Notes about GANs

A. G. Schwing

University of Illinois at Urbana-Champaign, 2017

Goals of this talk (adapted from ECE 547 lecture)

Goals of this talk (adapted from ECE 547 lecture)

- Getting to know Generative Adversarial Nets (GANs)

Goals of this talk (adapted from ECE 547 lecture)

- Getting to know Generative Adversarial Nets (GANs)
- Discussing some issues related to GANs

Goals of this talk (adapted from ECE 547 lecture)

- Getting to know Generative Adversarial Nets (GANs)
- Discussing some issues related to GANs
- Simplified proofs

Goals of this talk (adapted from ECE 547 lecture)

- Getting to know Generative Adversarial Nets (GANs)
- Discussing some issues related to GANs
- Simplified proofs
- Some research directions

Intuition

Classification:

Classification:

Which object is illustrated?

Classification:

Which object is illustrated?

- Car

Classification:

Which object is illustrated?

- Car
- Truck

Classification:

Which object is illustrated?

- Car
- Truck
- Recreational Vehicle

Classification:

Which object is illustrated?

- Car
- Truck
- Recreational Vehicle
- Ambulance truck

Classification:

Which object is illustrated?

- Car
- Truck
- Recreational Vehicle
- Ambulance truck
- Fire truck

Classification:

x: input data

Which object is illustrated?

- Car
- Truck
- Recreational Vehicle
- Ambulance truck
- Fire truck

Classification:

Which object is illustrated?

- Car
- Truck
- Recreational Vehicle
- Ambulance truck
- Fire truck

x: input data

$\boldsymbol{y}$: discrete output space

Classification:

Which object is illustrated?

- Car
- Truck
- Recreational Vehicle
- Ambulance truck
- Fire truck

x: input data

$\boldsymbol{y}$: discrete output space

Parametric ($\boldsymbol{w}$) score function:

$$F(\boldsymbol{y}, x, \boldsymbol{w})$$

Classification:

Which object is illustrated?

- Car
- Truck
- Recreational Vehicle
- Ambulance truck
- Fire truck

x: input data

$\boldsymbol{y}$: discrete output space

Parametric ($\boldsymbol{w}$) score function:

$$F(\boldsymbol{y}, x, \boldsymbol{w})$$

Model:

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

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

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

Why modeling a distribution $p(x)$?

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

Why modeling a distribution $p(x)$?

- Synthesis of plausible data (images, text)

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

Why modeling a distribution $p(x)$?

- Synthesis of plausible data (images, text)
- Environment simulator (reinforcement learning, planning)

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

Why modeling a distribution $p(x)$?

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

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

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

- Maximum likelihood approach:

$$\boldsymbol{w}^* = \max_{\boldsymbol{w}} \sum_i \log p_{\boldsymbol{w}}(x^{(i)})$$

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

- Maximum likelihood approach:

$$\boldsymbol{w}^* = \max_{\boldsymbol{w}} \sum_i \log p_{\boldsymbol{w}}(x^{(i)})$$

  - Fit mean and variance (= parameters $\boldsymbol{w}$) of a distribution (e.g., Gaussian)

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

- Maximum likelihood approach:

$$\boldsymbol{w}^* = \max_{\boldsymbol{w}} \sum_i \log p_{\boldsymbol{w}}(x^{(i)})$$

  - Fit mean and variance (= parameters $\boldsymbol{w}$) of a distribution (e.g., Gaussian)
  - Fit parameters $\boldsymbol{w}$ of a mixture distribution

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

- Maximum likelihood approach:

$$\boldsymbol{w}^* = \max_{\boldsymbol{w}} \sum_i \log p_{\boldsymbol{w}}(x^{(i)})$$

  - Fit mean and variance (= parameters $\boldsymbol{w}$) of a distribution (e.g., Gaussian)
  - Fit parameters $\boldsymbol{w}$ of a mixture distribution
  - Use a variational auto-encoder

Another approach:

Another approach:

Generative Adversarial Nets

Main idea:

Don't write a formula for $p_{\boldsymbol{w}}(x)$, just learn to sample directly

Main idea:

Don't write a formula for $p_{\boldsymbol{w}}(x)$, just learn to sample directly

Advantage:

Main idea:

Don't write a formula for $p_{\boldsymbol{w}}(x)$, just learn to sample directly

Advantage:

No summation over large probability spaces

Formulate the problem as a game between two players:

Formulate the problem as a game between two players:

- Generator G

Formulate the problem as a game between two players:

- Generator G
- Discriminator D

Formulate the problem as a game between two players:

- Generator G
- Discriminator D

Task of the players

Formulate the problem as a game between two players:

- Generator G
- Discriminator D

Task of the players

- G generates examples

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

Formulate the problem as a game between two players:

- Generator G
- Discriminator D

Task of the players

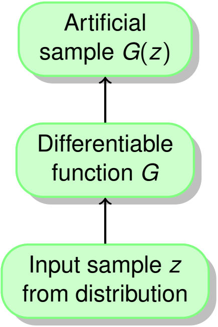
- G generates examples
- D predicts whether the example is artificial or real

Goal:

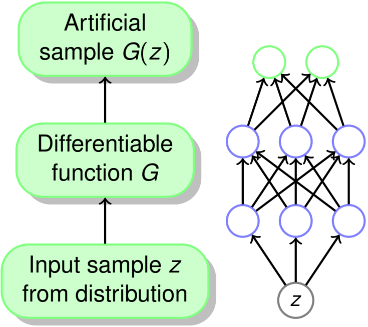
G tries to “trick” D by generating samples that are hard to distinguish

Pictorially:

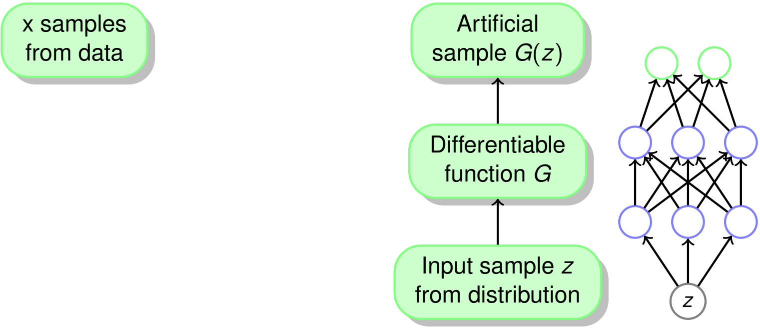
Pictorially:



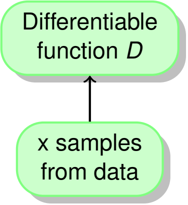
Pictorially:

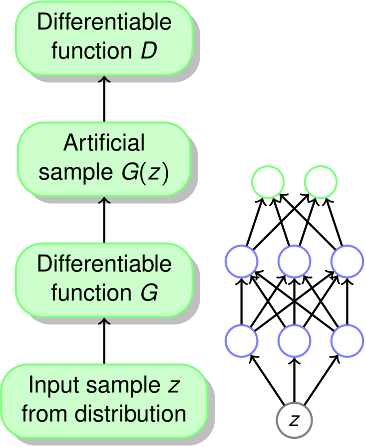


Pictorially:

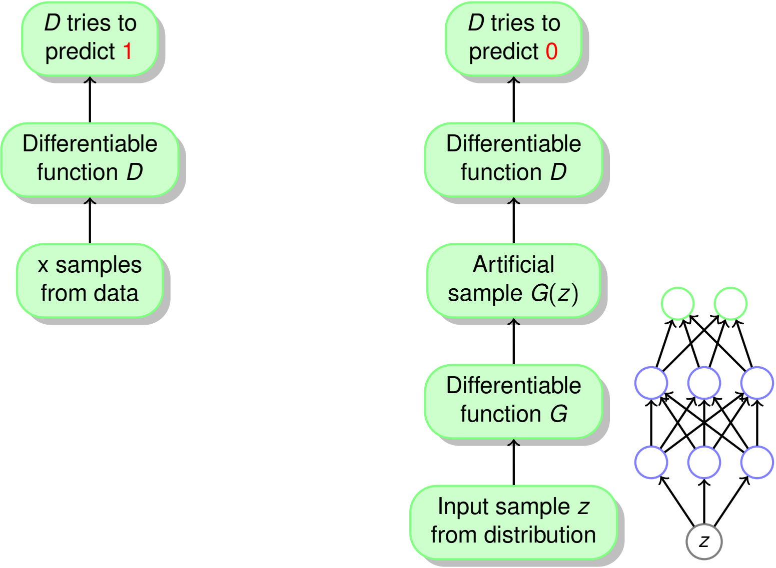


Pictorially:

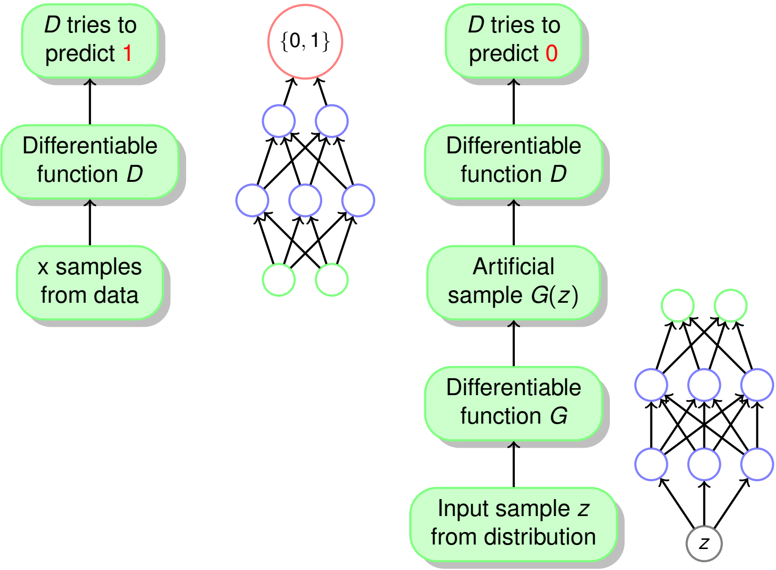




Pictorially:



Pictorially:



Mathematically:

Mathematically:

- Generator $G_\theta(z)$

Mathematically:

- Generator $G_\theta(z)$
- Discriminator $D_w(x) = p_w(y = 1|x)$ (recall general definition)

Mathematically:

- Generator $G_\theta(z)$
- Discriminator $D_w(x) = p_w(y = 1|x)$ (recall general definition)

How to choose w:

Mathematically:

- Generator $G_\theta(z)$
- Discriminator $D_w(x) = p_w(y = 1|x)$ (recall general definition)

How to choose w:

$$\min_w - \sum_x \log D_w(x)$$

Mathematically:

- Generator $G_\theta(z)$
- Discriminator $D_w(x) = p_w(y = 1|x)$ (recall general definition)

How to choose w:

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

Mathematically:

- Generator $G_\theta(z)$
- Discriminator $D_w(x) = p_w(y = 1|x)$ (recall general definition)

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:

Mathematically:

- Generator $G_\theta(z)$
- Discriminator $D_w(x) = p_w(y = 1|x)$ (recall general definition)

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:

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

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?

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

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

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

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:

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.

1. heuristic that improves optimization of:

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

1. heuristic that improves optimization of:

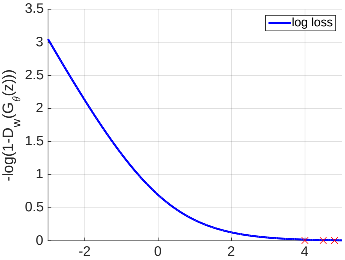
$$\max_{\theta} \min_{w} -\sum_{x} \log D_w(x) - \sum_{z} \log(1 - D_w(G_\theta(z)))$$

- If G is very bad and D is very good:

1. heuristic that improves optimization of:

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

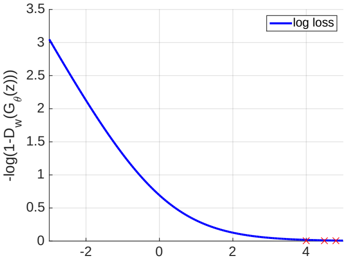
- If G is very bad and D is very good:



1. heuristic that improves 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

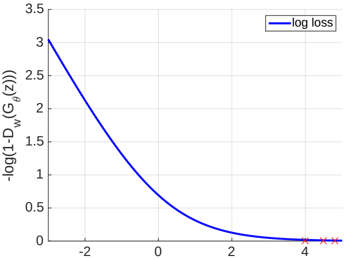


1. heuristic that improves optimization of:

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

- If G is very bad and D is very good: almost no gradient
- Solve instead

$$\min_{\theta} -\sum_{z} \log D_w(G_\theta(z))$$

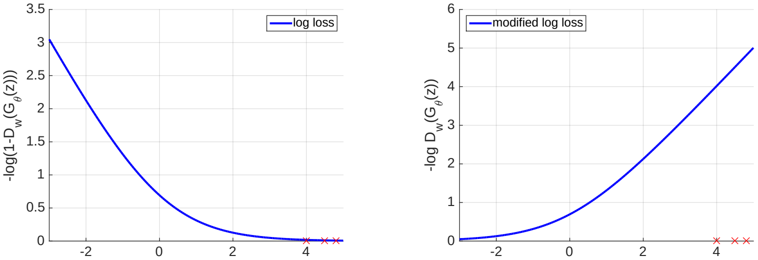


1. heuristic that improves optimization of:

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

- If G is very bad and D is very good: almost no gradient
- Solve instead

$$\min_\theta -\sum_z \log D_w(G_\theta(z))$$



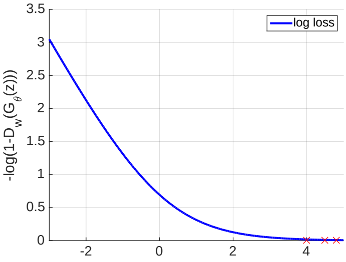
1. heuristic that improves optimization of:

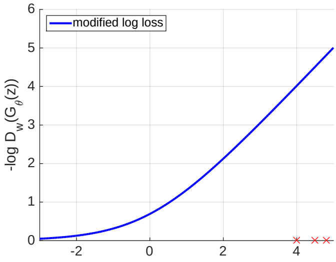
$$\max_{\theta} \min_{w} -\sum_{x} \log D_w(x) - \sum_{z} \log(1 - D_w(G_\theta(z)))$$

- 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





2. heuristic: restrict complexity of discriminator

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

2. heuristic: restrict complexity of discriminator

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

Assume $D_w(x) = p_w(y = 1|x)$ to be log-linear:

2. heuristic: restrict complexity of discriminator

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

Assume $D_w(x) = p_w(y = 1|x)$ to be log-linear:

$$D_w(x) = p_w(y = 1|x) = \frac{\exp w^\top x}{1 + \exp w^\top x}$$

2. heuristic: restrict complexity of discriminator

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

Assume $D_w(x) = p_w(y = 1|x)$ to be log-linear:

$$D_w(x) = p_w(y = 1|x) = \frac{\exp w^\top x}{1 + \exp w^\top x}$$

$$\max_\theta \min_w - \sum_x \left(w^T x - \log(1 + \exp w^T x) \right) +$$

2. heuristic: restrict complexity of discriminator

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

Assume $D_w(x) = p_w(y = 1|x)$ to be log-linear:

$$D_w(x) = p_w(y = 1|x) = \frac{\exp w^\top x}{1 + \exp w^\top x}$$

$$\max_{\theta} \min_{w} -\sum_{x} \left(w^T x - \log(1 + \exp w^T x)\right) + \sum_{z} \log(1 + \exp w^T G_\theta(z))$$

$$\max_{\theta} \min_{w} \frac{C}{2}\|w\|_2^2 - \sum_{x} \left(w^T x - \log(1 + \exp w^T x) \right) + \sum_{z} \log(1 + \exp w^T G_\theta(z))$$

$$\max_\theta \min_w \frac{C}{2}\|w\|_2^2 - \sum_x \left(w^T x - \log(1+\exp w^T x)\right) + \sum_z \log(1+\exp w^T G_\theta(z))$$

Why is this useful?

$$\max_{\theta} \min_{w} \frac{C}{2}\|w\|_2^2 - \sum_x \left(w^T x - \log(1 + \exp w^T x)\right) + \sum_z \log(1 + \exp w^T G_\theta(z))$$

Why is this useful?

- Convex in w

$$\max_{\theta} \min_{w} \frac{C}{2}\|w\|_2^2 - \sum_{x} \left(w^T x - \log(1 + \exp w^T x) \right) + \sum_{z} \log(1 + \exp w^T G_\theta(z))$$

Why is this useful?

- Convex in w
- We can compute its dual program

$$\max_\theta \min_w \frac{C}{2}\|w\|_2^2 - \sum_x \left(w^T x - \log(1+\exp w^T x)\right) + \sum_z \log(1+\exp w^T G_\theta(z))$$

Why is this useful?

- Convex in w
- We can compute its dual program
- We obtain a $\max_\theta \max_\lambda$ task

$$\max_\theta \min_w \frac{C}{2}\|w\|_2^2 - \sum_x \left(w^T x - \log(1+\exp w^T x)\right) + \sum_z \log(1+\exp w^T G_\theta(z))$$

Why is this useful?

- Convex in w
- We can compute its dual program
- We obtain a $\max_\theta \max_\lambda$ task
- Good empirical results

$$\max_\theta \min_w \frac{C}{2}\|w\|_2^2 - \sum_x \left(w^T x - \log(1+\exp w^T x)\right) + \sum_z \log(1+\exp w^T G_\theta(z))$$

Why is this useful?

- Convex in w
- We can compute its dual program
- We obtain a $\max_\theta \max_\lambda$ task
- Good empirical results
- Additional insights

Primal:

$$\max_{\theta} \min_{w} \frac{C}{2}\|w\|_2^2 - \sum_{x} \left(w^T x - \log(1 + \exp w^T x) \right) + \sum_{z} \log(1 + \exp w^T G_\theta(z))$$

Dual:

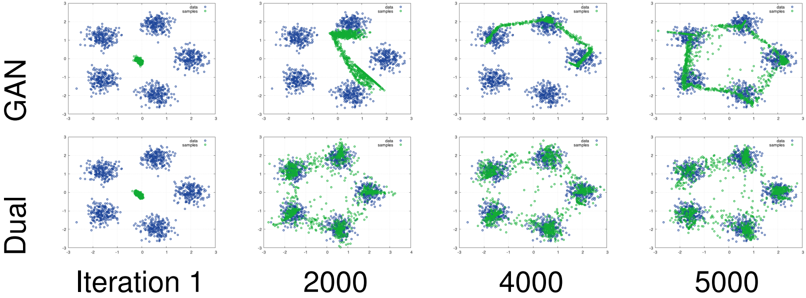
Primal:

$$\max_{\theta} \min_{w} \frac{C}{2}\|w\|_2^2 - \sum_{x}\left(w^T x - \log(1+\exp w^T x)\right) + \sum_{z}\log(1+\exp w^T G_\theta(z))$$

Dual:

$$\begin{array}{ll} \max\limits_{\theta,\lambda_x,\lambda_z} & \frac{-1}{2C}\left\|\sum\limits_{\boldsymbol{x}}(1-\lambda_x)\boldsymbol{x} - \sum\limits_{\boldsymbol{z}}\lambda_z G_\theta(\boldsymbol{z})\right\|_2^2 + \sum\limits_{\boldsymbol{x}} H(\lambda_x) + \sum\limits_{\boldsymbol{z}} H(\lambda_z) \\ \text{s.t.} & \begin{array}{ll} \forall \boldsymbol{x} & 0 \le \lambda_x \le 1 \\ \forall \boldsymbol{z} & 0 \le \lambda_z \le 1 \end{array} \end{array}$$

Some results on toy data:



Plenty of additional impressive tricks.

Analysis of generative adversarial nets:

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

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

$$\min_D : \quad -\int_x p_{\text{data}}(x) \log D(x) dx - \int_z p_z(z) \log(1 - D(G_\theta(z))) dz$$

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

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

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:

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:

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

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

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

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

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

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

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?

$$\frac{\partial L(x, D, \dot{D})}{\partial D} =$$

What is the optimal discriminator assuming arbitrary capacity?

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

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:

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?

$$-\int_x p_{\text{data}}(x)\log D^*(x) + p_G(x)\log(1 - D^*(x))dx$$

$$=$$

$$=$$

Given the optimal discriminator, what is the optimal generator?

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

Given the optimal discriminator, what is the optimal generator?

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

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

Given the optimal discriminator, what is the optimal generator?

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

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

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:

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:

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

Recall:

GANs optimize Jensen-Shannon divergence

Recall:

GANs optimize Jensen-Shannon divergence

How about other divergences/distances?

Wasserstein GAN:

Wasserstein GAN:

Other ways to measure distance between two distributions:

Wasserstein GAN:

Other ways to measure distance between two distributions:
Wasserstein distance:

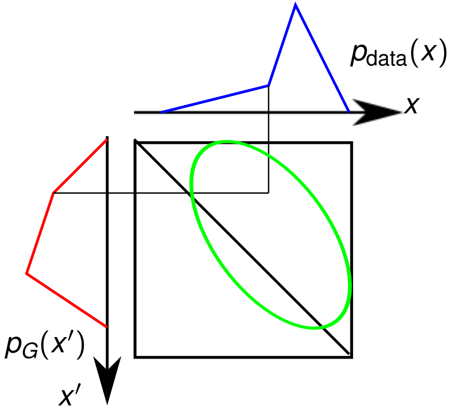
$$W(p_{\text{data}}, p_G) = \min_{p_J(x,x') \in \Pi(p_{\text{data}}, p_G)} \mathbb{E}_{p_J}[\|x - x'\|]$$

Wasserstein GAN:

Other ways to measure distance between two distributions:
Wasserstein distance:

$$W(p_{\text{data}}, p_G) = \min_{p_J(x,x')\in\Pi(p_{\text{data}},p_G)} \mathbb{E}_{p_J}[\|x - x'\|]$$

Pictorially:



How to formulate GANs using Wasserstein distance?

$$\min_{p_G} W(p_{\text{data}}, p_G) = \min_{p_G} \min_{p_J(x,x')\in\Pi(p_{\text{data}},p_G)} \mathbb{E}_{p_J}[\|x - x'\|]$$

How to formulate GANs using Wasserstein distance?

$$\min_{p_G} W(p_{\text{data}}, p_G) = \min_{p_G} \min_{p_J(x,x')\in\Pi(p_{\text{data}},p_G)} \mathbb{E}_{p_J}[\|x - x'\|]$$

What's the issue?

How to formulate GANs using Wasserstein distance?

$$\min_{p_G} W(p_{\text{data}}, p_G) = \min_{p_G} \min_{p_J(x,x')\in\Pi(p_{\text{data}},p_G)} \mathbb{E}_{p_J}[\|x - x'\|]$$

What's the issue?

- p_{data} not available, only samples

How to formulate GANs using Wasserstein distance?

$$\min_{p_G} W(p_{\text{data}}, p_G) = \min_{p_G} \min_{p_J(x,x') \in \Pi(p_{\text{data}}, p_G)} \mathbb{E}_{p_J}[\|x - x'\|]$$

What's the issue?

- p_{data} not available, only samples
- How to compute this joint distribution p_J?

How to formulate GANs using Wasserstein distance?

$$\min_{p_G} W(p_{\text{data}}, p_G) = \min_{p_G} \min_{p_J(x,x')\in\Pi(p_{\text{data}},p_G)} \mathbb{E}_{p_J}[\|x - x'\|]$$

What's the issue?

- p_{data} not available, only samples
- How to compute this joint distribution p_J?

Proposed solution:

How to formulate GANs using Wasserstein distance?

$$\min_{p_G} W(p_{\text{data}}, p_G) = \min_{p_G} \min_{p_J(x,x') \in \Pi(p_{\text{data}}, p_G)} \mathbb{E}_{p_J}[\|x - x'\|]$$

What's the issue?

- p_{data} not available, only samples
- How to compute this joint distribution p_J?

Proposed solution: Kantorovich-Rubinstein duality

How to formulate GANs using Wasserstein distance?

$$\min_{p_G} W(p_{\text{data}}, p_G) = \min_{p_G} \min_{p_J(x,x') \in \Pi(p_{\text{data}}, p_G)} \mathbb{E}_{p_J}[\|x - x'\|]$$

What's the issue?

- p_{data} not available, only samples
- How to compute this joint distribution p_J?

Proposed solution: Kantorovich-Rubinstein duality

$$W(p_{\text{data}}, p_G) = \max_{\|f\|_L \leq 1} \mathbb{E}_{p_{\text{data}}}[f(x)] - \mathbb{E}_{p_G}[f(x')]$$

How to formulate GANs using Wasserstein distance?

$$\min_{p_G} W(p_{\text{data}}, p_G) = \min_{p_G} \min_{p_J(x,x') \in \Pi(p_{\text{data}}, p_G)} \mathbb{E}_{p_J}[\|x - x'\|]$$

What's the issue?

- p_{data} not available, only samples
- How to compute this joint distribution p_J?

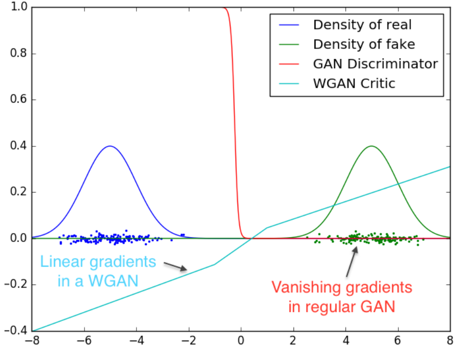
Proposed solution: Kantorovich-Rubinstein duality

$$W(p_{\text{data}}, p_G) = \max_{\|f\|_L \leq 1} \mathbb{E}_{p_{\text{data}}}[f(x)] - \mathbb{E}_{p_G}[f(x')]$$

$$\min_{p_G} \max_{w \in \mathcal{W}} \mathbb{E}_{p_{\text{data}}}[f_w(x)] - \mathbb{E}_{p_G}[f_w(x')]$$

Results:

Results:



1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−8
−6
−4
−2
0
2
4
6
8
Density of real
Density of fake
GAN Discriminator
WGAN Critic
Linear gradients in a WGAN
Vanishing gradients in regular GAN

Wasserstein GAN objective:

$$\min_{p_G} \max_{w \in \mathcal{W}} \mathbb{E}_{p_{\text{data}}}[f_w(x)] - \mathbb{E}_{p_G}[f_w(x')]$$

Wasserstein GAN objective:

$$\min_{p_G} \max_{w \in \mathcal{W}} \mathbb{E}_{p_{\text{data}}}[f_w(x)] - \mathbb{E}_{p_G}[f_w(x')]$$

Linear GAN dual:

$$\begin{array}{ll} \max\limits_{\theta,\lambda_x,\lambda_z} & \frac{-1}{2C}\left\|\sum_{\boldsymbol{x}}(1-\lambda_x)\boldsymbol{x} - \sum_{\boldsymbol{z}}\lambda_z G_\theta(\boldsymbol{z})\right\|_2^2 + \sum_{\boldsymbol{x}} H(\lambda_x) + \sum_{\boldsymbol{z}} H(\lambda_z) \\ \text{s.t.} & \begin{array}{ll} \forall \boldsymbol{x} & 0 \leq \lambda_x \leq 1 \\ \forall \boldsymbol{z} & 0 \leq \lambda_z \leq 1 \end{array}, \end{array}$$

Wasserstein GAN objective:

$$\min_{p_G} \max_{w \in \mathcal{W}} \mathbb{E}_{p_{\text{data}}}[f_w(x)] - \mathbb{E}_{p_G}[f_w(x')]$$

Linear GAN dual:

$$\begin{array}{ll} \displaystyle\max_{\theta, \lambda_x, \lambda_z} & \displaystyle\frac{-1}{2C} \left\| \sum_{\boldsymbol{x}} (1 - \lambda_x)\boldsymbol{x} - \sum_{\boldsymbol{z}} \lambda_z G_\theta(\boldsymbol{z}) \right\|_2^2 + \sum_{\boldsymbol{x}} H(\lambda_x) + \sum_{\boldsymbol{z}} H(\lambda_z) \\ \text{s.t.} & \begin{array}{ll} \forall \boldsymbol{x} & 0 \leq \lambda_x \leq 1 \\ \forall \boldsymbol{z} & 0 \leq \lambda_z \leq 1 \end{array}, \end{array}$$

Maximum mean discrepancy: [Gretton et al.]

Wasserstein GAN objective:

$$\min_{p_G} \max_{w \in \mathcal{W}} \mathbb{E}_{p_{\text{data}}}[f_w(x)] - \mathbb{E}_{p_G}[f_w(x')]$$

Linear GAN dual:

$$\begin{array}{ll} \max\limits_{\theta,\lambda_x,\lambda_z} & \dfrac{-1}{2C}\left\|\sum\limits_{\boldsymbol{x}}(1-\lambda_x)\boldsymbol{x} - \sum\limits_{\boldsymbol{z}}\lambda_z G_\theta(\boldsymbol{z})\right\|_2^2 + \sum\limits_{\boldsymbol{x}} H(\lambda_x) + \sum\limits_{\boldsymbol{z}} H(\lambda_z) \\ \text{s.t.} & \begin{array}{ll} \forall \boldsymbol{x} & 0 \le \lambda_x \le 1 \\ \forall \boldsymbol{z} & 0 \le \lambda_z \le 1 \end{array}, \end{array}$$

Maximum mean discrepancy: [Gretton et al.]

$$MMD(x, G_\theta(z)) := \max_f \left(\mathbb{E}_x[f(x)] - \mathbb{E}_z[G_\theta(z)]\right)$$

Other ways to optimize the Wasserstein distance:

$$W(p_{\text{data}}, p_G) = \min_{p_J(x,x') \in \Pi(p_{\text{data}}, p_G)} \mathbb{E}_{p_J}[\|x - x'\|]$$

Other ways to optimize the Wasserstein distance:

$$W(p_{\text{data}}, p_G) = \min_{p_J(x,x') \in \Pi(p_{\text{data}}, p_G)} \mathbb{E}_{p_J}[\|x - x'\|]$$

Sliced Wasserstein distance: [Rabin et al. (ECCV 2010); Texture Mixing]

Other ways to optimize the Wasserstein distance:

$$W(p_{\text{data}}, p_G) = \min_{p_J(x,x') \in \Pi(p_{\text{data}}, p_G)} \mathbb{E}_{p_J}[\|x - x'\|]$$

Sliced Wasserstein distance: [Rabin et al. (ECCV 2010); Texture Mixing]

$$\tilde{W}(X, X')^2 = \int_{w \in \Omega} \min_{\sigma_w} \sum_i |(X_i - X'_{\sigma_w(i)})^T w|^2 dw$$

with permutation: σ_w

Discussion