# Deep Generative Models
# Lecture 11

Roman Isachenko

Ozon Masters

Spring, 2021

# Recap of previous lecture

## Vanilla GAN

$$\min_G \max_D V(G, D) = \min_G \max_D \left[\mathbb{E}_{\pi(\mathbf{x})} \log D(\mathbf{x}) + \mathbb{E}_{p(\mathbf{z})} \log(1 - D(G(\mathbf{z})))\right]$$

## Main problems

- Vanishing gradients (non-saturating GAN does not suffer of it);
- Mode collapse (caused by behaviour of Jensen-Shannon divergence).

## Informal theoretical results

Distribution of real images $\pi(\mathbf{x})$ and distribution of generated images $p(\mathbf{x}|\boldsymbol{\theta})$ are low-dimensional and have disjoint supports. In this case

$$KL(\pi||p) = KL(p||\pi) = \infty, \quad JSD(\pi||p) = \log 2$$

---

*Goodfellow I. J. et al. Generative Adversarial Networks, 2014*
*Arjovsky M., Bottou L. Towards Principled Methods for Training Generative Adversarial Networks, 2017*

# Recap of previous lecture

## Wasserstein distance

$$W(\pi, p) = \inf_{\gamma \in \Gamma(\pi, p)} \mathbb{E}_{(\mathbf{x}, \mathbf{y}) \sim \gamma} \|\mathbf{x} - \mathbf{y}\| = \inf_{\gamma \in \Gamma(\pi, p)} \int \|\mathbf{x} - \mathbf{y}\| \gamma(\mathbf{x}, \mathbf{y}) d\mathbf{x} d\mathbf{y}$$

- $\Gamma(\pi, p)$ – the set of all joint distributions $\Gamma(\mathbf{x}, \mathbf{y})$ with marginals $\pi$ and $p$ ($\int \gamma(\mathbf{x}, \mathbf{y}) d\mathbf{x} = p(\mathbf{y})$, $\int \gamma(\mathbf{x}, \mathbf{y}) d\mathbf{y} = \pi(\mathbf{x})$)
- $\gamma(\mathbf{x}, \mathbf{y})$ – transportation plan (the amount of "dirt" that should be transported from point $\mathbf{x}$ to point $\mathbf{y}$).
- $\gamma(\mathbf{x}, \mathbf{y})$ – the amount, $\|\mathbf{x} - \mathbf{y}\|$– the distance.

## Kantorovich-Rubinstein duality

$$W(\pi || p) = \frac{1}{K} \max_{\|f\|_L \leq K} \left[ \mathbb{E}_{\pi(\mathbf{x})} f(\mathbf{x}) - \mathbb{E}_{p(\mathbf{x})} f(\mathbf{x}) \right],$$

where $\|f\|_L \leq K$ are $K-$Lipschitz continuous functions ($f : \mathcal{X} \to \mathbb{R}$).

*Arjovsky M., Chintala S., Bottou L. Wasserstein GAN, 2017*

# Recap of previous lecture

## Vanilla GAN objective

$$\min_{G} \max_{D} \mathbb{E}_{\pi(\mathbf{x})} \log D(\mathbf{x}) + \mathbb{E}_{p(\mathbf{z})} \log(1 - D(G(\mathbf{z})))$$

## WGAN objective

$$\min_{G} W(\pi || p) = \min_{G} \max_{\phi \in \boldsymbol{\Phi}} \left[ \mathbb{E}_{\pi(\mathbf{x})} f(\mathbf{x}, \phi) - \mathbb{E}_{p(\mathbf{z})} f(G(\mathbf{z}), \phi) \right].$$

- Discriminator $D$ is similar to the function $f$, but not the same (it is not a classifier anymore). In the WGAN model, function $f$ is usually called *critic*.
- "Weight clipping is a clearly terrible way to enforce a Lipschitz constraint". If the clipping parameter is large, it is hard to train the critic till optimality. If the clipping parameter is too small, it could lead to vanishing gradients.

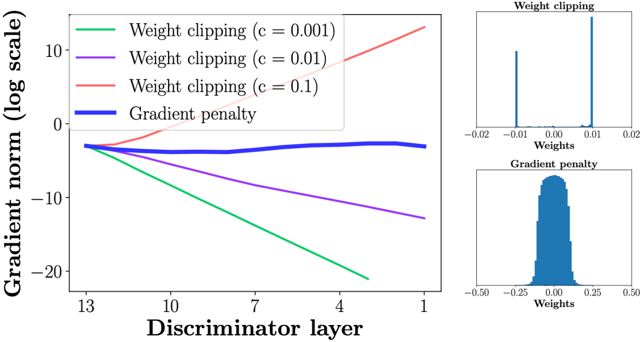
---

*Arjovsky M., Chintala S., Bottou L. Wasserstein GAN, 2017*

# Wasserstein GAN with Gradient Penalty

## Weight clipping annalysis

- The critic ignores higher moments of the data distribution.
- The gradients either grow or decay exponentially.




Gradient penalty makes the gradients more stable.

*Gulrajani I. et al. Improved Training of Wasserstein GANs, 2017*

# Wasserstein GAN with Gradient Penalty

### Theorem

Let $\pi(\mathbf{x})$ and $p(\mathbf{x})$ be two distribution in $\mathcal{X}$, a compact metric space. Then, there is 1-Lipschitz function $f^*$ which is the optimal solution of

$$\max_{\|f\|_L \leq 1} \left[\mathbb{E}_{\pi(\mathbf{x})} f(\mathbf{x}) - \mathbb{E}_{p(\mathbf{x})} f(\mathbf{x})\right].$$

Let $\gamma$ be the optimal transportation plan between $\pi(\mathbf{x})$ and $p(\mathbf{x})$. Then, if $f^*$ is differentiable, $\gamma(\mathbf{x} = \mathbf{y}) = 0$ and $\hat{\mathbf{x}}_t = t\mathbf{x} + (1-t)\mathbf{y}$ with $t \in [0, 1]$ it holds that

$$\mathbb{P}_{(\mathbf{x},\mathbf{y})\sim\gamma} \left[\nabla f^*(\hat{\mathbf{x}}_t) = \frac{\mathbf{y} - \hat{\mathbf{x}}_t}{\|\mathbf{y} - \hat{\mathbf{x}}_t\|}\right] = 1.$$

### Corollary

$f^*$ has gradient norm 1 almost everywhere under $\pi(\mathbf{x})$ and $p(\mathbf{x})$.

---

*Gulrajani I. et al. Improved Training of Wasserstein GANs, 2017*

# Wasserstein GAN with Gradient Penalty

A differentiable function is 1-Lipschtiz if and only if it has gradients with norm at most 1 everywhere.

## Gradient penalty

$$W(\pi || p) = \underbrace{\mathbb{E}_{\pi(\mathbf{x})} f(\mathbf{x}) - \mathbb{E}_{p(\mathbf{x})} f(\mathbf{x})}_{\text{original critic loss}} + \lambda \underbrace{\mathbb{E}_{\hat{\mathbf{x}}} \left[ (\|\nabla_{\hat{\mathbf{x}}} f(\hat{\mathbf{x}})\|_2 - 1)^2 \right]}_{\text{gradient penalty}},$$

- Samples $\hat{\mathbf{x}}$ are uniformly sampled along straight lines between pairs of points from the data distribution $\pi(\mathbf{x})$ and the generator distribution $p(\mathbf{x}|\boldsymbol{\theta})$.
- Enforcing the unit gradient norm constraint everywhere is intractable, it turns out to be sifficient to enforce it only along these straight lines.

*Gulrajani I. et al. Improved Training of Wasserstein GANs, 2017*

# Wasserstein GAN with Gradient Penalty

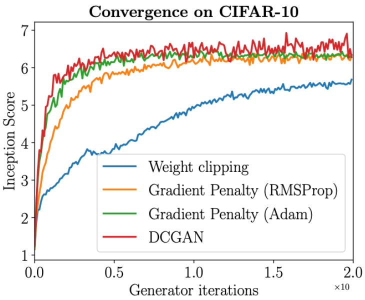
**Algorithm 1** WGAN with gradient penalty. We use default values of $\lambda = 10$, $n_{\text{critic}} = 5$, $\alpha = 0.0001$, $\beta_1 = 0$, $\beta_2 = 0.9$.

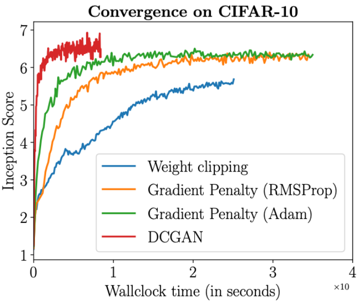
**Require:** The gradient penalty coefficient $\lambda$, the number of critic iterations per generator iteration $n_{\text{critic}}$, the batch size $m$, Adam hyperparameters $\alpha, \beta_1, \beta_2$.
**Require:** initial critic parameters $w_0$, initial generator parameters $\theta_0$.

**while** $\theta$ has not converged **do**
  **for** $t = 1, ..., n_{\text{critic}}$ **do**
    **for** $i = 1, ..., m$ **do**
      Sample real data $\boldsymbol{x} \sim \mathbb{P}_r$, latent variable $\boldsymbol{z} \sim p(\boldsymbol{z})$, a random number $\epsilon \sim U[0, 1]$.
      $\tilde{\boldsymbol{x}} \leftarrow G_\theta(\boldsymbol{z})$
      $\hat{\boldsymbol{x}} \leftarrow \epsilon \boldsymbol{x} + (1 - \epsilon)\tilde{\boldsymbol{x}}$
      $L^{(i)} \leftarrow D_w(\tilde{\boldsymbol{x}}) - D_w(\boldsymbol{x}) + \lambda(\|\nabla_{\hat{\boldsymbol{x}}} D_w(\hat{\boldsymbol{x}})\|_2 - 1)^2$
    **end for**
    $w \leftarrow \text{Adam}(\nabla_w \frac{1}{m} \sum_{i=1}^m L^{(i)}, w, \alpha, \beta_1, \beta_2)$
  **end for**
  Sample a batch of latent variables $\{\boldsymbol{z}^{(i)}\}_{i=1}^m \sim p(\boldsymbol{z})$.
  $\theta \leftarrow \text{Adam}(\nabla_\theta \frac{1}{m} \sum_{i=1}^m -D_w(G_\theta(\boldsymbol{z})), \theta, \alpha, \beta_1, \beta_2)$
**end while**

*Gulrajani I. et al. Improved Training of Wasserstein GANs, 2017*

# Wasserstein GAN with Gradient Penalty

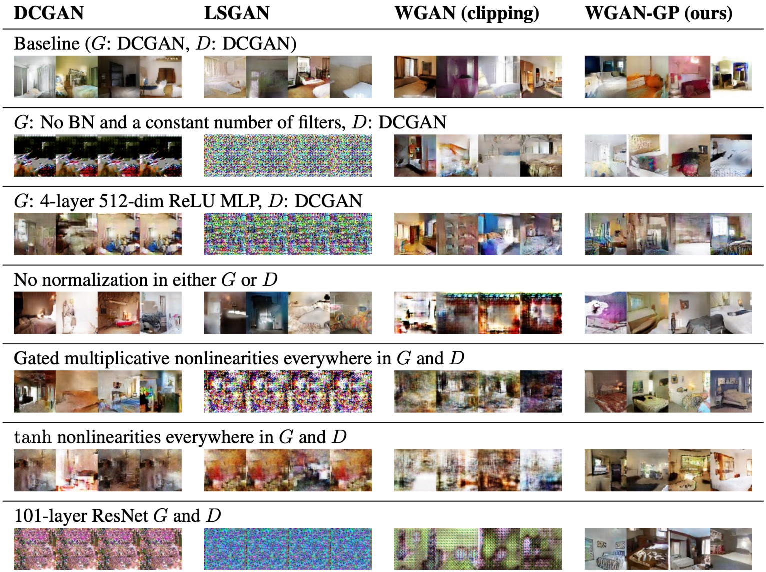



| **Min. score** | **Only GAN** | **Only WGAN-GP** | **Both succeeded** | **Both failed** |
|---|---|---|---|---|
| 1.0 | 0 | 8 | 192 | 0 |
| 3.0 | 1 | 88 | 110 | 1 |
| 5.0 | 0 | 147 | 42 | 11 |
| 7.0 | 1 | 104 | 5 | 90 |
| 9.0 | 0 | 0 | 0 | 200 |

*Gulrajani I. et al. Improved Training of Wasserstein GANs, 2017*

# Wasserstein GAN with Gradient Penalty




*Gulrajani I. et al. Improved Training of Wasserstein GANs, 2017*

# Spectral Normalization GAN

How else could we enforce Lipschitzness?

## Fact 1

Let denote by $\sigma(\mathbf{A})$ a spectral norm of matrix $\mathbf{A}$.

$$\sigma(\mathbf{A}) = \max_{\mathbf{h} \neq 0} \frac{\|\mathbf{A}\mathbf{h}\|_2}{\|\mathbf{h}\|_2} = \max_{\|\mathbf{h}\|_2 \leq 1} \|\mathbf{A}\mathbf{h}\|_2 = \lambda_{\max}(\mathbf{A}),$$

where $\lambda_{\max}(\mathbf{A})$ is the largest singular value of $\mathbf{A}$.
By definition, Lipschitz norm is

$$\|\mathbf{g}\|_L = \sup_{\mathbf{x}} \sigma(\nabla \mathbf{g}(\mathbf{x}))$$

## Fact 2

Lipschitz norm of superposition is bounded above by product of Lipschitz norms

$$\|\mathbf{g}_1 \circ \mathbf{g}_2\|_L \leq \|\mathbf{g}_1\|_L \cdot \|\mathbf{g}_2\|_L$$

---

*Miyato T. et al. Spectral Normalization for Generative Adversarial Networks, 2018*

# Spectral Normalization GAN

Let consider the critic $f(\mathbf{x}, \phi)$ of the following form:

$$f(\mathbf{x}, \phi) = \mathbf{W}_{K+1} a_K(\mathbf{W}_K a_{K-1}(\dots a_1(\mathbf{W}_1 \mathbf{x}) \dots)).$$

This feedforward network is a superposition of simple functions.

- $a_k$ is a pointwise nonlinearities. We assume that $\|a_k\|_L = 1$ (it holds for ReLU).
- $\mathbf{g}(\mathbf{h}) = \mathbf{W}\mathbf{h}$ is a linear transformation ($\nabla \mathbf{g}(\mathbf{x}) = \mathbf{W}$).

$$\|\mathbf{g}\|_L = \sup_{\mathbf{x}} \sigma(\nabla \mathbf{g}(\mathbf{x})) = \sigma(\mathbf{W}).$$

## Critic spectral norm

$$\|f\|_L \leq \|\mathbf{W}_{K+1}\| \cdot \prod_{k=1}^{K} \|a_k\|_L \cdot \|\mathbf{W}_k\| = \prod_{k=1}^{K+1} \sigma(\mathbf{W}_k).$$

If we replace the weights in the critic $f(\mathbf{x}, \phi)$ by $\mathbf{W}_k^{SN} = \mathbf{W}_k / \sigma(\mathbf{W}_k)$, we will get $\|f\|_L \leq 1$.

*Miyato T. et al. Spectral Normalization for Generative Adversarial Networks, 2018*

# Spectral Normalization GAN

How to compute $\sigma(\mathbf{W})$?
If we apply singular value decomposition to compute the $\sigma(\mathbf{W})$ at each round of the algorithm, the algorithm becomes intractable.

## Power iteration

- $\mathbf{u}_0$ – random vector.
- for $k = 0, \dots, n-1$: ($n$ is a large enough number of steps)

$$\mathbf{v}_{k+1} = \frac{\mathbf{W}^T \mathbf{u}_k}{\|\mathbf{W}^T \mathbf{u}_k\|}, \quad \mathbf{u}_{k+1} = \frac{\mathbf{W}\mathbf{v}_{k+1}}{\|\mathbf{W}\mathbf{v}_{k+1}\|}$$

- approximate the spectral norm

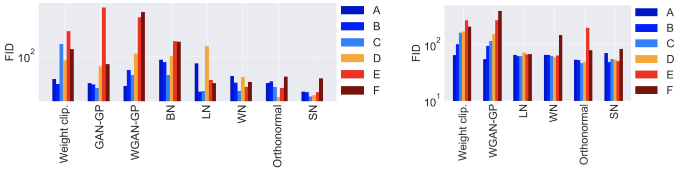
$$\sigma(\mathbf{W}) \approx \mathbf{u}_n^T \mathbf{W} \mathbf{v}_n$$

*Miyato T. et al. Spectral Normalization for Generative Adversarial Networks, 2018*

# Spectral Normalization GAN

**Algorithm 1** SGD with spectral normalization

- Initialize $\tilde{\boldsymbol{u}}_l \in \mathcal{R}^{d_l}$ for $l = 1, \ldots, L$ with a random vector (sampled from isotropic distribution).
- For each update and each layer $l$:
  1. Apply power iteration method to a unnormalized weight $W^l$:
  $$\tilde{\boldsymbol{v}}_l \leftarrow (W^l)^{\mathrm{T}} \tilde{\boldsymbol{u}}_l / \|(W^l)^{\mathrm{T}} \tilde{\boldsymbol{u}}_l\|_2 \tag{20}$$
  $$\tilde{\boldsymbol{u}}_l \leftarrow W^l \tilde{\boldsymbol{v}}_l / \|W^l \tilde{\boldsymbol{v}}_l\|_2 \tag{21}$$
  2. Calculate $\bar{W}_{\mathrm{SN}}$ with the spectral norm:
  $$\bar{W}^l_{\mathrm{SN}}(W^l) = W^l / \sigma(W^l), \text{ where } \sigma(W^l) = \tilde{\boldsymbol{u}}_l^{\mathrm{T}} W^l \tilde{\boldsymbol{v}}_l \tag{22}$$
  3. Update $W^l$ with SGD on mini-batch dataset $\mathcal{D}_M$ with a learning rate $\alpha$:
  $$W^l \leftarrow W^l - \alpha \nabla_{W^l} \ell(\bar{W}^l_{\mathrm{SN}}(W^l), \mathcal{D}_M) \tag{23}$$




(a) CIFAR-10 (b) STL-10

*Miyato T. et al. Spectral Normalization for Generative Adversarial Networks, 2018*

# Divergences

## What do we have?

- Forward KL divergence in maximum likelihood estimation
- Reverse KL in variational inference
- JS divergence in vanilla gan
- Wasserstein distance in WGAN

## What is a divergence?

Let $\mathcal{S}$ be the set of all possible probability distributions. Then $D : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$ is a divergence if

- $D(\pi || p) \geq 0$ for all $\pi, p \in \mathcal{S}$;
- $D(\pi || p) = 0$ if and only if $\pi \equiv p$.

## General divergence minimization task

$$\min_{p} D(\pi || p)$$

# f-divergence family

## f-divergence

$$D_f(\pi||p) = \mathbb{E}_{p(\mathbf{x})} f\left(\frac{\pi(\mathbf{x})}{p(\mathbf{x})}\right) = \int p(\mathbf{x}) f\left(\frac{\pi(\mathbf{x})}{p(\mathbf{x})}\right) d\mathbf{x}.$$

Here $f : \mathbb{R}_+ \to \mathbb{R}$ is a convex, lower semicontinuous function satisfying $f(1) = 0$.

| Name | $D_f(P\|Q)$ | Generator $f(u)$ |
|---|---|---|
| Kullback-Leibler | $\int p(x) \log \frac{p(x)}{q(x)}\, \mathrm{d}x$ | $u \log u$ |
| Reverse KL | $\int q(x) \log \frac{q(x)}{p(x)}\, \mathrm{d}x$ | $-\log u$ |
| Pearson $\chi^2$ | $\int \frac{(q(x)-p(x))^2}{p(x)}\, \mathrm{d}x$ | $(u-1)^2$ |
| Squared Hellinger | $\int \left(\sqrt{p(x)} - \sqrt{q(x)}\right)^2 \mathrm{d}x$ | $(\sqrt{u}-1)^2$ |
| Jensen-Shannon | $\frac{1}{2}\int p(x) \log \frac{2p(x)}{p(x)+q(x)} + q(x) \log \frac{2q(x)}{p(x)+q(x)}\, \mathrm{d}x$ | $-(u+1)\log \frac{1+u}{2} + u \log u$ |
| GAN | $\int p(x) \log \frac{2p(x)}{p(x)+q(x)} + q(x) \log \frac{2q(x)}{p(x)+q(x)}\, \mathrm{d}x - \log(4)$ | $u \log u - (u+1)\log(u+1)$ |

*Nowozin S., Cseke B., Tomioka R. f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization, 2016*

# f-divergence family

## Fenchel conjugate

$$f^*(t) = \sup_{u \in \mathrm{dom}_f} \left( ut - f(u) \right), \quad f(u) = \sup_{t \in \mathrm{dom}_{f^*}} \left( ut - f^*(t) \right)$$

**Important property:** $f^{**} = f$ for convex $f$.

## f-divergence

$$D_f(\pi || p) = \mathbb{E}_{p(\mathbf{x})} f\left(\frac{\pi(\mathbf{x})}{p(\mathbf{x})}\right) = \int p(\mathbf{x}) f\left(\frac{\pi(\mathbf{x})}{p(\mathbf{x})}\right) d\mathbf{x} =$$

$$= \int p(\mathbf{x}) \sup_{t \in \mathrm{dom}_{f^*}} \left( \frac{\pi(\mathbf{x})}{p(\mathbf{x})} t - f^*(t) \right) d\mathbf{x} =$$

$$= \int \sup_{t \in \mathrm{dom}_{f^*}} \left( \pi(\mathbf{x}) t - p(\mathbf{x}) f^*(t) \right) d\mathbf{x}.$$

Here we seek value of $t$, which gives us maximum value of $\pi(\mathbf{x}) t - p(\mathbf{x}) f^*(t)$, for each data point $\mathbf{x}$.

---

*Nowozin S., Cseke B., Tomioka R. f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization, 2016*

# f-divergence family

## f-divergence

$$D_f(\pi||p) = \mathbb{E}_{p(\mathbf{x})} f\left(\frac{\pi(\mathbf{x})}{p(\mathbf{x})}\right) = \int p(\mathbf{x}) f\left(\frac{\pi(\mathbf{x})}{p(\mathbf{x})}\right) d\mathbf{x}.$$

## Variational f-divergence estimation

$$D_f(\pi||p) = \int \sup_{t \in \mathrm{dom}_{f^*}} \left(\pi(\mathbf{x}) t - p(\mathbf{x}) f^*(t)\right) d\mathbf{x} \geq$$

$$\geq \sup_{T \in \mathcal{T}} \int \left(\pi(\mathbf{x}) T(\mathbf{x}) - p(\mathbf{x}) f^*(T(\mathbf{x}))\right) d\mathbf{x} =$$

$$= \sup_{T \in \mathcal{T}} \left[\mathbb{E}_\pi T(\mathbf{x}) - \mathbb{E}_p f^*(T(\mathbf{x}))\right]$$

This is a lower bound because of Jensen-Shannon inequality and restricted class of functions $\mathcal{T} : \mathcal{X} \to \mathbb{R}$.

**Note:** To evaluate lower bound we only need samples from $\pi(\mathbf{x})$ and $p(\mathbf{x})$. Hence, we could fit implicit generative model.

*Nowozin S., Cseke B., Tomioka R. f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization, 2016*

# f-divergence family

## Variational divergence estimation

$$D_f(\pi||p) \geq \sup_{T \in \mathcal{T}} \left[\mathbb{E}_\pi T(\mathbf{x}) - \mathbb{E}_p f^*(T(\mathbf{x}))\right]$$

The lower bound is tight for $T^*(\mathbf{x}) = f'\left(\frac{\pi(\mathbf{x})}{p(\mathbf{x})}\right)$.

(a) GAN (b) KL (c) Squared Hellinger

---

*Nowozin S., Cseke B., Tomioka R. f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization, 2016*

# Evaluation of likelihood-free models

How to evaluate generative models?

## Likelihood-based models

- Split data to train/val/test.
- Fit model on the train part.
- Tune hyperparameters on the validation part.
- Evaluate generalization by reporting likelihoods on the test set.

## Not all models have tractable likelihoods

- VAE: compare ELBO values.
- GAN: ???

# Evaluation of likelihood-free models

Let take some pretrained image classification model to get the conditional label distribution $p(y|\mathbf{x})$ (e.g. ImageNet classifier).

## What do we want from samples?

- **Sharpness**

  Low sharpness

  High sharpness

  The conditional distribution $p(y|\mathbf{x})$ should have low entropy (each image $\mathbf{x}$ should have distinctly recognizable object).

- **Diversity**

  Low diversity

  High diversity

  The marginal distribution $p(y) = \int p(y|\mathbf{x})p(\mathbf{x})d\mathbf{x}$ should have high entropy (there should be as many classes generated as possible).

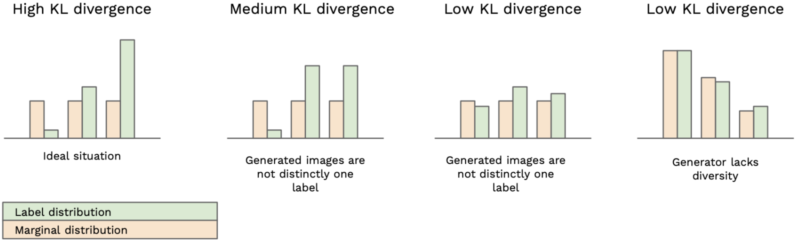
---

*image credit: https://deepgenerativemodels.github.io*

# Evaluation of likelihood-free models

## What do we want from samples?

- **Sharpness.** The conditional distribution $p(y|\mathbf{x})$ should have low entropy (each image $\mathbf{x}$ should have distinctly recognizable object).
- **Diversity.** The marginal distribution $p(y) = \int p(y|\mathbf{x})p(\mathbf{x})d\mathbf{x}$ should have high entropy (there should be as many classes generated as possible).




---

*image credit: https://medium.com/octavian-ai/a-simple-explanation-of-the-inception-score-372dff6a8c7a*

# Evaluation of likelihood-free models

## What do we want from samples?

- Sharpness $\Rightarrow$ low $H(y|\mathbf{x}) = -\sum_y \int_\mathbf{x} p(y, \mathbf{x}) \log p(y|\mathbf{x}) d\mathbf{x}$.
- Diversity $\Rightarrow$ high $H(y) = -\sum_y p(y) \log p(y)$.

## Inception Score

$$
\begin{aligned}
IS &= \exp(H(y) - H(y|\mathbf{x})) \\
&= \exp\left(-\sum_y p(y)\log p(y) + \sum_y \int_\mathbf{x} p(y, \mathbf{x}) \log p(y|\mathbf{x}) d\mathbf{x}\right) \\
&= \exp\left(\sum_y \int_\mathbf{x} p(y, \mathbf{x}) \log \frac{p(y|\mathbf{x})}{p(y)} d\mathbf{x}\right) \\
&= \exp\left(\mathbb{E}_\mathbf{x} \sum_y p(y|\mathbf{x}) \log \frac{p(y|\mathbf{x})}{p(y)}\right) = \exp\left(\mathbb{E}_\mathbf{x} KL(p(y|\mathbf{x}) || p(y))\right)
\end{aligned}
$$

*Salimans T. et al. Improved Techniques for Training GANs, 2016*

# Evaluation of likelihood-free models

## Inception Score

$$IS = \exp\left(\mathbb{E}_{\mathbf{x}} KL(p(y|\mathbf{x})||p(y))\right)$$

## IS limitations

- Inception score depends on the quality of the pretrained classifier $p(y|\mathbf{x})$.
- If generator produces images with a different set of labels from the classifier training set, IS will be low.
- If the generator produces one image per class, the IS will be perfect (there is no measure of intra-class diversity).
- IS only require samples from the generator and do not take into account the desired data distribution $\pi(\mathbf{x})$ directly (only implicitly via a classifier).

---

*Barratt S., Sharma R. A Note on the Inception Score, 2018*

# Evaluation of likelihood-free models

### Theorem

If $\pi(\mathbf{x})$ and $p(\mathbf{x}|\boldsymbol{\theta})$ has moment generation functions then

$$\pi(\mathbf{x}) = p(\mathbf{x}|\boldsymbol{\theta}) \Leftrightarrow \mathbb{E}_{\pi}\mathbf{x}^k = \mathbb{E}_p\mathbf{x}^k, \quad \forall k \geq 1.$$

This is intractable to calculate all moments.

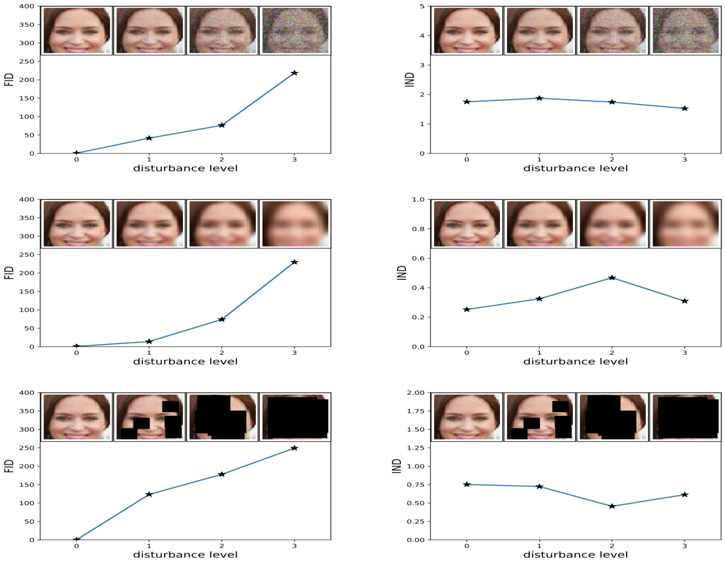
### Frechet Inception Distance

$$D^2(\pi, p) = \|\mathbf{m}_{\pi} - \mathbf{m}_p\|_2^2 + \mathrm{Tr}\left(\mathbf{C}_{\pi} + \mathbf{C}_p - 2\sqrt{\mathbf{C}_{\pi}\mathbf{C}_p}\right)$$

- $\mathbf{m}_{\pi}$, $\mathbf{C}_{\pi}$ are mean vector and covariance matrix of feature representations for real samples from $\pi(\mathbf{x})$
- $\mathbf{m}_p$, $\mathbf{C}_p$ are mean vector and covariance matrix of feature representations for generated samples from $p(\mathbf{x}|\boldsymbol{\theta})$.
- Representations are outputs of intermediate layer from pretrained classification model.

*Heusel M. et al. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium, 2017*

# Evaluation of likelihood-free models




*Heusel M. et al. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium, 2017*

# Evaluation of likelihood-free models

## Frechet Inception Distance

$$D^2(\pi, p) = \|\mathbf{m}_\pi - \mathbf{m}_p\|_2^2 + \mathrm{Tr}\left(\mathbf{C}_\pi + \mathbf{C}_p - 2\sqrt{\mathbf{C}_\pi \mathbf{C}_p}\right)$$

## FID limitations

- FID depends on the pretrained classification model.
- FID needs a large samples size for evaluation.
- Calculation of FID is slow.
- FID extimates only two sample moments.

---

*Heusel M. et al. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium, 2017*

# Summary

- Wasserstein GAN uses Kantorovich-Rubinstein duality to estimate Wasserstein distance.
- Gradient Penalty proposes the regularizer to enforce Lipschitzness.
- Spectral normalization is a weight normalization technique to enforce Lipshitzness.
- f-divergence family is a unified framework for divergence minimization.
- Inception Score and Frechet Inception Distance are the common metrics for GAN evaluation.

# Summary