Autoregressive models and Generative Adversarial Networks

Lecture 12

Konstantin Yakovlev [1]

[1]MIPT
Moscow, Russia

MIPT 2024

Outline

- Autoregressive models: PixelCNN, LMConv
- Generative Adversarial Networks
- Evaluation metrics for GANs
- GANs' Representative variants: Cycle GAN

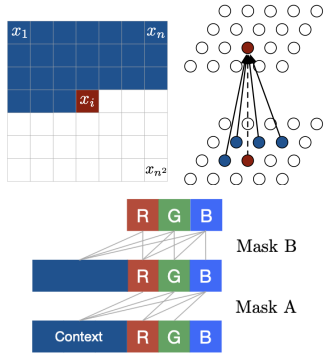
PixelCNN[1]

The goal: assign a probability $p_\theta(\mathbf{x})$ to each image $\mathbf{x} \in \mathbb{R}^{n \times n \times 3}$. The following decomposition is true:

$$p_\theta(\mathbf{x}) = \prod_{i=1}^{n^2} p(x_i|\mathbf{x}_{<i}), \quad p_\theta(x_i|\mathbf{x}_{<i}) =$$

$$= p_\theta(x_{i,R}|\mathbf{x}_{<i}) p(x_{i,G}|\mathbf{x}_{<i}, x_{i,R}) p_\theta(x_{i,B}|\mathbf{x}_{<i}, x_{i,R}, x_{i,G})$$

Architecture: Consider a single-channel image. Use two types of masked convolutions.
Type A: restrict the connection to the current pixel.
Type B: allow the connextion to itself.



Note: Masked convolutions could be implemented by zeroing the parameters corresponding to the restricted area.

[1] van den Oord A. et, al, Pixel Recurrent Neural Networks, 2016

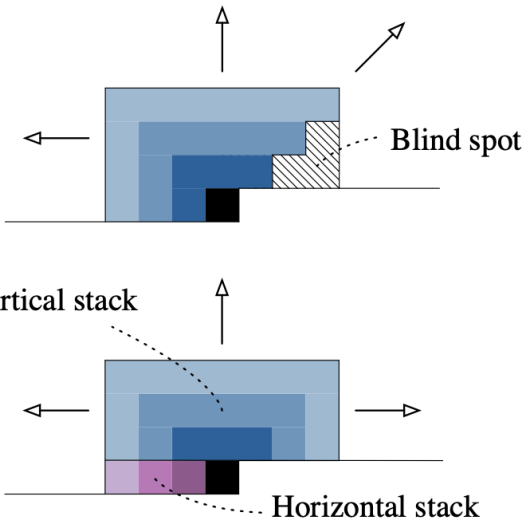
PixelCNN: Blind spot problem[2]

Challenge: a significant portion of the input image is ignored by the masked 3x3 convolutional architecture.

Solution: decompose the architecture into a horizontal and a vertical stack of convolutions. *Vertical stack* conditions on on all rows above resolving the issue.

Horizontal stack conditions on the current row so far and the output of the vertical stack.

Note that the proposed architecture does not violate the autoregressive rule.



[2] van der Oord A. et. al, Conditional Image Generation with PixelCNN Decoders, 2016

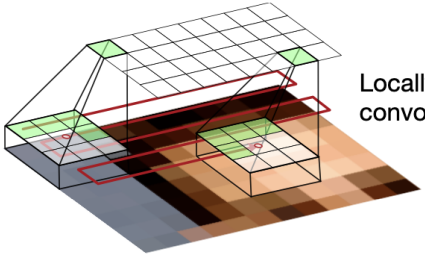
Locally Masked Convolution[3]

Challenge: for tasks such as image completion, PixelCNN is unable to use much of the observed context.

Solution: learn an ensemble of models that share parameters but differ in generation order. Let π define an order over the dimensions. Consider multiple autoregressive decomposition from the predefined set: raster scan, S-curve, Hilber curve.

$$\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}\mathbb{E}_{\pi}\log p_{\theta}(\mathbf{x};\pi) \to \min_{\theta},$$

$$\log p_{\theta}(\mathbf{x};\pi) = \sum_{i=1}^{n^2}\log p_{\theta}\left(x_{\pi(i)}|\mathbf{x}_{\pi(<i)}\right)$$



[3] Jain A. et. al, Locally Masked Convolution for Autoregressive Models, 2020

LMConv: implementation and evaluation

Training: there exists an efficient procedure that extracts $3 \times k \times k$ patches from the image, masks them, and performs a forward pass of the convolutional network.

Whole-image density estimation

BINARIZED MNIST, 28x28	NLL (nats)
DARN (Intractable) (Gregor et al., 2014)	≈84.13
NADE (Uria et al., 2014)	88.33
EoNADE 2hl (128 orders) (Uria et al., 2014)	85.10
EoNADE-5 2hl (128 orders) (Raiko et al., 2014)	84.68
MADE 2hl (32 orders) Germain et al. (2015)	86.64
PixelCNN (van den Oord et al., 2016b)	81.30
PixelRNN (van den Oord et al., 2016b)	79.20
Ours, S-curve (1 order)	78.47
Ours, S-curve (8 orders)	**77.58**

We see that the likelihood is further improved by using ensemble averaging across 8 orders.

Image completion:

Compute th NLL of the top/left/bottom half of the image given the observed one. Note that we have 2 S-shaped orders that cover the whole context.

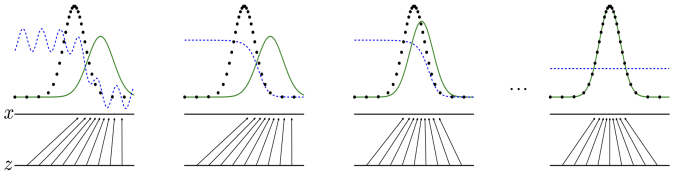
BINARIZED MNIST 28x28 (nats)	T	L	B
Ours (adversarial order)	41.76	39.83	43.35
Ours (1 max context order)	34.99	32.47	36.57
Ours (2 max context orders)	**34.82**	**32.25**	**36.36**

It could be clearly seen that averaging over two maximum context orders further improves log likelihood.

Generative Adversarial Networks[4]

Task: suppose that we want to draw samples from the unknown $\pi(\mathbf{x})$ given a dataset $\mathfrak{D} = \{\mathbf{x}_i\}_{i=1}^n$, $\mathbf{x}_i \sim \pi(\mathbf{x})$.
Method: First, introduce a prior on input noise variables $p(\mathbf{z})$. Second, introduce a *generator* $G(\mathbf{z}, \boldsymbol{\theta}_g)$ which performs a mapping into the data space. Third, introduce a *discriminator* $D(\mathbf{x}, \boldsymbol{\theta}_d)$ which outputs the probability that $\mathbf{x}$ came from $\pi(.)$. The models are optimized through the minimax game. **Note**: here G and D are differentiable functions.

Example



(a) D and G are near near convergence.
(b) The discriminator is optimal given the generator.
(c) The generator is optimal given the discriminatior.
(d) After several steps the discriminatior is unable to differentiate between the two distributions.

$$\min_G \max_D \Big(\mathbb{E}_{\mathbf{x}\sim\pi(\mathbf{x})}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}[\log(1 - D(G(\mathbf{z})))] \Big)$$

[4] Goodfellow I. et. al, Generative Adversarial Nets, 2014

Generative Adversarial Networks: theoretical results

Theorem

For G fixed the optimal discriminator is

$$D^*_G(\mathbf{x}) = \frac{\pi(\mathbf{x})}{\pi(\mathbf{x}) + p_g(\mathbf{x})}$$

Proof: maximize the following objective w.r.t. $D(.)$:

$$V(G, D) = \int \pi(\mathbf{x}) \log(D(\mathbf{x}))d\mathbf{x}$$
$$+ \int p(\mathbf{z}) \log(1 - D(G(\mathbf{z})))d\mathbf{z}$$
$$= \int (\pi(\mathbf{x}) \log D(\mathbf{x}) + p_g(\mathbf{x}) \log(1 - D(\mathbf{x})))d\mathbf{x}$$

Now for each $\mathbf{x}$ minimize the integrand w.r.t. $D(.)$.

$$\arg \min_{d \in [0,1]} (\pi(\mathbf{x}) \log d + p_g(\mathbf{x}) \log(1 - d))$$
$$= \frac{\pi(\mathbf{x})}{\pi(\mathbf{x}) + p_g(\mathbf{x})}.$$

Note that the discriminator does not need to be defined outside of $\mathrm{supp}(p_g) \cup \mathrm{supp}(\pi)$, concluding the proof. Finally,

$$V(G, D^*_G) = \mathbb{E}_{\mathbf{x} \sim \pi(\mathbf{x})} \log \frac{\pi(\mathbf{x})}{\pi(\mathbf{x}) + p_g(\mathbf{x})}$$
$$+ \mathbb{E}_{\mathbf{x} \sim p_g(\mathbf{x})} \log \frac{p_g(\mathbf{x})}{\pi(\mathbf{x}) + p_g(\mathbf{x})}.$$

Generative Adversarial Networks: theoretical results

Theorem

The global minimum of $V(G, D_G^*)$ is achieved if and only if $\pi(.) = p_g(.)$.

Proof: rewrite $V(G, D_G^*)$:

$$V(G, D_G^*) = \mathbb{E}_{\mathbf{x}\sim\pi(\mathbf{x})} \log \frac{\pi(\mathbf{x})}{\pi(\mathbf{x}) + p_g(\mathbf{x})}$$
$$+ \mathbb{E}_{\mathbf{x}\sim p_g(\mathbf{x})} \log \frac{p_g(\mathbf{x})}{\pi(\mathbf{x}) + p_g(\mathbf{x})}$$
$$= -\log 4 + \mathrm{KL}\left(\pi(\mathbf{x}) || \frac{\pi(\mathbf{x}) + p_g(\mathbf{x})}{2}\right)$$
$$+ \mathrm{KL}\left(p_g(\mathbf{x}) || \frac{\pi(\mathbf{x}) + p_g(\mathbf{x})}{2}\right)$$
$$= \mathrm{JS}(\pi(\mathbf{x}) || p_g(\mathbf{x})) - \log 4$$

Since $\mathrm{JS}(.||.) \geq 0$ and is zero only when the distributions are equal, we have that $\pi(.) = p_g(.)$ is the only global minimum.

Limitation: in practice we are capable to parametrize only a limited family of $p_g(.)$ via $G(., \boldsymbol{\theta}_g)$. Therefore, there are no theoretical guarantees of the fulfillment of the conditions of the theorem.

Generative Adversarial Networks: vanishing gradients on the generator[5]

Optimization w.r.t. D:

$$\max_{\theta_d} \mathbb{E}_{\mathbf{x}\sim\pi(\mathbf{x})} \log D_{\theta_d}(\mathbf{x}) + \mathbb{E}_{\mathbf{z}} \log(1 - \log D_{\theta_d}(G(\mathbf{z}))).$$

Optimization w.r.t. G:

$$\min_{\theta_g} \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})} \log(1 - D(G_{\theta_g}(\mathbf{z}))).$$

Limitation: In practice, at the beginning of training when G is weak, D can reject fake samples with high confidence. In this case $\log(1 - D(G(\mathbf{z})))$ saturates.

Theorem(Vanishing gradients on the generator)

Let some mild conditions on the supports of $\pi(.)$ and $p_g(.)$. Let also $\max_{\mathbf{x}} |D(\mathbf{x}) - D^*_G(\mathbf{x})| < \epsilon$ and $\mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}[\| \frac{\partial G_{\theta_g}(\mathbf{z})}{\partial \theta_g} \|_2^2] \leq M^2$. Then

$$\|\nabla_{\theta_g} \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})} \log(1 - D(G_{\theta_g}(\mathbf{z})))\|_2 \leq M \frac{\epsilon}{1-\epsilon}.$$

Solution: Rather then minimizing $\mathbb{E}_{p(\mathbf{z})} \log(1 - D(G(\mathbf{z})))$, maximize $\mathbb{E}_{p(\mathbf{z})} \log D(G(\mathbf{z}))$. Note that this objective function results in the same fixed point of the dynamics of G and D.

[5] Arjovsky M. et. al, Towards principled methods for training Generative Adversarial Networks, 2017

Generative Adversarial Networks: optimization tricks[6]

One-sided label smoothing: penalize oversaturation of the discriminator. replace positive hard targets with $(\alpha, 1-\alpha)$ and negative with $(\beta, 1-\beta)$. Example: $\alpha = 0.9, \beta = 0.1$ (experimentally $\beta = 0$). Therefore, the optimal discriminator:

$$D^*(\mathbf{x}) = \frac{\alpha\pi(\mathbf{x}) + \beta p_g(\mathbf{x})}{\pi(\mathbf{x}) + p_g(\mathbf{x})}.$$

Semi-supervised learning: $\mathcal{L} = \mathcal{L}_{\text{sup}} + \mathcal{L}_{\text{unsup}}$

$$\mathcal{L}_{\text{sup}} = -\mathbb{E}_{(\mathbf{x},y)\sim\pi(\mathbf{x},y)} \log p_{\text{clr}}(y|\mathbf{x}, y < K+1),$$
$$\mathcal{L}_{\text{unsup}} = -\mathbb{E}_{\mathbf{x}\sim\pi(\mathbf{x})} \log(1 - p_{\text{clr}}(y = K+1|\mathbf{x})) - \mathbb{E}_{\mathbf{x}\sim p_g(\mathbf{x})} \log p_{\text{clr}}(y = K+1|\mathbf{x}).$$

Data: $\mathfrak{D} = \{\mathbf{x}_i\}_{i=1}^n$
Result: learned $\boldsymbol{\theta}_g, \boldsymbol{\theta}_d$
while *not converged* **do**
  $\mathbf{z}_{1:M} \sim p(\mathbf{z})$;
  $\mathbf{x}_{1:M} \sim \pi(\mathbf{x})$;
  $\hat{\mathbf{g}}_d \leftarrow \nabla_{\boldsymbol{\theta}_d} \frac{1}{m} \sum_{i=1}^m [\log D(\mathbf{x}_i) + \log(1 - D(G(\mathbf{z}_i)))]$;
  update the discriminator using $\hat{\mathbf{g}}_d$;
  $\hat{\mathbf{g}}_g \leftarrow \nabla_{\boldsymbol{\theta}_g} \frac{1}{m} \sum_{i=1}^m \log(1 - D(G(\mathbf{z}_i)))$;
  update the generator with $\hat{\mathbf{g}}_g$;
end
Algorithm 1: Minibatch SGD for GAN optimization

[6] Salimans T. et. al, Improved Techniques for Training GANs, 2016

Non-convergence issues: Mode collapse[7]

Definition (mode collapse)

Mode collapse is a problem that occurs when the generator learns to map several different input $\mathbf{z}$ values to the same output point.

Example:

Explanation: Mode collapse may arise because the *maximin* solution to the GAN game is different from the *minimax* solution.

$$\min_G \max_D V(G, D), \quad \text{the original order,}$$

$$\max_D \min_G V(G, D), \quad \text{the exchanged order,}$$

$$\arg\min_{p_g} \mathbb{E}_{\mathbf{x} \sim p_g(\mathbf{x})} \log(1 - D(\mathbf{x})) = \delta(\mathbf{x} - \mathbf{x}^*),$$

$$\mathbf{x}^* \in \arg\min_{\mathbf{x}} \log(1 - D(\mathbf{x})).$$

[7] Goodfellow I., NIPS 2016 Tutorial: Generative Adversarial Networks, 2016

Unrolled GANs[8]

Challenge: mode collapse problem.

Solution: introduce unrolling optimization of the discriminator objective during training.

$$\boldsymbol{\theta}_g^* = \arg\min_{\boldsymbol{\theta}_g} V(\boldsymbol{\theta}_g, \boldsymbol{\theta}_d^*),$$
$$\text{s.t.} \quad \boldsymbol{\theta}_d^* = \arg\max_{\boldsymbol{\theta}_d} V(\boldsymbol{\theta}_g, \boldsymbol{\theta}_d).$$

Unrolling GANs:

$$\boldsymbol{\theta}_d^{k+1} = \boldsymbol{\theta}_d^k + \eta_k \frac{\partial V(\boldsymbol{\theta}_g, \boldsymbol{\theta}_d^k)}{\partial \boldsymbol{\theta}_d},$$
$$\boldsymbol{\theta}_d^*(\boldsymbol{\theta}_g) = \lim_{k \to \infty} \boldsymbol{\theta}_d^k,$$
$$\boldsymbol{\theta}_g \leftarrow \boldsymbol{\theta}_g - \eta \frac{dV(\boldsymbol{\theta}_g, \boldsymbol{\theta}_d^K(\boldsymbol{\theta}_g))}{d\boldsymbol{\theta}_g}.$$

Deriving the gradient:

$$\frac{dV(\boldsymbol{\theta}_g, \boldsymbol{\theta}_d^K(\boldsymbol{\theta}_g))}{d\boldsymbol{\theta}_g} =$$
$$\underbrace{\frac{\partial V(\boldsymbol{\theta}_g, \boldsymbol{\theta}_d^K)}{\partial \boldsymbol{\theta}_g}}_{\text{standard GAN}} + \underbrace{\frac{\partial V(\boldsymbol{\theta}_g, \boldsymbol{\theta}_d^K)}{\partial \boldsymbol{\theta}_d} \frac{d\boldsymbol{\theta}_d^K(\boldsymbol{\theta}_g)}{d\boldsymbol{\theta}_g}}_{\text{discriminator's reaction}}$$

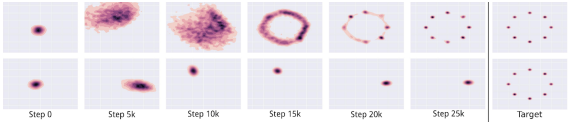
Note: as $K \to \infty$, $\boldsymbol{\theta}_d^K \to \boldsymbol{\theta}_d^*(\boldsymbol{\theta}_g)$, and then the second term goes to zero.

Note: in practice the use of an automatic differentiation package means that the implementation of differentiating through optimization does not need to be programmed explicitly.

[8] Metz L. et. al, Unrolled Generative Adversarial Networks, 2017

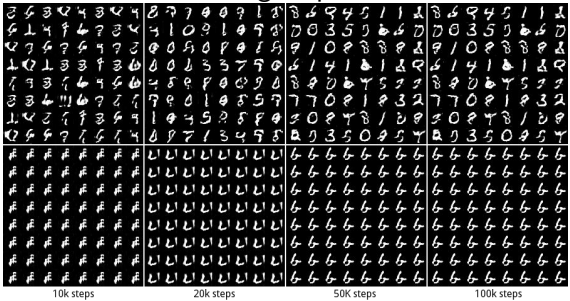
Unrolled GANs: evaluation

2D mixture of Gaussians dataset: The top row shows training for a GAN with 10 unrolling steps. The bottom row shows vanilla GAN training.



The generator of the unrolled GAN quickly spreads out and converges to the target distribution.

MNIST dataset: The top row was run with 20 unrolling steps. The bottom row is a standard GAN, with 0 unrolling steps.



It could be clearly seen that without unrolling the model quickly collapses.

Energy-based GAN[9]

Challenge: the original objective function and optimization procedure suffer from gradient vanishing and mode collapse problems.

Solution: re-design the objective function. Assume that D produces non-negative values. Given $m > 0$.

$$\mathcal{L}_D(\mathbf{x}, \mathbf{z}) = D(\mathbf{x}) + (m - D(G(\mathbf{z})))_+,$$
$$\mathcal{L}_G(\mathbf{z}) = D(G(\mathbf{z})),$$
$$\min_D V(G, D) = \mathbb{E}_{\mathbf{x}\sim\pi(\mathbf{x})}\mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}\mathcal{L}_D(\mathbf{x}, \mathbf{z}),$$
$$\min_G U(G, D) = \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}\mathcal{L}_G(\mathbf{z}).$$

Note: the gradients are neither vanished nor exploded.

Theorem

If (D^*, G^*) is a Nash equilibrium of the system ($D^* \in \arg\min_D V(G^*, D)$ and $G^* \in \arg\min_G U(G, D^*)$), then $\pi(.) = p_g(.)$ and $V(D^*, G^*) = m$.

Repelling regularizer: keeps the model from producing samples that are clustered in one or only few modes of $\pi(.)$.

$$R(\mathbf{h}_{1:M}) = \frac{2}{M(M-1)} \sum_{i<j} \cos(\mathbf{h}_i, \mathbf{h}_j)^2,$$

where $\mathbf{h}$ denotes a hidden representation from the generator ($\mathcal{L}_G \leftarrow \mathcal{L}_G - R$).

[9] Zhao J. et. al, Energy-Based Generative Adversarial Networks, 2017

EBGAN: evaluation

MNIST dataset: Left: Best GAN model; Middle: Best EBGAN model. Right: Best EBGAN-PT model (with repelling regularizer).

We see that samples drawn from EBGAN family are more realistic than those that were drawn from the vanilla GAN.

Wasserstein GAN[10]

Challenge: standard GANs suffer from training instabilities and mode collapse.

Solution: introduce the Earth Mover (EM) distance. This is a much more sensible cost function for GAN problem than the Jensen-Shannon divergence.

Example: Let $z \sim \mathcal{U}[0, 1]$, let $\mathbb{P}_0$ be the distribution of $(0, z)$ and let $\mathbb{P}_\theta$ be the distribution of (θ, z). It is easy to see that:

$$W(\mathbb{P}_0, \mathbb{P}_\theta) = |\theta|,$$
$$\mathrm{JS}(\mathbb{P}_0, \mathbb{P}_\theta) = [\theta \neq 0] \log 2,$$
$$\mathrm{KL}(\mathbb{P}_0, \mathbb{P}_\theta) = \mathrm{KL}(\mathbb{P}_\theta, \mathbb{P}_0) = [\theta \neq 0](+\infty).$$

Definition (Earth-Mover distance, Wasserstein-1)

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma} \|x - y\|_2,$$

where $\Pi(\mathbb{P}_r, \mathbb{P}_g)$ is the set of all joint distributions whose marginals are respectively $\mathbb{P}_r, \mathbb{P}_g$.

Theorem

Let $g_{\boldsymbol{\theta}}(.)$ be a feedforward neural network and $p(\mathbf{z})$ such that $\mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})} \|\mathbf{z}\| < \infty$. Then $W(\mathbb{P}_{\text{data}}, \mathbb{P}_\theta)$ is continuous everywhere and differentiable almost everywhere. Here $\mathbb{P}_\theta$ is the distribution of $g_{\boldsymbol{\theta}}(\mathbf{z})$, $\mathbf{z} \sim p(\mathbf{z})$.

[10] Arjovsky M. et. al, Wasserstein Generative Adversarial Networks, 2017

WGAN: optimization

Kantorovich-Rubinstein duality: note that the infimum in is highly intractable, therefore:

$$W(\mathbb{P}_{\text{data}}, \mathbb{P}_\theta) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{\mathbf{x}\sim\pi(\mathbf{x})} f(\mathbf{x}) - \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_\theta} f(\mathbf{x}),$$

$$\|f\|_L \leq 1 \Leftrightarrow |f(\mathbf{x}) - f(\mathbf{y})| \leq \|\mathbf{x} - \mathbf{y}\|_2.$$

Finally, we parametrize f with $\mathbf{w}$ and optimize $K \cdot W(.,.)$ subject to $\|f\|_L \leq K$:

$$\max_{\mathbf{w}\in\mathcal{W}} \mathbb{E}_{\mathbf{x}\sim\pi(\mathbf{x})} f_{\mathbf{w}}(\mathbf{x}) - \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_\theta} f_{\mathbf{w}}(\mathbf{x}).$$

Weight clipping: the weights $\mathbf{w}$ are constrained to lie within $[-c, c]$. If c is small $\Rightarrow$ vanishing gradients. If c is large $\Rightarrow$ slow convergence.

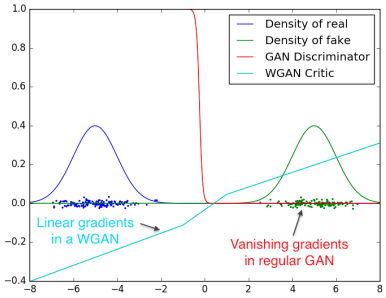
Gradient w.r.t. θ: given the optimal $\mathbf{w}^*$. Then

$$\nabla_\theta W(\mathbb{P}_{\text{data}}, \mathbb{P}_\theta) \approx -\mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})} \nabla_\theta f_{\mathbf{w}^*}(g_\theta(\mathbf{z})).$$

while *not converged* **do**
  foreach $t = 0 \dots n_{critic}$ **do**
    $\mathbf{x}_{1:M} \sim \mathbb{P}_{\text{data}}$, $\mathbf{z}_{1:M} \sim p(\mathbf{z})$;
    $\hat{\mathbf{g}}_{\mathbf{w}} \leftarrow \nabla_{\mathbf{w}} \frac{1}{M}[\sum_{i=1}^{M}(f_w(\mathbf{x}_i) - f_w(g_\theta(\mathbf{z}_i)))]$;
    update $\mathbf{w}$ with $\hat{\mathbf{g}}_{\mathbf{w}}$;
    $\mathbf{w} \leftarrow \text{clip}(\mathbf{w}, -c, c)$;
  end
  $\mathbf{z}_{1:M} \sim p(\mathbf{z})$;
  $\hat{g}_\theta \leftarrow -\nabla_\theta \frac{1}{m} \sum_{i=1}^{m} f_w(g_\theta(\mathbf{z}_i))$;
  update θ with $\hat{\mathbf{g}}_\theta$;
end

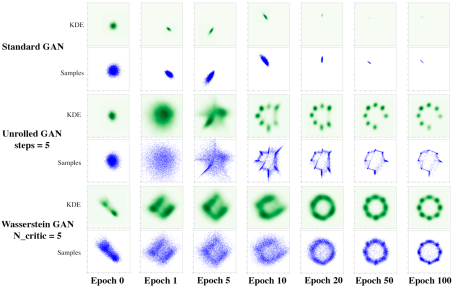
WGAN: evaluation

Toy experiment: learning to differentiate two Gaussians:



As we can see, the traditional GAN discriminator saturates and results in vanishing gradients. The proposed WGAN critic provides very clean gradients on all parts of the space.

Toy experiment: learning a mixture of 8 gaussians:



These plot reveal that WGAN is able to learn the distribution without mode collapse. Moreover, WGAN seems to capture first the low dimensional structure of the data (the approximate circle).

InfoGAN: Mutual Information for Inducing Latent Codes[11]

Challenge: it is possible that the noise will be used by the generator in a highly entangled way, causing the individual dimensions of $\mathbf{z}$ to not correspond to semantic features of the data.
Solution: decompose the input vector into a source noize $\mathbf{z}$ and a latent code $\mathbf{c}$.
Let $P(c_1, \ldots, c_L) = \prod_{i=1}^{L} p(c_i)$. The *mutual information* between two random variables

$$I(X; Y) = H(X) - \underbrace{H(X|Y)}_{-\mathbb{E}[\log p(y|x)]}.$$

The information-regularized minimax game

$$\min_G \max_D V(G, D) - \lambda I(\mathbf{c}; G(\mathbf{z}, \mathbf{c})).$$

Variational Mutual Information Maximization

$$I(\mathbf{c}; G(\mathbf{z}, \mathbf{c})) = H(\mathbf{c}) + \mathbb{E}[\log p(\mathbf{c}|G(\mathbf{z}, \mathbf{c}))].$$

The second term is intractable.

Lemma

For random variables X, Y and function $f(x, y)$ under suitable regularity conditions

$$\mathbb{E}_{x\sim X, y\sim Y|x}[f(x, y)] = \mathbb{E}_{x\sim X, y\sim Y|x, x'\sim X|y}[f(x', y)].$$

[11] Chen Xi, et. al, InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets, 2016

InfoGAN: Variational MI maximization

proof:

$$
\begin{aligned}
&\mathbb{E}_{x\sim X, y\sim Y|x}[f(x,y)] \\
&= \int_x P(x) \int_y P(y|x) f(x,y) dy dx \\
&= \int_x \int_y P(x,y) f(x,y) dy dx \\
&= \int_y p(y) \int_{x'} p(x'|y) f(x',y) dx' dy \\
&= \int_x \int_y p(x,y) \int_{x'} p(x'|y) f(x',y) dx' dy dx \\
&= \int_x p(x) \int_y p(y|x) \int_{x'} p(x'|y) f(x',y) dx' dy dx \\
&= \mathbb{E}_{x\sim X, y\sim Y|x, x'\sim X|y}[f(x',y)].
\end{aligned}
$$

Variational MI bound. Let $Q(\mathbf{c}|\mathbf{x})$ be a variational distribution.

$$
\begin{aligned}
&I(\mathbf{c}; G(\mathbf{z},\mathbf{c})) \\
&= H(\mathbf{c}) - H(\mathbf{c}|G(\mathbf{z},\mathbf{c})) \\
&= H(\mathbf{c}) + \mathbb{E}_{\mathbf{x}\sim g(\mathbf{z},\mathbf{c}), \mathbf{c}'\sim p(\mathbf{c}'|\mathbf{x})}[\log p(\mathbf{c}'|G(\mathbf{z},\mathbf{c}))] \\
&= \mathbb{E}_{\mathbf{x}\sim g(\mathbf{z},\mathbf{c})}[\mathrm{KL}(p(.|\mathbf{x})||Q(.|\mathbf{x})) + \\
&+ \mathbb{E}_{\mathbf{c}'\sim P(\mathbf{c}|\mathbf{x})}[\log Q(\mathbf{c}'|\mathbf{x})]] + H(\mathbf{c}) \\
&\geq \mathbb{E}_{\mathbf{x}\sim g(\mathbf{z},\mathbf{c}), \mathbf{c}'\sim p(\mathbf{c}|\mathbf{x})}[\log Q(\mathbf{c}'|\mathbf{x})] + H(\mathbf{c}) \\
&\overset{Lem}{=} \mathbb{E}_{\mathbf{c}\sim P(\mathbf{c}), \mathbf{z}\sim p(\mathbf{z})}[\log Q(\mathbf{c}|G(\mathbf{z},\mathbf{c}))].
\end{aligned}
$$

InfoGAN: Experimental Evaluation

Disentangled Representation. Let $c_1 \sim \mathrm{Cat}(K = 10, p = 0.1)$ and $c_2, c_3 \sim \mathcal{U}[-1, 1]$. Let also $Q(\mathbf{c}|\mathbf{x})$ and $D(.)$ share the same convolutional layers.

Results:

- discrete code c_1 captures drastic change in shape.
- Continuous codes c_2, c_3 capture continuous variations in style: c_2 models rotation of digits and c_3 controls the width.

(a) Varying c_1 on InfoGAN (Digit type)

(b) Varying c_1 on regular GAN (No clear meaning)

(c) Varying c_2 from -2 to 2 on InfoGAN (Rotation)

(d) Varying c_3 from -2 to 2 on InfoGAN (Width)

Inception score[12]

Challenge: human annotation is required to evaluate the samples from GAN.
Solution: introduce an automatic method that correlate with human evaluation.
Use the Inception model to gen the conditional label distribution for each sample $p(y|\mathbf{x})$.
Natural requirements:

- Images that contain meaningful objects should have $p(y|\mathbf{x})$ with low entropy.
- Moreover, it is expected that the model generates varied images, so $p(y) = \mathbb{E}_{\mathbf{x}\sim p_g(\mathbf{x})} p(y|\mathbf{x})$ should have high entropy.

Combining these two requirements:

$$\begin{aligned}\text{IS} &= \exp(\mathbb{E}_{\mathbf{x}\sim p_g(\mathbf{x})}\text{KL}(p(y|\mathbf{x})||p(y))) \\ &= \exp(\mathbb{E}_{\mathbf{x}\sim p_g(\mathbf{x})}[\sum_{y\in\mathcal{Y}} p(y|\mathbf{x})\log p(y|\mathbf{x})] - \sum_{y\in\mathcal{Y}} p(y)\log p(y)).\end{aligned}$$

Note: A higher inception score is interpreted as better.

[12] Salimans T. et. al, Improved Techniques for Training GANs, 2016

Frechet Inception Distance[13]

Challenge: Inception score is poorly consistent with the noise level.

Solution: introduce a metric that captures the disturbance level very well.

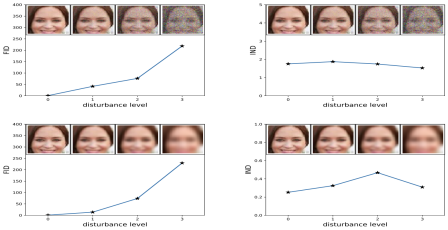
We call Frechet Inception Distance between the gaussian with parameters $(\mathbf{m}_\pi, \Sigma_\pi)$ obtained from the data distribution π and the gaussian with parameters $(\mathbf{m}_{\boldsymbol{\theta}}, \Sigma_{\boldsymbol{\theta}})$ obtained from $p_{\boldsymbol{\theta}}$:

$$\mathrm{FID}(\pi, p_{\boldsymbol{\theta}}) = \|\mathbf{m}_\pi - \mathbf{m}_{\boldsymbol{\theta}}\|_2^2 + \mathrm{tr}(\Sigma_\pi + \Sigma_{\boldsymbol{\theta}} - 2(\Sigma_\pi \Sigma_{\boldsymbol{\theta}})^{1/2}).$$

Note: this is also known as Wasserstein-2 distance between two gaussians:

$$\mathrm{FID}^2(\pi, p_{\boldsymbol{\theta}})^2 = \inf_{\gamma \in \Pi(\pi, p_{\boldsymbol{\theta}})} \mathbb{E}_{(\mathbf{x},\mathbf{y})\in\gamma} \|\mathbf{x} - \mathbf{y}\|_2^2.$$

comparison of FID and IS:



The FID captures the disturbance level very well by monotonically increasing whereas the Inception Score fluctuates, stays flat or even, in the worst case, decreases.

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

GANs' Representative variants: Cycle GAN[14]

Challenge: for many image-to-image translation tasks paired training data will not be available.

Solution: introduce a model that expoits "cycle consistency" property.

The goal is to learn two mapping functions between domains X and Y, distributed as $\mathbf{x} \sim \pi(\mathbf{x})$, $\mathbf{y} \sim \pi(\mathbf{y})$. Introduce two generators $G : X \to Y$ and $F : Y \to X$ and discriminators $D_X(.)$ and $D_Y(.)$.

Adversarial Loss for G:

$$\mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) = \mathbb{E}_{\mathbf{y}\sim\pi(\mathbf{y})} \log D_Y(\mathbf{y}) + \mathbb{E}_{\mathbf{x}\sim\pi(\mathbf{x})} \log(1 - D_Y(G(\mathbf{x}))).$$

Challenge: a network can map the same set of input images to any random permutation of images in the target domain.

Solution: Cycle Consistency Loss

$$\mathcal{L}_{\text{cyc}}(G, F) = \mathbb{E}_{\mathbf{x}\sim\pi(\mathbf{x})} \|F(G(\mathbf{x})) - \mathbf{x}\|_1 + \mathbb{E}_{\mathbf{y}\sim\pi(\mathbf{y})} \|G(F(\mathbf{y})) - \mathbf{y}\|_1.$$

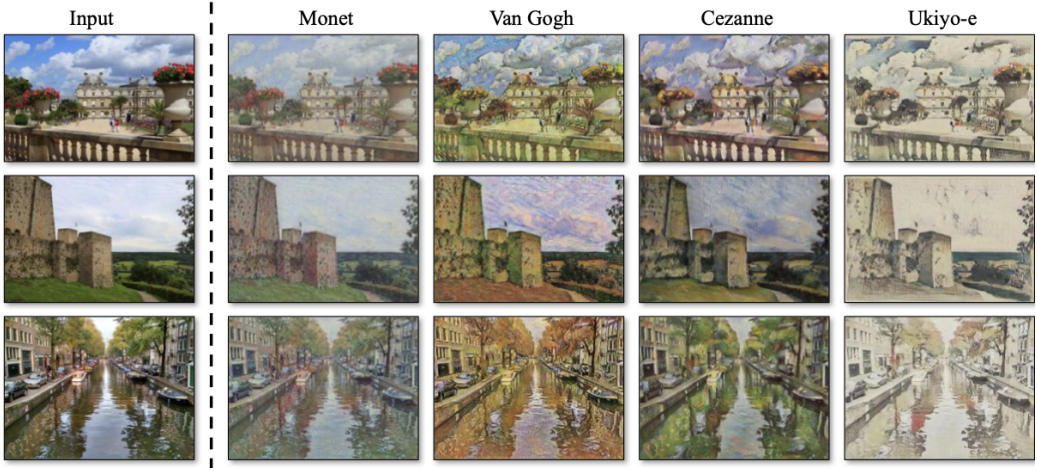
Full objective:

$$\mathcal{L}_{\text{total}}(G, F, D_X, D_Y) = \mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) + \mathcal{L}_{\text{GAN}}(F, D_X, Y, X) + \lambda \mathcal{L}_{\text{cyc}}(G, F).$$

[14] Zhu J. et. al, Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks, 2017

Cycle GAN: evaluation

Style Transfer:



We see that the proposed method achieved compelling results.

Summary

- Autoregressive models: PixelCNN, LMConv
- Generative Adversarial Networks: theoretical results, vanishing gradients, mode collapse
- Generative Adversarial Networks: Unrolled GAN, Energy-based GAN, Wasserstein GAN
- Evaluation metrics for GAN: Inception Score, Frechet Inception Distance
- GAN's representative: Cycle GAN