Generative Adversarial Networks and Wasserstein Loss

SE Kritische Forschungsanalyse: Deep Learning & Inverse Probleme

Christoph Angermann

https://applied-math.uibk.ac.at/

Motivation

Two main approaches of deep learning:

Motivation

Two main approaches of deep learning:

- **discriminative models**
  - model on the conditional probability of target Y given observation x of variable $X \Rightarrow \mathbb{P}(Y \mid X = x)$
  - map high-dimensional, rich sensory data to class label
  - VGG-nets

Motivation

Two main approaches of deep learning:

- **discriminative models**
  - model on the conditional probability of target Y given observation x of variable $X \Rightarrow \mathbb{P}(Y \mid X = x)$
  - map high-dimensional, rich sensory data to class label
  - VGG-nets
- **generative models**
  - learn unknown distribution of data set to generate new data with variations

Motivation

What does it mean to learn a probability distribution?

- Classical answer: learn a probability density by defining parametric family of densities $(P_\theta)_{\theta \in \mathbb{R}^d}$ and maximize likelihood on data $\{x_i\}_{i=1}^m$:

$$\max_{\theta \in \mathbb{R}^d} \frac{1}{m} \sum_{i=1}^{m} \log P_\theta(x_i)$$

- If real data distribution $\mathbb{P}_r$ admits density and $\mathbb{P}_\theta$ is distribution of parametrized density P_θ, this amounts to minimizing

$$D_{KL}(\mathbb{P}_r \,||\, \mathbb{P}_\theta) = \int_x \log\left(\frac{P_r(x)}{P_\theta(x)}\right) P_r(x) dx$$

Issues:

- Need the model density P_θ to exist.
- Computationally difficult to generate samples given arbitrary high dimensional density.

Motivation

Solution:

- Define RV $Z \sim \mathbb{P}_z$ and pass it through a parametric function $G_\theta : \mathcal{Z} \to \mathcal{X}$ that directly generates samples following certain distribution $\mathbb{P}_\theta$.
- Varying θ can make generated distribution closer to $\mathbb{P}_r$.
- Easy generation of samples is often more useful than knowing the density (e.g. superresolution, segmentation)
- Well-known examples: Variational Auto-Encoders (VAEs), Generative Adversarial Networks (GANs)

Generative adversarial networks

In 2014, John Goodfellow leveraged the idea of highly-developed discriminative models to overcome approximation difficulties of generative ones

[1] **Yann LeCunn**, research director Facebook AI, **Turing Award Recipient 2018**

Generative adversarial networks

In 2014, John Goodfellow leveraged the idea of highly-developed discriminative models to overcome approximation difficulties of generative ones
$\Rightarrow$ generative adversarial networks (**GANs**)

- algorithmic architecture consisting of 2 neural networks $G_{\theta_g} : \mathcal{Z} \rightarrow \mathcal{X}$ and $D_{\theta_d} : \mathcal{X} \rightarrow [0, 1]$, where $\mathcal{X}, \mathcal{Z}$ denote data space and d-dimensional latent space, respectively
- $\mathbb{P}_r \triangleq$ distribution over real data space $\mathcal{X}$
- $\mathbb{P}_{\theta_g} \triangleq$ distribution over $\{G_{\theta_g}(z),\ z \in \mathcal{Z}\}$
- G_{θ_g} generates samples following generator distribution $\mathbb{P}_{\theta_g}$.
- discriminator D estimates probability that realisation of sample X came from real data ($X \sim \mathbb{P}_r$) rather than from G_{θ_g} ($X \sim \mathbb{P}_{\theta_g}$).
- G_{θ_g} pitted against discriminator D_{θ_d} to generate new synthetic instances

[1]**Yann LeCunn**, research director Facebook AI, **Turing Award Recipient 2018**

Generative adversarial networks

In 2014, John Goodfellow leveraged the idea of highly-developed discriminative models to overcome approximation difficulties of generative ones
$\Rightarrow$ generative adversarial networks (**GANs**)

- algorithmic architecture consisting of 2 neural networks $G_{\theta_g} : \mathcal{Z} \to \mathcal{X}$ and $D_{\theta_d} : \mathcal{X} \to [0, 1]$, where $\mathcal{X}, \mathcal{Z}$ denote data space and d-dimensional latent space, respectively
- $\mathbb{P}_r \triangleq$ distribution over real data space $\mathcal{X}$
- $\mathbb{P}_{\theta_g} \triangleq$ distribution over $\{G_{\theta_g}(z),\ z \in \mathcal{Z}\}$
- G_{θ_g} generates samples following generator distribution $\mathbb{P}_{\theta_g}$.
- discriminator D estimates probability that realisation of sample X came from real data ($X \sim \mathbb{P}_r$) rather than from G_{θ_g} ($X \sim \mathbb{P}_{\theta_g}$).
- G_{θ_g} pitted against discriminator D_{θ_d} to generate new synthetic instances

"*...the most interesting idea in the last 10 years in machine learning*"[1].

[1]**Yann LeCunn**, research director Facebook AI, **Turing Award Recipient 2018**

Generative adversarial networks

- minimax two-player game
- train G_{θ_g} to fool a steadily improving discriminator D_{θ_d}
- train D_{θ_d} to maximize probability of assigning correct labels to samples drawn from $\mathbb{P}_{\theta_g}$ and $\mathbb{P}_r$ respectively
- train G_{θ_g} to minimize $\log\big(1 - D_{\theta_d}(G_{\theta_g}(z))\big)$, $z \in \mathcal{Z}$
- **full objective:**

$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x\sim\mathbb{P}_r}\big[\log D_{\theta_d}(x)\big] + \mathbb{E}_{z\sim\mathcal{N}(0,I_d)}\big[\log(1 - D_{\theta_d}(G_{\theta_g}(z)))\big] \right]$$

- **algorithm:** alternate between k steps of optimizing D_{θ_d} and and one step of optimizing G_{θ_g}

Generative adversarial networks

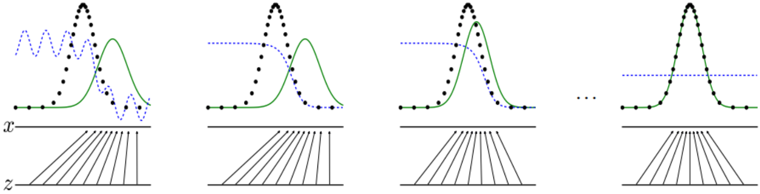


Figure: GANs are trained by iteratively updating the discriminative distribution (blue, dashed) to discriminate between samples from P_X (black, dotted) and from those drawn by the generative distribution P_G (green, solid).

GAN loss

What does the loss function represent?

- We have a well-defined GAN loss function

$$L(G, D) = \int_x \Big(P_r(x) \log(D(x)) + P_\theta(x) log(1 - D(x)) \Big) dx$$

Proposition

For G fixed and corresponding generator distribution $\mathbb{P}_\theta$, the optimal discriminator D is

$$D^*(x) = \frac{P_r(x)}{P_r(x) + P_\theta(x)}$$

GAN loss

Proof.

Let $\tilde{x} \triangleq D(x)$ and $f(\tilde{x}) \triangleq P_r(x) \log \tilde{x} + P_\theta(x) \log(1 - \tilde{x})$. Then:

$$\frac{df(\tilde{x})}{d\tilde{x}} = P_r(x)\frac{1}{\tilde{x}} - P_\theta(x)\frac{1}{1-\tilde{x}} =$$

$$= \frac{P_r(x) - (P_r(x) + P_\theta(x))\tilde{x}}{\tilde{x}(1-\tilde{x})}$$

$$\implies D^*(x) = \frac{P_r(x)}{P_r(x) + P_\theta(x)}$$

□

GAN loss

What does the loss function represent?

For optimal D, we obtain

$$
\begin{aligned}
D_{JS}(\mathbb{P}_r||\mathbb{P}_\theta) &= \frac{1}{2}D_{KL}(\mathbb{P}_r||(\mathbb{P}_r+\mathbb{P}_\theta)/2) + \frac{1}{2}D_{KL}(\mathbb{P}_\theta||(\mathbb{P}_r+\mathbb{P}_\theta)/2) = \\
&= \frac{1}{2}\Bigg[\log 2 + \int_x P_r(x)\log\frac{P_r(x)}{P_r(x)+P_\theta(x)}dx + \\
&\quad + \log 2 + \int_x P_\theta(x)\log\frac{P_\theta(x)}{P_r(x)+P_\theta(x)}dx\Bigg] = \\
&= \frac{1}{2}\big(\log 4 + L(G, D^*)\big) \\
&\Rightarrow L(G, D^*) = 2\cdot D_{JS}(\mathbb{P}_r||\mathbb{P}_\theta) - 2\log 2
\end{aligned}
$$

Therefore, for optimal discriminator D^* the GAN loss quantifies distance between $\mathbb{P}_r$ and $\mathbb{P}_\theta$ by the *Jensen-Shannon* divergence.

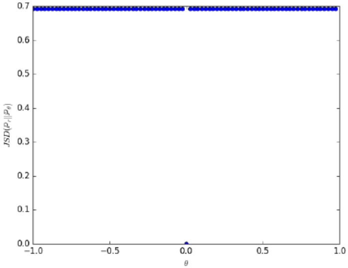
Problems with GANs

- **Hard to achieve Nash equilibrium:**
  Nash equilibrium...solution of a non-cooperative game involving two concurrently players.
  Each model updates its cost independently with no respect to the other one $\Rightarrow$ updating models' gradients concurrently cannot guarantee a convergence.
- **Vanishing gradient:**
  In case of perfect discriminator, i.e. $D(x) = 1$ for all x following $\mathbb{P}_r$ and $D(G(z)) = 0$ for $z \in \mathcal{Z}$, loss function falls to zero $\Rightarrow$ no gradient for update.
  Therefore, GAN faces a dilemma:
  1. If the discriminator behaves badly, no valuable updates for the generator are obtained.
  2. If the discriminator is almost perfect, gradient of loss function drops down and training becomes super slow or stuck.

Problems with GANs

- **Low dimensional supports:**
  Because $\mathbb{P}_r$ and $\mathbb{P}_\theta$ rest in low-dimensional manifolds, they are almost gonna be disjoint $\Rightarrow$ Kullback-Leibler divergence returns infinity.
  Example: Let $Z \sim \mathcal{U}[0,1]$, $\mathbb{P}_r$ the distribution of $(0, Z) \in \mathbb{R}^2$ and let $G_\theta(z) \triangleq (\theta, z)$. Then:

$$D_{KL}(\mathbb{P}_\theta || \mathbb{P}_0) = \begin{cases} \infty, & \text{if } \theta \neq 0 \\ 0, & \text{if } \theta = 0 \end{cases}, \qquad D_{JS}(\mathbb{P}_\theta || \mathbb{P}_0) = \begin{cases} \log 2, & \text{if } \theta \neq 0 \\ 0, & \text{if } \theta = 0 \end{cases}$$



Requirements for loss ρ

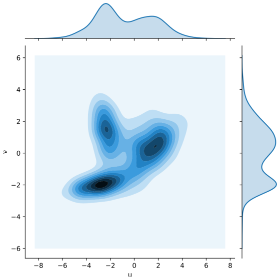
- $\theta \mapsto \rho(\mathbb{P}_\theta, \mathbb{P}_r)$ continuous
- no vanishing gradients
- high reliable generator updates

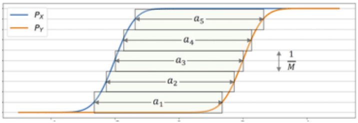
Wasserstein-1 distance

The Wasserstein-1 distance (Earth mover distance) is defined as

$$W_1(\mathbb{P}_1, \mathbb{P}_2) = \inf_{J \in \mathcal{J}(\mathbb{P}_1, \mathbb{P}_2)} \mathbb{E}_{(x,y)\sim J} \|x - y\|,$$

where $\mathbb{P}_1, \mathbb{P}_2$ are the considered distributions and $\mathcal{J}(\mathbb{P}_1, \mathbb{P}_2)$ the set of all joint distributions with marginals $\mathbb{P}_1$ and $\mathbb{P}_2$. Can also be formulated in the setting of a optimal mass transport problem, where one aims to find a transference plan, that transports a unit mass from one point to another, as cheap as possible regarding a given cost function.





Wasserstein-1 distance

Example: Let $Z \sim \mathcal{U}[0,1]$, $\mathbb{P}_r$ the distribution of $(0, Z) \in \mathbb{R}^2$ and let $G_\theta(z) \triangleq (\theta, z)$. Then:

$$W_1(\mathbb{P}_r, \mathbb{P}_\theta) = |\theta|, \qquad D_{JS}(\mathbb{P}_\theta || \mathbb{P}_0) = \begin{cases} \log 2, & \text{if } \theta \neq 0 \\ 0, & \text{if } \theta = 0 \end{cases}$$

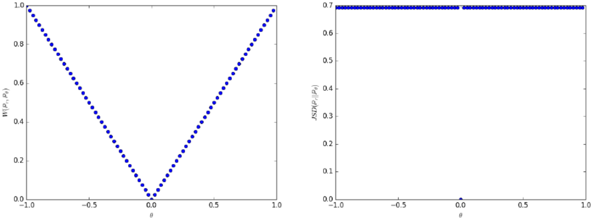


Figure: Earth mover distance is continuous and provides a usable gradient everywhere contrary to Jensen-Shannon divergence.

Wasserstein-1 distance

Theorem

Let $\mathbb{P}_r$ be a fixed distribution over $\mathcal{X}$. Let Z be a RV over another space $\mathcal{Z}$. Let $G : \mathcal{Z} \times \mathbb{R}^d \to \mathcal{X}$ be a function, that will be denoted $G_\theta(z)$ with z the first coordinate and θ the second. Let $\mathbb{P}_\theta$ denote the distribution of $G_\theta(z)$. Then,

1. *If G is continuous in θ, so is $W_1(\mathbb{P}_r, \mathbb{P}_\theta)$.*
2. *If G is locally Lipschitz and satisfies* ***regularity assumption*** *$\mathbb{E}_z L(\theta, z) < \infty$, then $W_1(\mathbb{P}_r, \mathbb{P}_\theta)$ is continuous everywhere and differentiable almost everywhere.*
3. *Statements 1-2 are false for Jensen-Shannon divergence $D_{JS}(\mathbb{P}_r, \mathbb{P}_\theta)$.*

Wasserstein-1 distance

Proof.

Let $\theta, \theta' \in \mathbb{R}^d$ and γ denote the distribution of coupling $(G_\theta(Z), G_{\theta'}(Z))$. Then, $\gamma \in \mathcal{J}(\mathbb{P}_\theta, \mathbb{P}_{\theta'})$ and

$$W_1(\mathbb{P}_\theta, \mathbb{P}_{\theta'}) \leqslant \int_{\mathcal{X}\times\mathcal{X}} \|x - y\| \, d\gamma = \mathbb{E}_{(x,y)\sim\gamma} \|x - y\| = \mathbb{E}_z \|G_\theta(z) - G_{\theta'}(z)\|$$

G is continuous in $\theta \Rightarrow G_\theta(z) \rightarrow_{\theta\rightarrow\theta'} G_{\theta'}(z)$. Furthermore, $\mathcal{X}$ is compact $\Rightarrow \|G_\theta(z) - G_{\theta'}(z)\| \leqslant M$ for some constant M and all θ and all z. Due to the dominated convergence theorem

$$W_1(\mathbb{P}_\theta, \mathbb{P}_{\theta'}) \leqslant \mathbb{E}_z \|G_\theta(z) - G_{\theta'}(z)\| \rightarrow_{\theta\rightarrow\theta'} 0$$
$$\Rightarrow |W_1(\mathbb{P}_r, \mathbb{P}_\theta) - W_1(\mathbb{P}_r, \mathbb{P}_{\theta'})| \leqslant W_1(\mathbb{P}_\theta, \mathbb{P}_{\theta'}) \rightarrow_{\theta\rightarrow\theta'} 0$$

□

Wasserstein-1 distance

Proof.

Let G be locally Lipschitz $\Rightarrow$ for given pair (θ, z) there exists a constant $L(\theta, z)$ and a open set U with $(\theta, z) \in U$, such that

$$\begin{aligned} \forall(\theta', z') \in U : \quad & \|G_\theta(z) - G_{\theta'}(z)\| \leqslant L(\theta, z)(\left\|\theta - \theta'\right\| + \left\|z - z'\right\|) \\ \Rightarrow & \mathbb{E}_z \|G_\theta(z) - G_{\theta'}(z)\| \leqslant \left\|\theta - \theta'\right\| \mathbb{E}_z L(\theta, z) \end{aligned}$$

Let $U_\theta \triangleq \{\theta' \mid (\theta', z) \in U\}$ and $L(\theta) \triangleq \mathbb{E}_z L(\theta, z)$ (regularity assumption!). Then:

$$\forall \theta' \in U_\theta : \quad |W_1(\mathbb{P}_r, \mathbb{P}_\theta) - W_1(\mathbb{P}_r, \mathbb{P}_{\theta'})| \leqslant W_1(\mathbb{P}_\theta, \mathbb{P}_{\theta'}) \leqslant L(\theta) \left\|\theta - \theta'\right\|$$

As a result, $W_1(\mathbb{P}_r, \mathbb{P}_\theta)$ is locally Lipschitz $\Rightarrow$ $W_1(\mathbb{P}_r, \mathbb{P}_\theta)$ is everywhere continuous. Due to Radamacher's theorem, we follow it has to be differentiable almost everywhere. □

Wasserstein-1 distance

Corollary

Let G_θ be any feedforward neural network parametrized by θ, and $p(z)$ a prior over z such that $\mathbb{E}_{z\sim p(z)} \|z\| < \infty$ (e.g. Gaussian, uniform, etc). Then the regularity assumption (from previous theorem) is satisfied and therefore $W_1(\mathbb{P}_r, \mathbb{P}_\theta)$ is continuous everywhere and differentiable almost everywhere as function of θ.

Wasserstein-1 distance

Proof.

We consider the proof for networks composed by affine transformations and smooth Lipschitz non-linearities (sigmoid, tanh, elu, etc). The proof for relu activations is much more technical.

Since G is $\mathcal{C}^1$ as a function of $(\theta, z) \Rightarrow$ for any fixed (θ, z) is $L(\theta, z) \leqslant \left\|\nabla_{\theta,z} G_\theta(z)\right\| + \epsilon$ an acceptable local Lipschitz constant for all $\epsilon > 0$.

Let H denote the number of layers. Then $\nabla_z G_\theta(z) = \Pi_{k=1}^H W_k D_k$, where W_k are the weight matrices and D_k the diagonal Jacobians of the non-linearities. Furthermore, $\nabla_{W_k} G_\theta(z) = \left(\left(\Pi_{i=k+1}^H W_i D_i\right) D_k\right) f_{1:k-1}(z)$. Due to the choice of the activation functions, we have $\|D_i\| \leqslant L_{nl}$ for all $i = 1, \ldots, H$ and some constant L_{nl}, and $\|f_{1:k-1}(z)\| \leqslant \|z\| \, L_{nl}^{k-1} \Pi_{i=1}^{k-1} W_i$. Putting all together:

$$\left\|\nabla_{\theta,z} G_\theta(z)\right\| \leqslant \left\|\Pi_{k=1}^H W_k D_k\right\| + \sum_{i=1}^{H} \left\|\left(\left(\Pi_{i=k+1}^H W_i D_i\right) D_k\right) f_{1:k-1}(z)\right\| \leqslant$$

$$\leqslant \underbrace{L_{nl}^H (\Pi_{i=1}^H \|W_i\|}_{C_1(\theta)} + \|z\| \, \underbrace{L_{nl}^H \sum_{k=1}^{H} \left(\Pi_{i=1}^{k-1} \|W_i\|\right)\left(\Pi_{i=k+1}^H \|W_i\|\right))}_{C_2(\theta}$$

$$\Rightarrow \mathbb{E}_{z\sim p(z)} \left\|\nabla_{\theta,z} G_\theta(z)\right\| \leqslant C_1(\theta) + C_2(\theta) \mathbb{E}_{z \sim p(z)} \|z\| < \infty. \qquad \square$$

Wasserstein GAN

- For generator feedforward networks the function $\theta \rightarrow W_1(\mathbb{P}_r, \mathbb{P}_\theta)$ is continuous everywhere and differentiable almost everywhere $\Rightarrow$ might have nicer properties during optimization than $D_{JS}(\mathbb{P}_r||\mathbb{P}_\theta)$
- Infimum is highly intractable $\Rightarrow$ utilize Kantorovich-Rubinstein duality:

$$W_1(\mathbb{P}_r, \mathbb{P}_\theta) = \sup_{\|f\|_L \leqslant 1} \left[\mathbb{E}_{x \sim \mathbb{P}_r} f(x) - \mathbb{E}_{x \sim \mathbb{P}_\theta} f(x)\right]$$

- Replacing $\|f\|_L \leqslant 1$ for $\|f\|_L \leqslant K$ in the supremum for some constant K yields $K \cdot W_1(\mathbb{P}_r, \mathbb{P}_\theta)$
- Idea: Utilize parametric family $\{f_w\}_{w \in \mathcal{W}}$ of K-Lipschitz functions and consider solving the problem

$$\max_{w \in \mathcal{W}} \left[\mathbb{E}_{x \sim \mathbb{P}_r} f_w(x) - \mathbb{E}_{z \sim \mathbb{P}_z} f_w(G_\theta(z))\right]$$

Wasserstein GAN

Theorem

Let $\mathbb{P}_r$ be any distribution. Let $\mathbb{P}_\theta$ be the distribution of $G_\theta(Z)$ with $Z \sim \mathbb{P}_z$ a RV and G_θ a function satisfying the regularity assumption. Then, there is a solution $f : \mathcal{X} \to \mathbb{R}$ to the problem

$$\max_{\|f\|_L \leqslant 1} \left[\mathbb{E}_{x\sim\mathbb{P}_r} f(x) - \mathbb{E}_{x\sim\mathbb{P}_\theta} f(x)\right]$$

and we have

$$\nabla_\theta W_1(\mathbb{P}_r, \mathbb{P}_\theta) = -\mathbb{E}_{z\sim\mathbb{P}_z} \nabla_\theta f(G_\theta(z))$$

Wasserstein GAN

Proof.

Let $V(\tilde{f},\theta) \triangleq \mathbb{E}_{x\sim\mathbb{P}_r}\tilde{f}(x) - \mathbb{E}_{x\sim\mathbb{P}_\theta}\tilde{f}(x) = \mathbb{E}_{x\sim\mathbb{P}_r}\tilde{f}(x) - \mathbb{E}_{z\sim\mathbb{P}_z}\tilde{f}(G_\theta(z))$, where $\tilde{f} \in \mathcal{F} \triangleq \{f : \mathcal{X} \to \mathbb{R} \mid f \in \mathcal{C}_b(\mathcal{X}), \|f\|_L \leqslant 1\}$. Since $\mathcal{X}$ is compact, the Kantorovich-Rubenstein duality implies that there is an $\tilde{f} \in \mathcal{F}$ that attains the value

$$W_1(\mathbb{P}_r, \mathbb{P}_\theta) = \sup_{\tilde{f}\in\mathcal{F}} V(\tilde{f},\theta) = V(f,\theta)$$

Let $X^*(\theta) \triangleq \{f \in \mathcal{F} \mid V(f,\theta) = W_1(\mathbb{P}_r, \mathbb{P}_\theta)\}$ (non empty). Envelope theorem implies that for all $f \in X^*(\theta)$ the following holds:

$$\nabla_\theta W_1(\mathbb{P}_r, \mathbb{P}_\theta) = \nabla_\theta V(f,\theta).$$

Therefore,

$$\begin{aligned}\nabla_\theta W_1(\mathbb{P}_r, \mathbb{P}_\theta) &= \nabla_\theta V(f,\theta) = \\ &= \nabla_\theta\left[\mathbb{E}_{x\sim\mathbb{P}_r} f(x) - \mathbb{E}_{z\sim\mathbb{P}_z} f(G_\theta(x))\right] = \\ &= -\nabla_\theta\left[\mathbb{E}_{z\sim\mathbb{P}_z} f(G_\theta(x))\right]\end{aligned}$$

The proof is completed by showing the commutativity of the gradient and the expectation value via some technical steps. □

Wasserstein GAN

How to find the critic f_w?

1. approximation via a neural network parametrised via weights vector w in a compact space $\mathcal{W}$
2. backpropagate through $\mathbb{E}_{z\sim\mathbb{P}_z}\nabla_\theta f(G_\theta(z))$
3. To ensure that paramaters w lie in compact space after each update, weights are clipped to a fixed box e.g. $\mathcal{W} = [-0.1, 0.1]^l$

Algorithm:

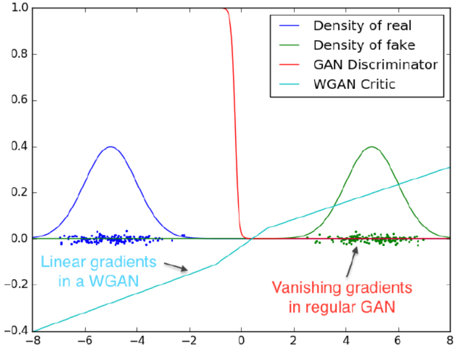
Require: : α, the learning rate. c, the clipping parameter. m, the batch size. n_{critic}, the number of iterations of the critic per generator iteration.
Require: : w_0, initial critic parameters. θ_0, initial generator's parameters.

while θ has not converged **do**
  for $t = 0, ..., n_{\text{critic}}$ **do**
    Sample $\{x^{(i)}\}_{i=1}^m \sim \mathbb{P}_r$ a batch from the real data.
    Sample $\{z^{(i)}\}_{i=1}^m \sim p(z)$ a batch of prior samples.
    $g_w \leftarrow \nabla_w \left[\frac{1}{m}\sum_{i=1}^m f_w(x^{(i)}) - \frac{1}{m}\sum_{i=1}^m f_w(g_\theta(z^{(i)}))\right]$
    $w \leftarrow w + \alpha \cdot \text{RMSProp}(w, g_w)$
    $w \leftarrow \text{clip}(w, -c, c)$
  end for
  Sample $\{z^{(i)}\}_{i=1}^m \sim p(z)$ a batch of prior samples.
  $g_\theta \leftarrow -\nabla_\theta \frac{1}{m}\sum_{i=1}^m f_w(g_\theta(z^{(i)}))$
  $\theta \leftarrow \theta - \alpha \cdot \text{RMSProp}(\theta, g_\theta)$
end while

Wasserstein GAN

Advantages:

- W_1 distance is continuous and differentiable a.e.$\Rightarrow$ it is possible to train the critic f_w until optimality $\Rightarrow$ more reliable generator updates without facing vanishing gradients
- This is not the case for D_{JS}: as the discriminator gets better (and updates more reliable), the gradients start to vanish since the true gradient is zero due to saturation.



Wasserstein GAN

Advantages:

- Critics trained until optimality avoid mode collapses during GAN training.

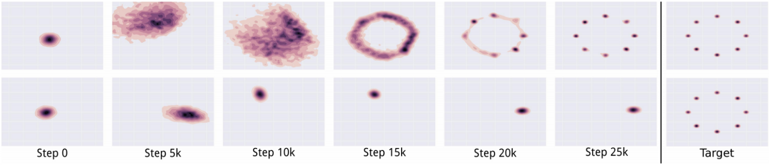


Figure: Mode collapse - If the generator is trained extensively without updates to the discriminator, it will converge to find the optimal image which fools discriminator the most and therefore will become independent of latent space input. Both networks are then overfitted to exploit short-term opponent weakness.

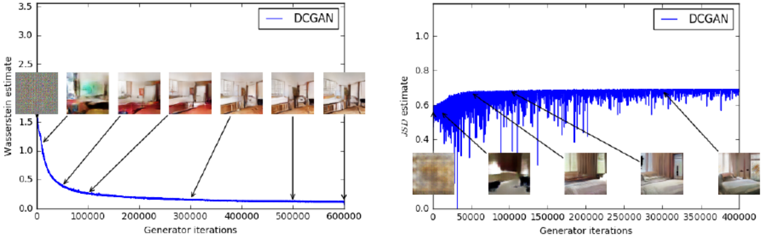
Wasserstein GAN

Figure: Example of a generator trained with standard GAN algorithm suffering from mode collapse. The model generates similar images for different latent space input.

Wasserstein GAN

Advantages:

- Wasserstein GAN loss shows properties of convergence, i.e. one is able to quantify which models are doing better than others and has not to stare at generated samples during iteration to detect failure modes.



Wasserstein GAN

Disadvantages

- Clipping weights is a terrible way to enforce Lipschitz constraint. Large clipping parameters make it harder to train critic until optimality, small clipping parameter can easily lead to vanishing gradients for high model complexity.
- Wasserstein GAN training becomes instable for momentum based optimizers such as Adam on the critic (critic loss is nonstationary!)

Thank you for your attention!

Christoph Angermann
https://applied-math.uibk.ac.at/