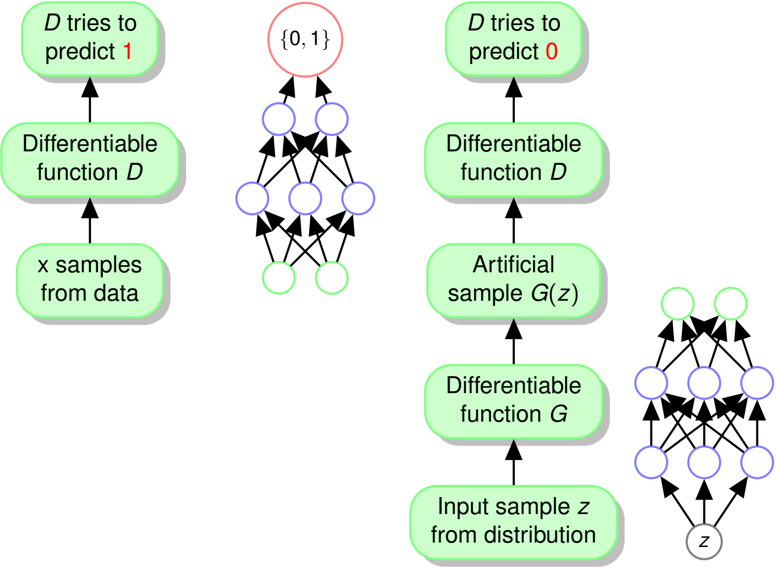
- Generator G
- Discriminator D

Task of the players

- G generates examples
- D predicts whether the example is artificial or real

G tries to "trick" D by generating samples that are hard to distinguish

Pictorially:



Mathematically:

- Generator $G_\theta(z)$
- Discriminator $D_w(x) = p(y = 1|x)$

How to choose w:

$$\min_w - \sum_x \log D_w(x) - \sum_z \log(1 - D_w(G_\theta(z)))$$

How to choose θ:

$$\max_\theta \min_w - \sum_x \log D_w(x) - \sum_z \log(1 - D_w(G_\theta(z)))$$

Generative adversarial nets:

$$\max_{\theta} \min_{w} -\sum_{x} \log D_w(x) - \sum_{z} \log(1 - D_w(G_\theta(z)))$$

How to optimize this theoretically?

Repeat until stopping criteria

1. Gradient step w.r.t. w
2. Gradient step w.r.t. θ

In practice:

Heuristics make this optimization more stable.

Analysis of generative adversarial nets:
What is the optimal discriminator assuming arbitrary capacity?

$$\begin{aligned}\min_D : \quad & -\int_x p_{\text{data}}(x) \log D(x) dx - \int_z p_z(z) \log(1 - D(G_\theta(z))) dz \\ &= -\int_x p_{\text{data}}(x) \log D(x) + p_G(x) \log(1 - D(x)) dx\end{aligned}$$

Euler-Lagrange formalism:

$$S(D) = \int_x L(x, D, \dot{D}) dx$$

Stationary D from

$$\frac{\partial L(x, D, \dot{D})}{\partial D} - \cancel{\frac{d}{dx} \frac{\partial L(x, D, \dot{D})}{\partial \dot{D}}} = 0$$

What is the optimal discriminator assuming arbitrary capacity?

$$\frac{\partial L(x, D, \dot{D})}{\partial D} = -\frac{p_{\text{data}}}{D} + \frac{p_G}{1 - D} = 0$$

Consequently:

$$D^*(x) = \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_G(x)}$$

Given the optimal discriminator, what is the optimal generator?

$$
\begin{aligned}
& -\int_x p_{\text{data}}(x)\log D^*(x) + p_G(x)\log(1-D^*(x))dx \\
= \quad & -\int_x p_{\text{data}}(x)\log\frac{p_{\text{data}}(x)}{p_{\text{data}}(x)+p_G(x)} + p_G(x)\log\frac{p_G(x)}{p_{\text{data}}(x)+p_G(x)}dx \\
= \quad & -2\,\text{JSD}(p_{\text{data}}, p_G) + \log(4)
\end{aligned}
$$

$$\text{JSD}(p_{\text{data}}, p_G) = \frac{1}{2}\text{KL}(p_{\text{data}}, M) + \frac{1}{2}\text{KL}(p_G, M) \quad \text{with} \quad M = \frac{1}{2}(p_{\text{data}} + p_G)$$

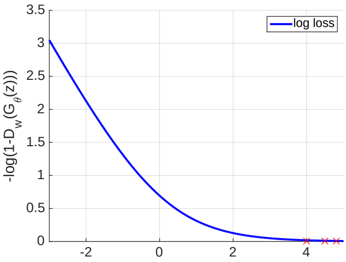
Consequently:

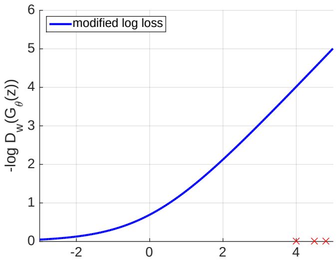
$$p_G(x) = p_{\text{data}}(x)$$

Some heuristics that improve optimization of:

$$\max_{\theta}\min_{w} -\sum_{x}\log D_w(x) - \sum_{z}\log(1 - D_w(G_\theta(z)))$$

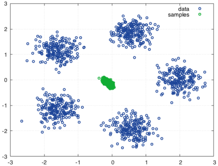
- If G is very bad and D is very good: almost no gradient
- Solve instead
$$\min_{\theta} -\sum_{z}\log D_w(G_\theta(z))$$
- Issue: no joint cost function for D and G



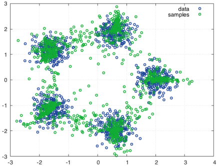


Plenty of additional impressive tricks.

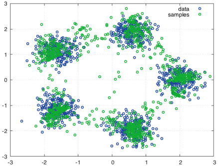
Some results on toy data:



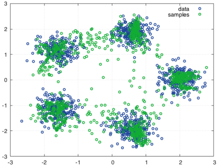
0 Iterations



2000 Iterations

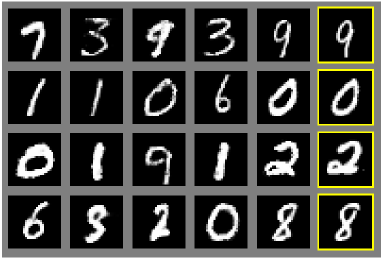


4000 Iterations



5000 Iterations

Application: modeling of images



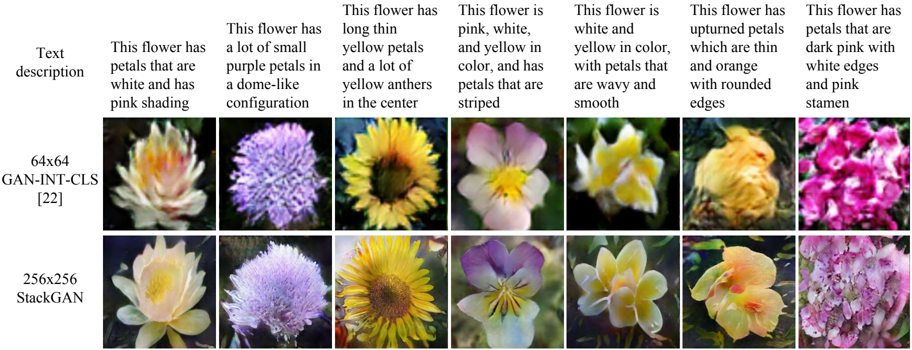


Figure 4. Example results by our proposed StackGAN and GAN-INT-CLS [22] conditioned on text descriptions from Oxford-102 test set.

Quiz:

- What generative modeling techniques do you know about?
- What are the short-comings of those techniques?
- What differentiates GANs from other generative models?
- What are the short-comings of GANs.

Important topics of this lecture

- Getting to know generative adversarial nets (GANs)
- Understanding their advantages and disadvantages

Next up:

Other image generation/completion techniques