Deep Generative Models
Lecture 11

Roman Isachenko

Ozon Masters

Spring, 2021

Recap of previous lecture

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 ($f : \mathcal{X} \to \mathbb{R}$).

Gradient penalty

$$W(\pi || p) = \underbrace{\mathbb{E}_{\pi(\mathbf{x})} f(\mathbf{x}) - \mathbb{E}_{p(\mathbf{x})} f(\mathbf{x})}_{\text{original critic loss}} + \lambda \underbrace{\mathbb{E}_{U[0,1]} \left[\left(\|\nabla_{\hat{\mathbf{x}}} f(\hat{\mathbf{x}})\|_2 - 1 \right)^2 \right]}_{\text{gradient penalty}}.$$

Samples $\hat{\mathbf{x}}_t = t\mathbf{x} + (1-t)\mathbf{y}$ with $t \in [0, 1]$ are uniformly sampled along straight lines between pairs of points: $\mathbf{x}$ from the data distribution $\pi(\mathbf{x})$ and $\mathbf{y}$ from the generator distribution $p(\mathbf{x}|\boldsymbol{\theta})$.

Arjovsky M., Chintala S., Bottou L. Wasserstein GAN, 2017
Gulrajani I. et al. Improved Training of Wasserstein GANs, 2017

Recap of previous lecture

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

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

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

Spectral Normalization GAN

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 $\|f\|_L \leq 1$.

Power iteration approximates the value of $\sigma(\mathbf{W})$.

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

Recap of previous lecture

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 $\pi \equiv p$.

f-divergence minimization

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

Variational divergence estimation

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

where $f^*(t) = \sup_{u \in \mathrm{dom}_f} (ut - f(u))$ – Fenchel conjugate.

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

Recap of previous lecture

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

Evaluation of likelihood-free models

- 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

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

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 is related to moment matching.

Salimans T. et al. Improved Techniques for Training GANs, 2016
Heusel M. et al. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium, 2017

Evolution of GANs


2014


2015


2016


2017


2018

- **Vanilla GAN** https://arxiv.org/abs/1406.2661
- **DCGAN** https://arxiv.org/abs/1511.06434
- **CoGAN** https://arxiv.org/abs/1606.07536
- **ProGAN** https://arxiv.org/abs/1710.10196
- **StyleGAN** https://arxiv.org/abs/1812.04948

Self-Attention GAN

- Convolutional layers process the information in a local neighborhood.
- Using convolutional layers alone is computationally inefficient for modeling long-range dependencies in images.

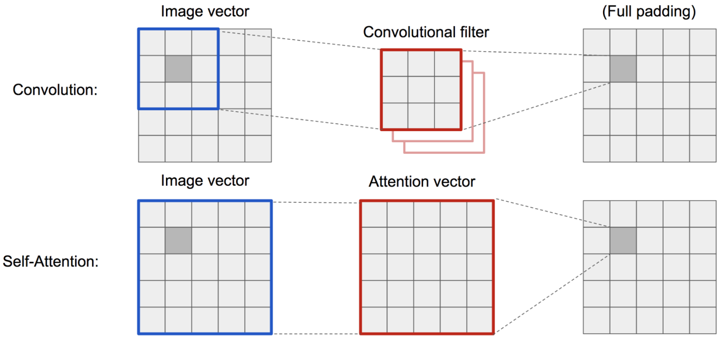
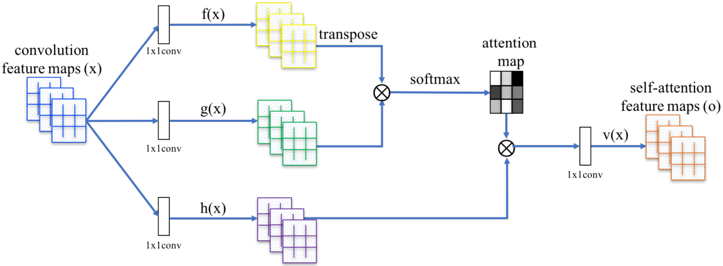


image credit: https://lilianweng.github.io/lil-log/2018/06/24/attention-attention.html

Self-Attention GAN



- $\mathbf{x}$ – feature vector for one feature location.
- N – number of feature locations.

$$\mathbf{f}(\mathbf{x}) = \mathbf{W}_f\mathbf{x}, \quad \mathbf{g}\mathbf{x} = \mathbf{W}_g\mathbf{x}, \quad \mathbf{h}\mathbf{x} = \mathbf{W}_h\mathbf{x}, \quad \mathbf{v}\mathbf{x} = \mathbf{W}_v\mathbf{x}$$

$$s_{ij} = \mathbf{f}(\mathbf{x}_i)^T\mathbf{g}(\mathbf{x}_j), \quad a_{ij} = \frac{\exp s_{ij}}{\sum_{i=1}^N \exp s_{ij}}, \quad \mathbf{o}_j = \mathbf{v}\left(\sum_{i=1}^N a_{ij}\mathbf{h}(\mathbf{x}_i)\right)$$

Zhang H. et al. Self-Attention Generative Adversarial Networks, 2018

Self-Attention GAN

Technical Details

- Hinge loss for training.
- SpectralNorm in both the generator and the discriminator.
- Separate learning rates for the generator and the discriminator.

Model	Inception Score	Intra FID	FID
AC-GAN (Odena et al., 2017)	28.5	260.0	/
SNGAN-projection (Miyato & Koyama, 2018)	36.8	92.4	27.62*
SAGAN	**52.52**	**83.7**	**18.65**

Visualization of attention maps

Zhang H. et al. Self-Attention Generative Adversarial Networks, 2018

BigGAN

Technical Details

- Hinge loss.
- Self-Attention GAN baseline.
- **Orthogonal regularization**

$$\|\mathbf{W}^T\mathbf{W} - \mathbf{I}\|^2 \quad \Rightarrow \quad \|\mathbf{W}^T\mathbf{W} - \mathrm{diag}(\mathbf{W}^T\mathbf{W})\|^2$$

- **Truncation trick.** Components of $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$ which fall outside a predefined range are resampled.

Batch	Ch.	Param (M)	Shared	Skip-z	Ortho.	Itr $\times 10^3$	FID	IS
256	64	81.5	SA-GAN Baseline			1000	18.65	52.52
512	64	81.5	✗	✗	✗	1000	15.30	58.77($\pm$1.18)
1024	64	81.5	✗	✗	✗	1000	14.88	63.03($\pm$1.42)
2048	64	81.5	✗	✗	✗	732	12.39	76.85($\pm$3.83)
2048	96	173.5	✗	✗	✗	295($\pm$18)	9.54($\pm$0.62)	92.98($\pm$4.27)
2048	96	160.6	✓	✗	✗	185($\pm$11)	9.18($\pm$0.13)	94.94($\pm$1.32)
2048	96	158.3	✓	✓	✗	152($\pm$7)	8.73($\pm$0.45)	98.76($\pm$2.84)
2048	96	158.3	✓	✓	✓	165($\pm$13)	8.51($\pm$0.32)	99.31($\pm$2.10)
2048	64	71.3	✓	✓	✓	371($\pm$7)	10.48($\pm$0.10)	86.90($\pm$0.61)

Brock A., Donahue J., Simonyan K. Large Scale GAN Training for High Fidelity Natural Image Synthesis, 2018

BigGAN

Samples (512x512)

Interpolations

Brock A., Donahue J., Simonyan K. Large Scale GAN Training for High Fidelity Natural Image Synthesis, 2018

Progressive Growing GAN

Problems with HR image generation

- Disjoint manifolds $\Rightarrow$ gradient problem.
- Small minibatch $\Rightarrow$ training instability.

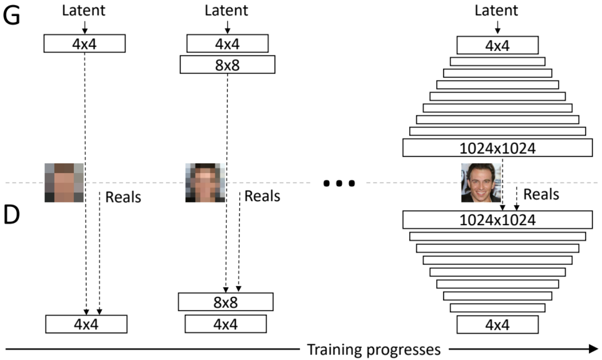
Samples (1024x1024)

Karras T. et al. Progressive Growing of GANs for Improved Quality, Stability, and Variation, 2017

Progressive Growing GAN

Grow both the generator and discriminator progressively, new layers will introduce higher-resolution details as the training progresses.

- Train GAN which generate 4x4 images (2 convs for G and D).
- Add upsampling layers to G, downsampling layers to D.
- Train GAN which generate 8x8 images.
- etc.



Karras T. et al. Progressive Growing of GANs for Improved Quality, Stability, and Variation, 2017

StyleGAN

- Generating of HR images is hard.
- Progressive growing greatly simplifies the task.
- The ability to control specific features of the generated image is very limited.

Face image features

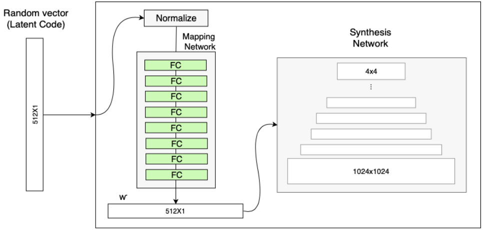
- Coarse (pose, general hair style, face shape). Resolution $4^2 - 8^2$.
- Middle (finer facial features, hair style, eyes open/closed). Resolution $16^2 - 32^2$.
- Fine (color scheme (eye, hair and skin) and micro features). Resolution $64^2 - 1024^2$.

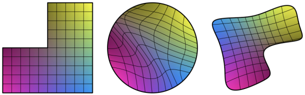
Karras T., Laine S., Aila T. A Style-Based Generator Architecture for Generative Adversarial Networks, 2018

StyleGAN

Mapping Network

- Generator input is likely to be **disentangled**. Each component of input vector **z** should be responsible for one generative factor.
- Mapping network $f : \mathcal{Z} \to \mathcal{W}$ is used to reduce correlations between components of **z**.





(a) Distribution of features in training set (b) Mapping from $\mathcal{Z}$ to features (c) Mapping from $\mathcal{W}$ to features

---

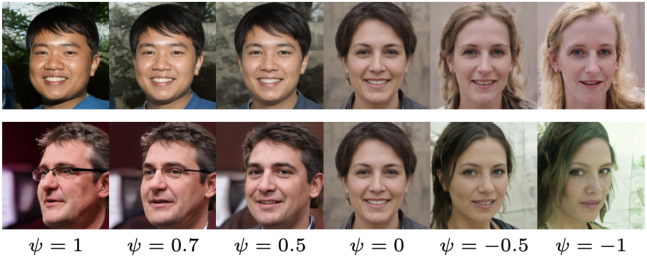
Karras T., Laine S., Aila T. A Style-Based Generator Architecture for Generative Adversarial Networks, 2018

StyleGAN

Truncation trick

$$\mathbf{w}' = \hat{\mathbf{w}} + \psi \cdot (\mathbf{w} - \hat{\mathbf{w}}), \quad \hat{\mathbf{w}} = \mathbb{E}_{\mathbf{z}} p(f(\mathbf{z}))$$

- Constant ψ is a tradeoff between diversity and fidelity.
- $\psi = 0.7$ is used for most of the results.
- Truncation is done only at the low-resolution layers.



Karras T., Laine S., Aila T. A Style-Based Generator Architecture for Generative Adversarial Networks, 2018

StyleGAN

Results

	Method	CelebA-HQ	FFHQ
A	Baseline Progressive GAN [30]	7.79	8.04
B	+ Tuning (incl. bilinear up/down)	6.11	5.25
C	+ Add mapping and styles	5.34	4.85
D	+ Remove traditional input	5.07	4.88
E	+ Add noise inputs	**5.06**	4.42
F	+ Mixing regularization	5.17	**4.40**

Samples (1024x1024)

Karras T., Laine S., Aila T. A Style-Based Generator Architecture for Generative Adversarial Networks, 2018

Density ratio trick

Consider two distributions $q_1(\mathbf{x})$, $q_2(\mathbf{x})$ and probabilistic model

$$p(\mathbf{x}|y) = \begin{cases} q_1(\mathbf{x}), & \text{if } y = 1, \\ q_2(\mathbf{x}), & \text{if } y = 0, \end{cases} \qquad y \sim \text{Bern}(0.5).$$

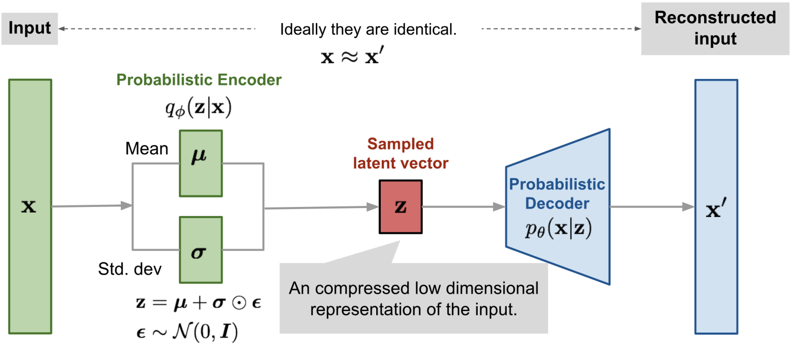
$$p(y = 0) = p(y = 1) = 0.5.$$

Density ratio

$$\frac{q_1(\mathbf{x})}{q_2(\mathbf{x})} = \frac{p(\mathbf{x}|y = 1)}{p(\mathbf{x}|y = 0)} = \frac{p(y = 1|\mathbf{x})p(\mathbf{x})}{p(y = 1)} \bigg/ \frac{p(y = 0|\mathbf{x})p(\mathbf{x})}{p(y = 0)} =$$
$$= \frac{p(y = 1|\mathbf{x})}{p(y = 0|\mathbf{x})} = \frac{p(y = 1|\mathbf{x})}{1 - p(y = 1|\mathbf{x})} = \frac{D(\mathbf{x})}{1 - D(\mathbf{x})}$$

Here $D(\mathbf{x})$ could be treated as a discriminator model the output of which is a probability that $\mathbf{x}$ is a sample from $q_1(\mathbf{x})$ rather than from $q_2(\mathbf{x})$.

VAE recap



- Encoder $q(\mathbf{z}|\mathbf{x}, \phi) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_\phi(\mathbf{x}), \boldsymbol{\sigma}_\phi(\mathbf{x}))$.
- Variational posterior $q(\mathbf{z}|\mathbf{x}, \phi)$ originally approximates the true posterior $p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})$.
- Which methods are you already familiar with to make the posterior is more flexible?

image credit: https://lilianweng.github.io/lil-log/2018/08/12/from-autoencoder-to-beta-vae.html

Adversarial Variational Bayes

ELBO objective

$$\mathcal{L}(\phi, \boldsymbol{\theta}) = \mathbb{E}_{q(\mathbf{z}|\mathbf{x},\phi)} \left[\log p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) + \log p(\mathbf{z}) - \log q(\mathbf{z}|\mathbf{x}, \phi)\right] \rightarrow \max_{\boldsymbol{\phi},\boldsymbol{\theta}}.$$

What is the problem to make the variational posterior model an implicit model?

- The first term is reconstruction loss that needs only samples from $q(\mathbf{z}|\mathbf{x}, \phi)$ to evaluate.
- Reparametrization trick allows to get gradients of reconstruction loss

$$\nabla_\phi \int q(\mathbf{z}|\mathbf{x}, \phi) f(\mathbf{z}) d\mathbf{z} = \nabla_\phi \int r(\epsilon) f(\mathbf{z}) d\epsilon$$
$$= \int r(\epsilon) \nabla_\phi f(g(\mathbf{x}, \epsilon, \phi)) d\epsilon \approx \nabla_\phi f(g(\mathbf{x}, \epsilon^*, \phi)),$$

where $\boldsymbol{\epsilon}^* \sim r(\boldsymbol{\epsilon})$, $\quad \mathbf{z} = g(\mathbf{x}, \boldsymbol{\epsilon}, \phi)$, $\quad \mathbf{z} \sim q(\mathbf{z}|\mathbf{x}, \phi)$.

Mescheder L., Nowozin S., Geiger A. Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks, 2017

Summary