

رَبِّ الْعَالَمِينَ



یادگیری عمیق

مدرس: محمدرضا محمدی

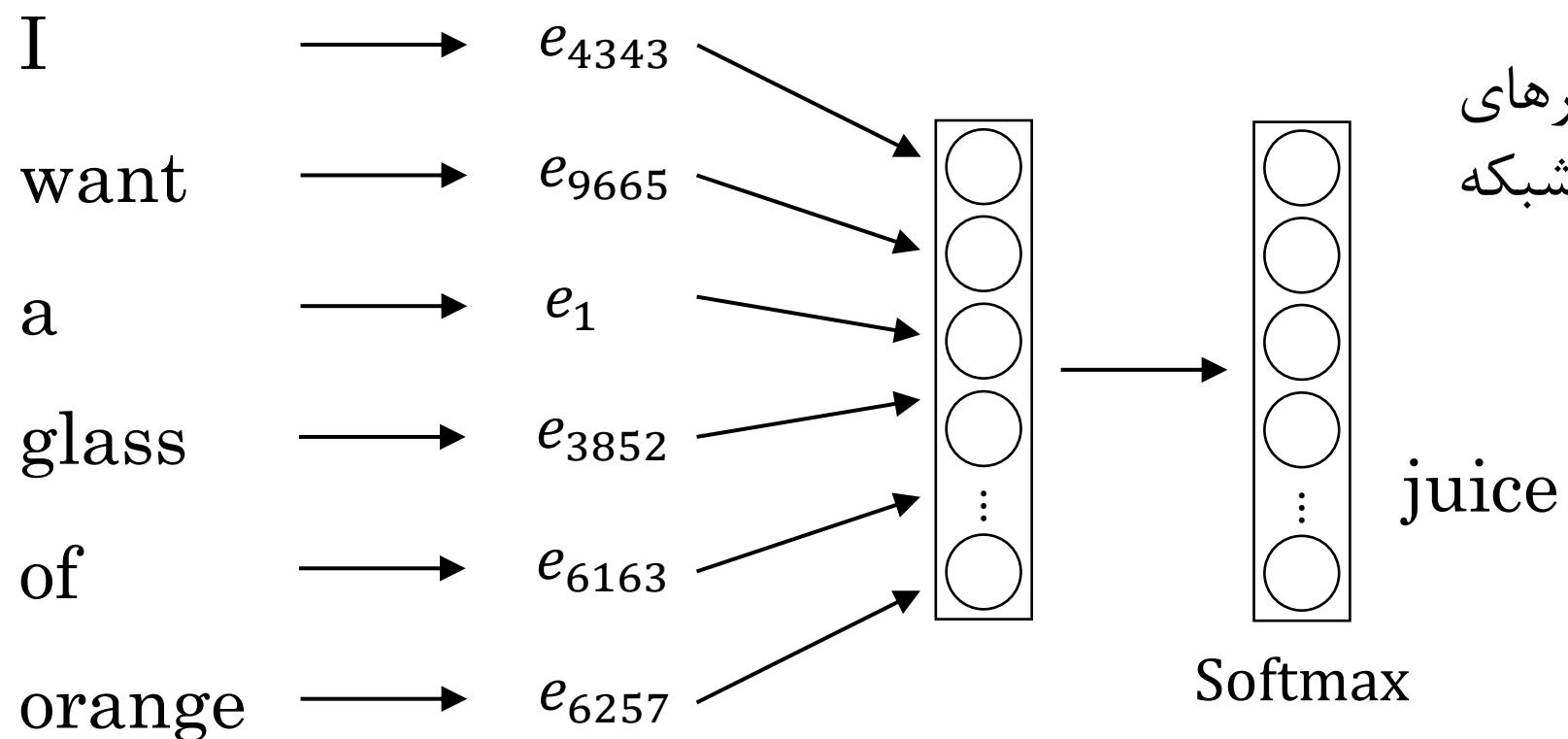
بهار ۱۴۰۲

یادگیری بازنمایی

Representation Learning

مدل زبان طبیعی

- در این مدل می‌خواهیم کلمه بعدی را پیش‌بینی کنیم



- هدف ما دستیابی به بردارهای جانمایی است و وزن‌های دیگر شبکه هدف نیستند

جفت‌های Context/Target دیگر

- جمله نمونه

I want a glass of orange juice to go along with my cereal. -

- Context

- چهار کلمه قبل

- چهار کلمه قبل و بعد

- یک کلمه قبل

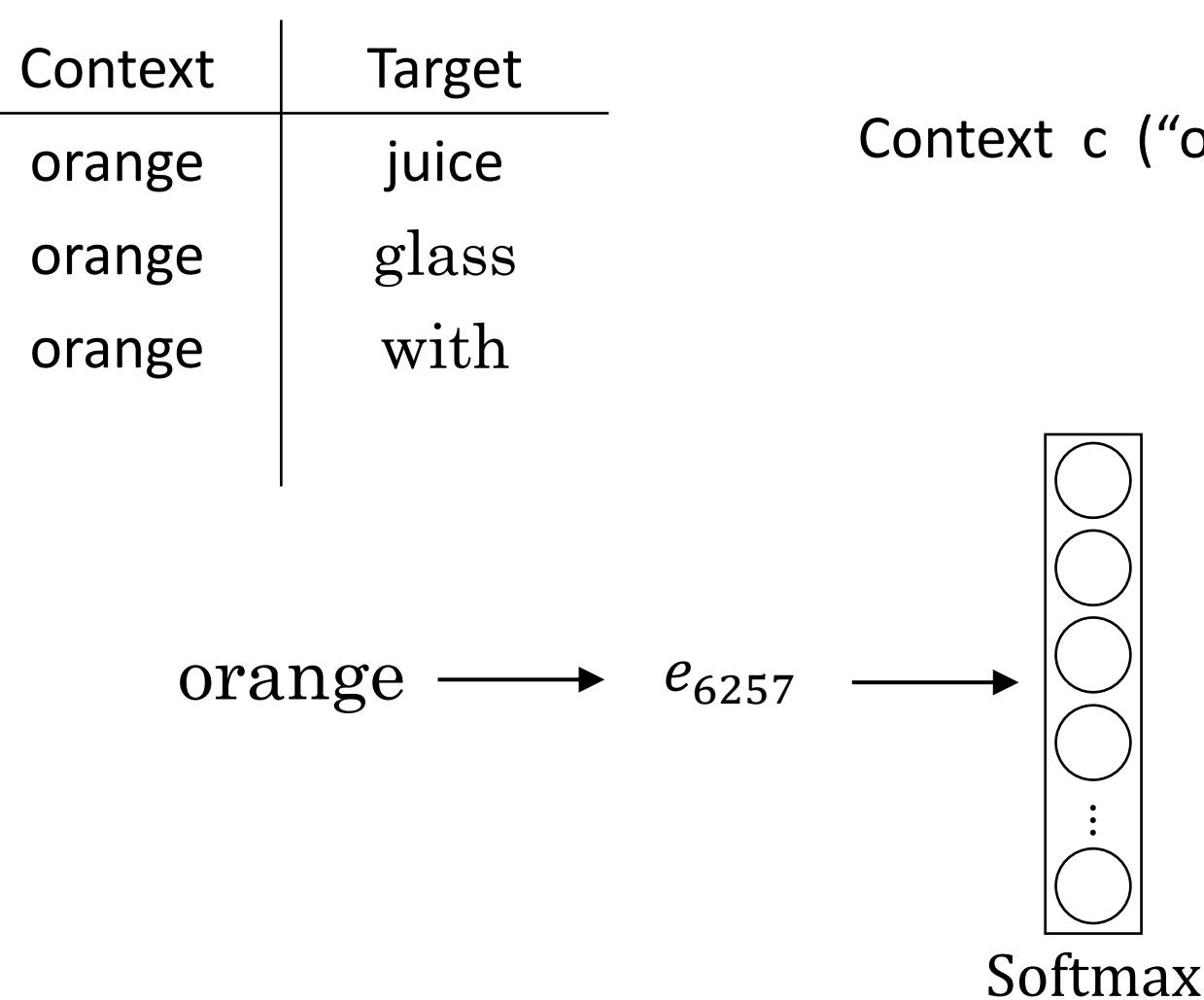
- یک کلمه در نزدیکی

Word2Vec

Context	Target	Skip-grams
orange		<ul style="list-style-type: none">- بجای اینکه همیشه از آخرین کلمات استفاده شود▪ به طور تصادفی یک کلمه را به عنوان Context انتخاب کنید▪ به طور تصادفی کلمه دیگری را در پنجره‌ای اطراف آن انتخاب کنید- یک مسئله یادگیری با ناظر تنظیم کنید که در آن:<ul style="list-style-type: none">▪ با توجه به کلمه Context▪ کلمه‌ای که به طور تصادفی انتخاب شده است را پیش‌بینی کنید

I want a glass of orange juice to go along with my cereal.

Word2Vec



- اگر ۱۰۰۰ کلمه یکتا داشته باشیم •

$$\mathcal{L}_{ce} = - \sum_{j=1}^{10,000} y_j \log \hat{y}_j$$

$$p(t|c) = \frac{e^{w_t^T e_c + b_t}}{\sum_{j=1}^{10,000} e^{w_j^T e_c + b_j}}$$

Word2Vec

- مشکلات Softmax

- مشکل اصلی سرعت محاسبات است

- مخصوصاً برای مجموعه واژگان بسیار بزرگ (مثلاً ۱۰۰،۰۰۰ یا ۱،۰۰۰،۰۰۰ واژه)

$$p(t|c) = \frac{e^{w_t^T e_c + b_t}}{\sum_{j=1}^{10,000} e^{w_j^T e_c + b_j}}$$

Word2Vec

Context	Word	Target

- نمونه برداری منفی (Negative sampling)

- یک مسئله یادگیری باناظر جدید
- یک جفت کلمه را ورودی بگیرد
- پیش‌بینی کند که یک جفت Context-Target است؟
- بنابراین مسئله این است که پیش‌بینی کند آیا دو کلمه در کنار هم استفاده می‌شوند یا خیر
- مقدار پیشنهادی برای k :
- ۵ تا ۲۰ برای مجموعه‌داده‌های کوچکتر
- ۲ تا ۵ برای مجموعه‌داده‌های بزرگتر

I want a glass of orange juice to go along with my cereal.

Word2Vec

Context	Word	Target
orange	juice	1
orange	king	0
orange	book	0
orange	the	0
orange	of	0



$$p(t|c) = \frac{e^{w_t^T e_c + b_t}}{\sum_{j=1}^{10,000} e^{w_j^T e_c + b_j}}$$

$$p(y = 1|t, c) = \sigma(w_t^T e_c + b_t)$$

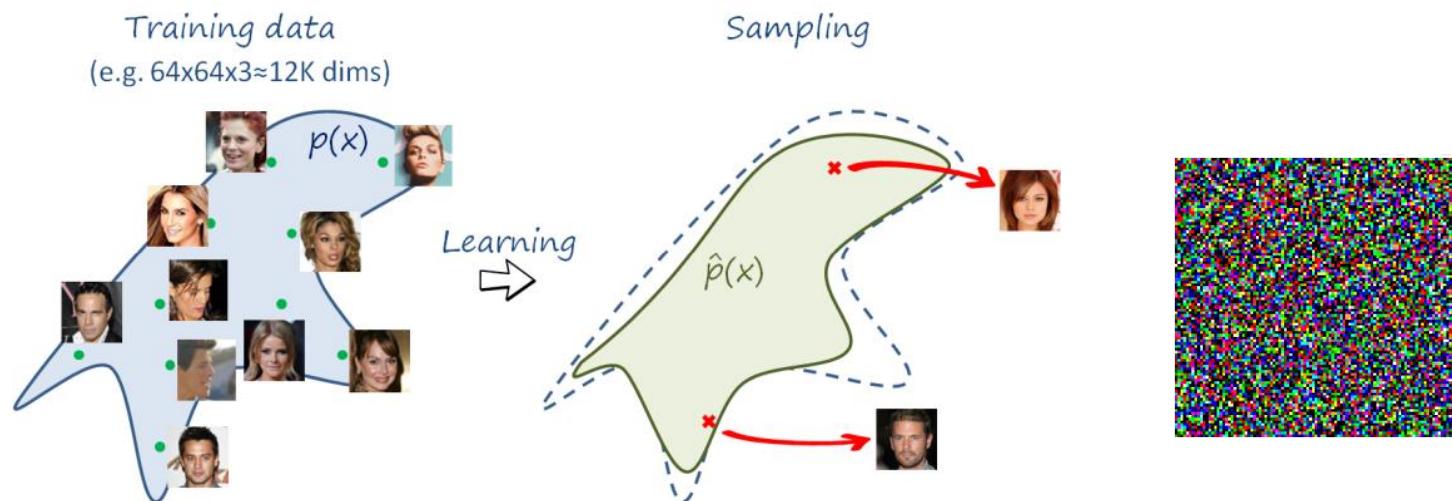
- نمونه برداری منفی

مدل‌های مول

Generative Models

مدل‌های مولد

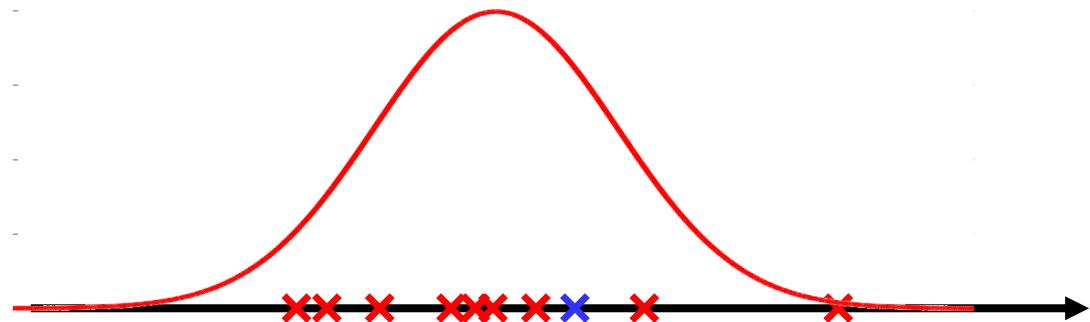
- مدل‌های مولد می‌توانند پس از مشاهده تصاویر آموزشی، تصاویر جدید از همان توزیع تولید کنند



- داده‌های آموزشی دارای توزیع $p_{data}(x)$ هستند
- هدف آموختن $p_{model}(x)$ مشابه با $p_{data}(x)$ است
- می‌توان از توزیع $p_{model}(x)$ نمونه تولید کرد

مدل‌های مولّد

- به طور مثال، فرض کنید می‌دانیم توزیع یک متغیر تصادفی گاووسی باشد و چندین نمونه از آن مشاهده شده باشد
- با استفاده از مشاهدات می‌توان پارامترهای مدل (میانگین و انحراف معیار) را تخمین زد
- با استفاده از تابع $m + \sigma \times randn()$ می‌توان از این توزیع نمونه جدید تولید کرد



مدل‌های مولّد

7	2	1	0	4
5	9	7	3	4
3	1	3	4	7
5	1	2	4	4

training data $\sim p_{\text{data}}(x)$

$$\Rightarrow p_{\text{model}}(x) = ? \Rightarrow$$

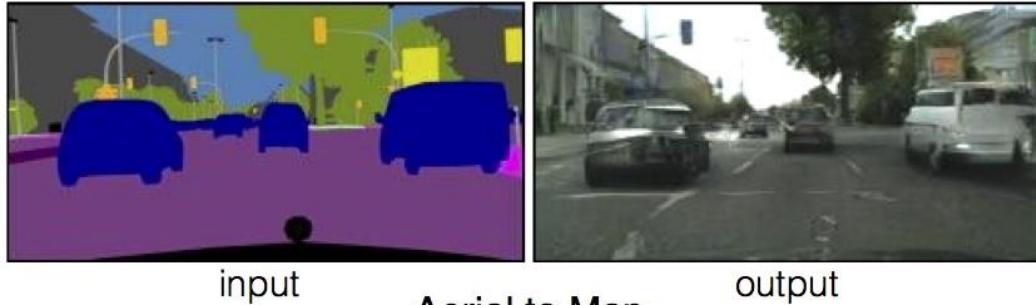
9	8	9	8	8
8	2	9	2	1
4	9	4	8	0
6	0	3	2	0

generated samples $\sim p_{\text{model}}(x)$

- مسئله اصلی تخمین چگالی احتمال است
 - تخمین صریح: به طور صریح مقدار $p_{\text{model}}(x)$ را تخمین می‌زند
 - تخمین ضمنی: مدلی را آموزش می‌بیند که بتواند از $p_{\text{model}}(x)$ نمونه تولید کند

کاربردهای مدل‌های مولد

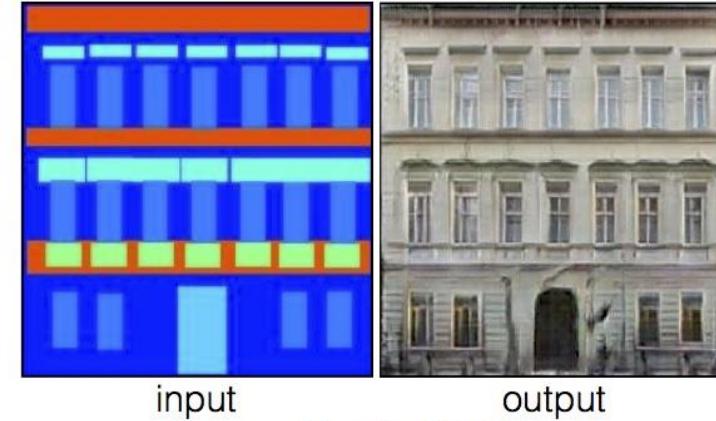
Labels to Street Scene



input

output

Labels to Facade



input

output

BW to Color



input

output

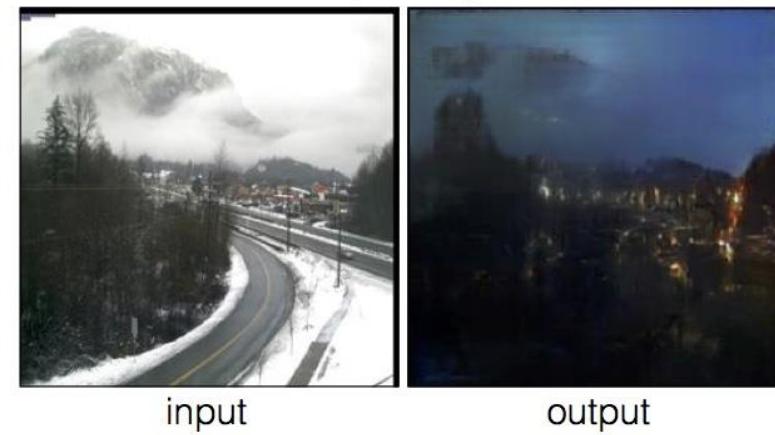
Aerial to Map



input

output

Day to Night



input

output

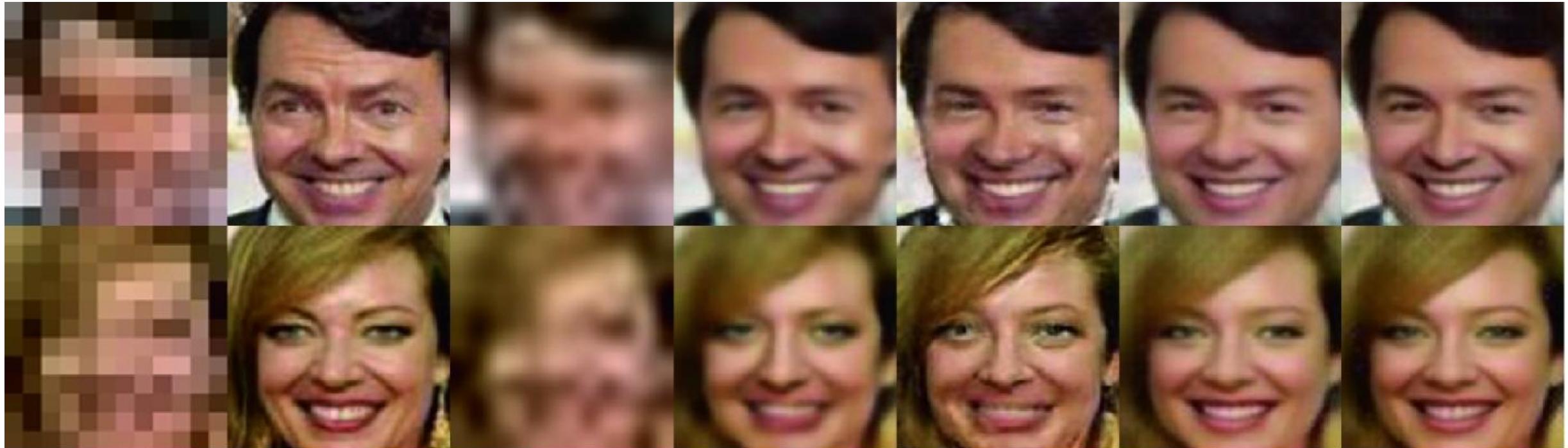
Edges to Photo



input

output

کاربردهای مدل‌های مولد



(a) LR

(b) HR

(c) Bicubic

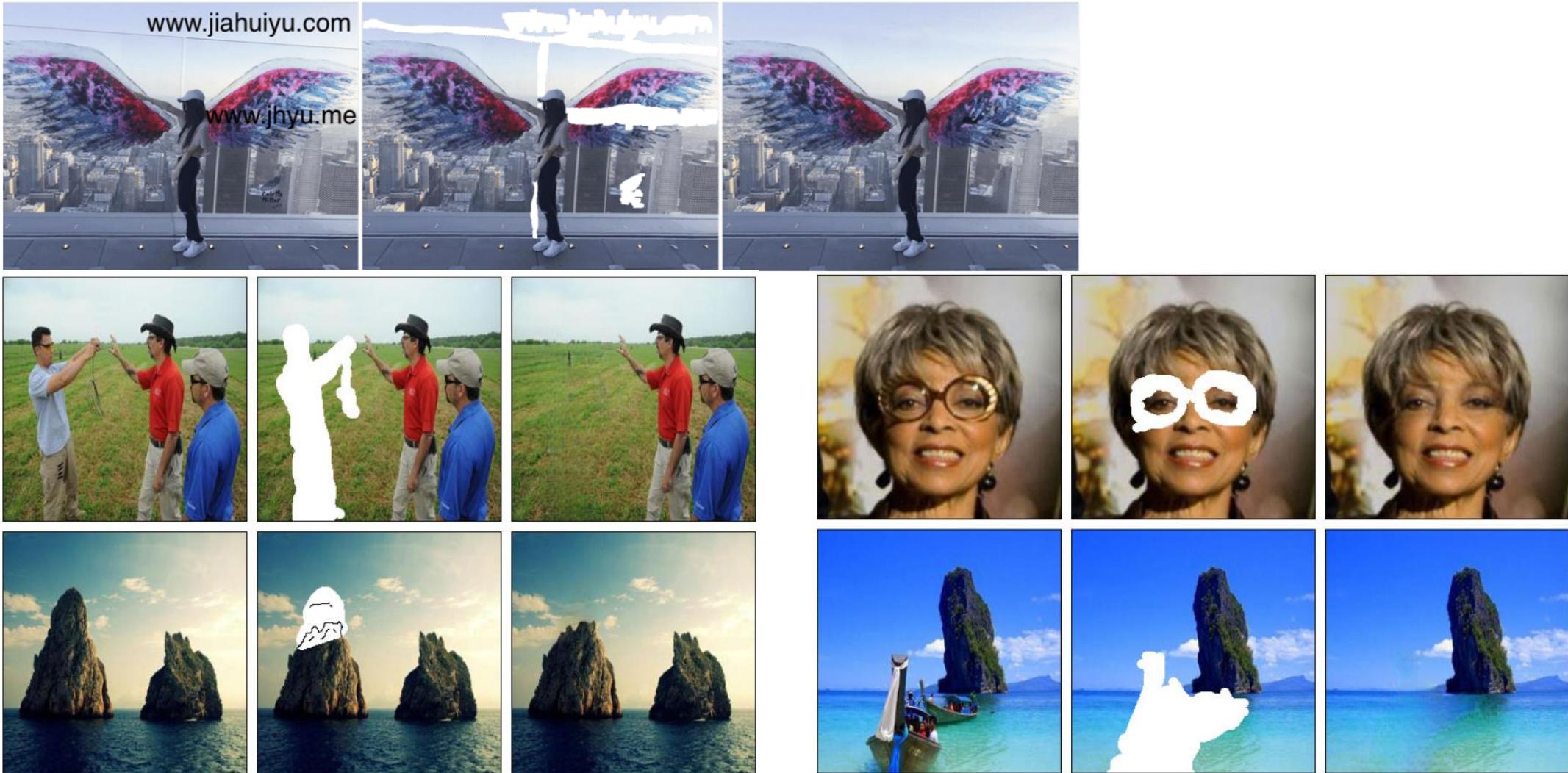
(d) AACNN

(e) AAGAN

(f) Our-Attr0

(g) Our-Attr5

کاربردهای مدل‌های مولد



شبکه‌های رقابتی مولد

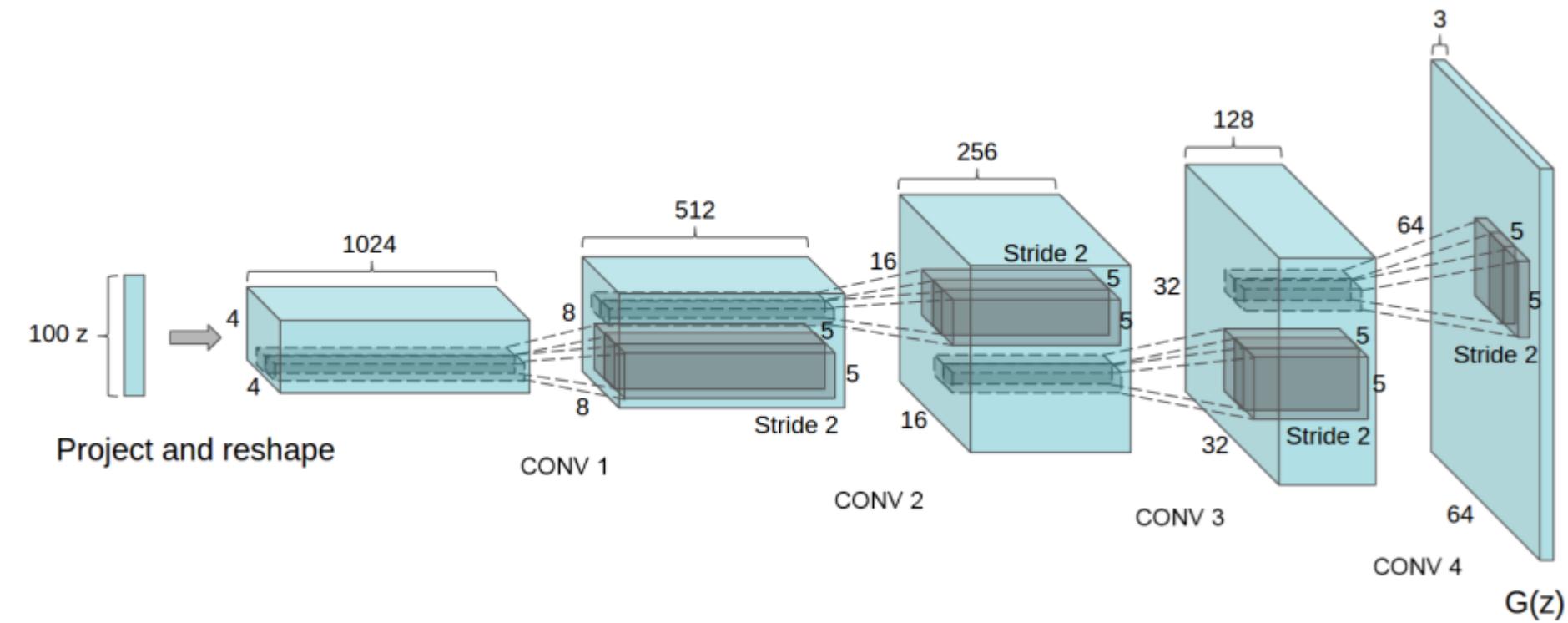
- در GAN (Generative Adversarial Networks)، مدل‌سازی صریح چگالی احتمال انجام نمی‌شود بلکه و تنها نمونه‌برداری از آن مد نظر قرار می‌گیرد
- در این روش، یک شبکه داده تصادفی تولید می‌کند و مشابهت آن با داده‌های آموزشی سنجیده می‌شود و شبکه به صورتی آموزش می‌بیند که بتواند داده‌هایی مشابه با داده‌های آموزشی تولید کند
- برای سنجش مشابهت داده‌های تولید شده با داده‌های آموزشی نیز از یک شبکه دیگر استفاده می‌شود
- فرآیند آموزش GAN مشابه با یک بازی دو نفره است که در آن یک بخش از شبکه داده‌های مصنوعی تولید می‌کند و بخش دوم از شبکه تلاش می‌کند تا ساختگی بودن داده‌ها را شناسایی کند

شبکه مولد

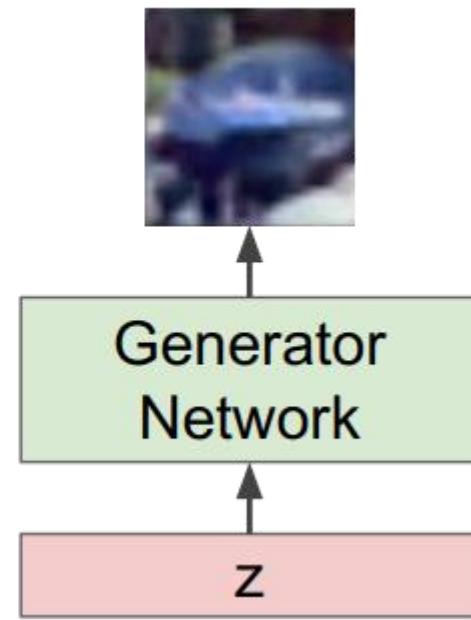
- شبکه مولد باید بتواند از یک توزیع احتمال با ابعاد بالا و بسیار پیچیده نمونه تولید کند
 - روش مستقیمی برای این منظور وجود ندارد!
- یک راه حل آن است که از یک توزیع ساده (مانند نویز تصادفی) نمونه تولید کرد و سپس تبدیلی را آموزش داد که آن را به توزیع داده‌های آموزشی تبدیل کند
 - برای آموزش این تابع تبدیل از شبکه‌های عصبی استفاده می‌کنیم

شبکه مولد

- یک راه حل آن است که از یک توزیع ساده (مانند نویز تصادفی) نمونه تولید کرد و سپس تبدیلی را آموزش داد که آن را به توزیع داده‌های آموزشی تبدیل کند
- نمونه‌ای از شبکه مولد



Output: Sample from
training distribution

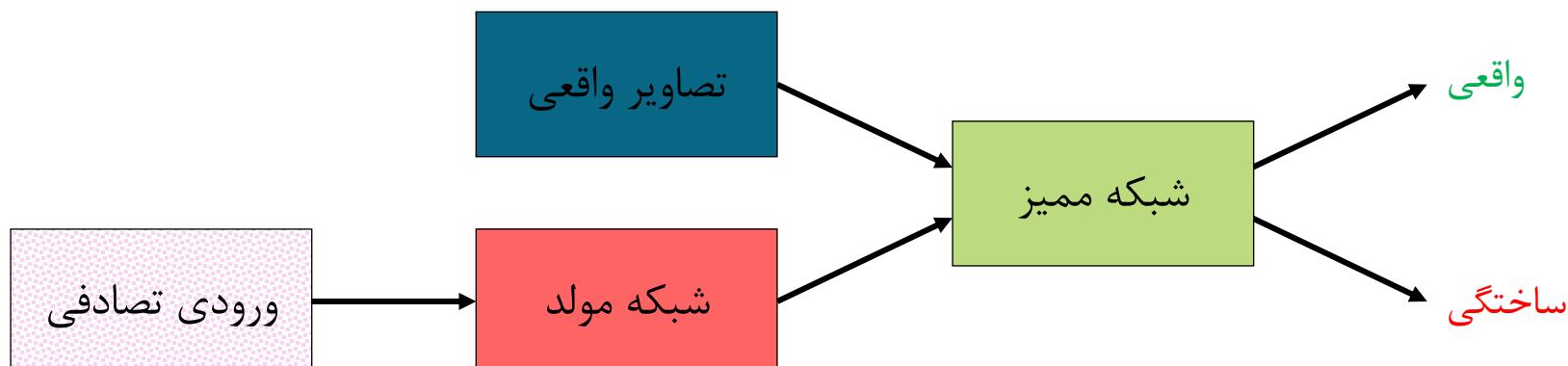


Input: Random noise

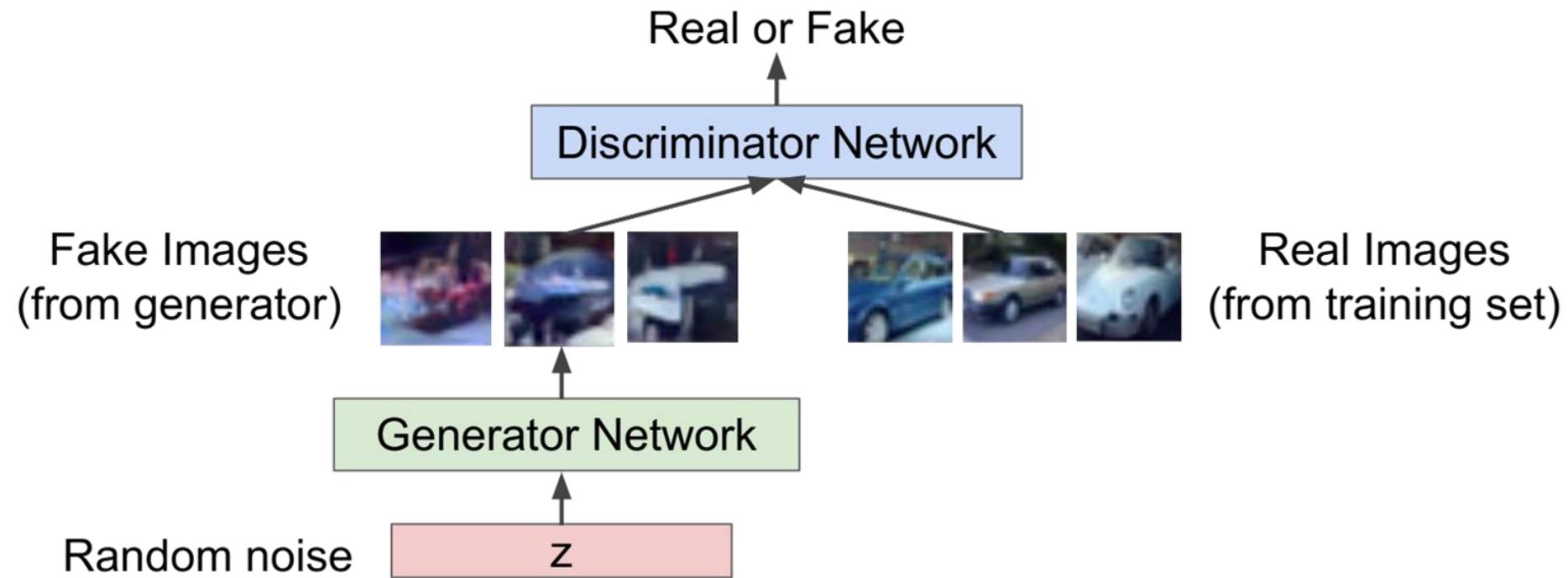
- چگونه پارامترهای شبکه را بهینه کنیم؟
- باید کیفیت تصاویر تولید شده را بسنجیم
- برای این منظور، از یک شبکه ممیز (discriminator) استفاده می‌کنیم

GAN

- آموزش GAN از ایده بازی دو نفره استفاده می‌کند
- شبکه مولد: تلاش می‌کند تصاویری تولید کند که به تصاویر واقعی شبیه باشند و شبکه دوم نتواند آنها را تشخیص دهد
- شبکه ممیز: تلاش می‌کند تا تصاویر واقعی و ساختگی را شناسایی کند



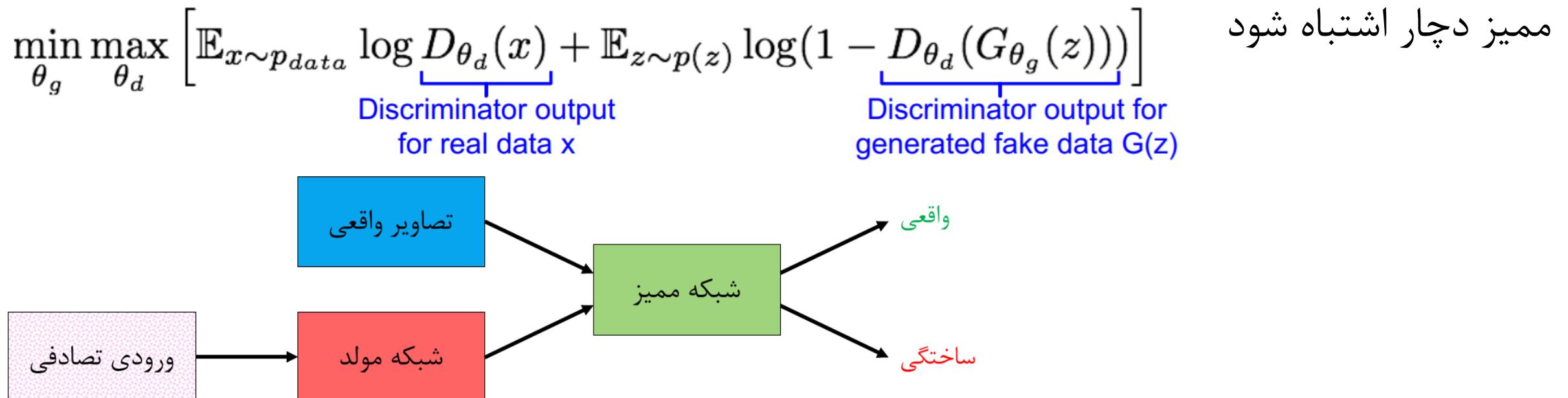
GAN



$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

GAN

- شبکه ممیز (با پارامترهای θ_d) می‌خواهد تابع هدف را بیشینه کند به گونه‌ای که $D(x)$ برای داده‌های واقعی نزدیک به ۱ و $D(G(z))$ برای داده‌های ساختگی نزدیک به صفر باشد
- شبکه مولد (با پارامترهای θ_g) می‌خواهد تابع هدف را کمینه کند تا $D(G(z))$ نزدیک به ۱ باشد و شبکه ممیز دچار اشتباہ شود



GAN

$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log \underbrace{D_{\theta_d}(x)}_{\text{Discriminator output for real data } x} + \mathbb{E}_{z \sim p(z)} \log(1 - \underbrace{D_{\theta_d}(G_{\theta_g}(z))}_{\text{Discriminator output for generated fake data } G(z)}) \right]$$

- روش پیشنهادی برای بهینه‌سازی پارامترهای شبکه به صورت تکراری:
 - گرادیان افزایشی برای شبکه ممیز

$$\max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

- گرادیان کاهشی برای شبکه مولد

$$\min_{\theta_g} \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z)))$$

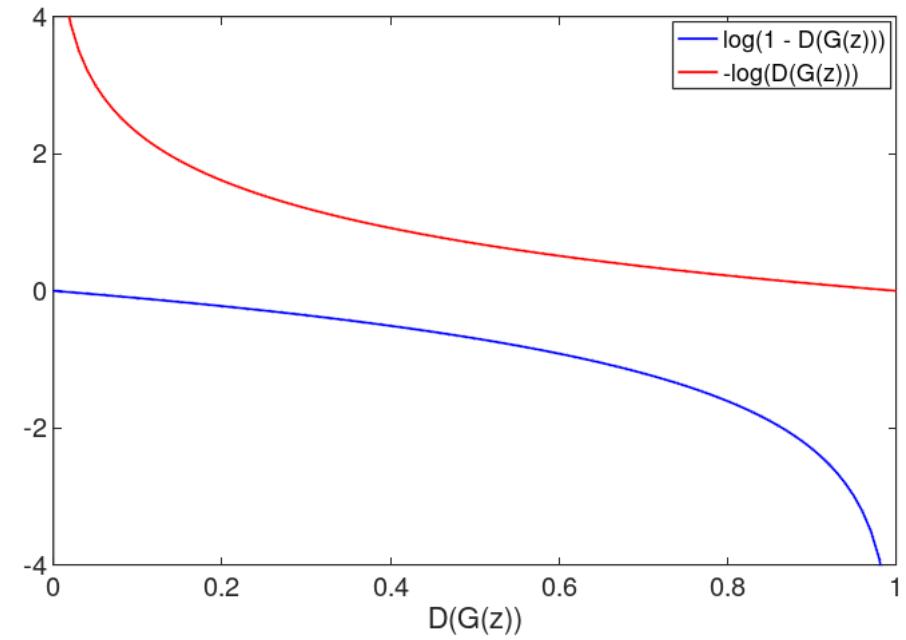
GAN

$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log \underbrace{D_{\theta_d}(x)}_{\text{Discriminator output for real data } x} + \mathbb{E}_{z \sim p(z)} \log(1 - \underbrace{D_{\theta_d}(G_{\theta_g}(z))}_{\text{Discriminator output for generated fake data } G(z)}) \right]$$

$$\max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

$$\min_{\theta_g} \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z)))$$

$$\max_{\theta_g} \mathbb{E}_{z \sim p(z)} \log(D_{\theta_d}(G_{\theta_g}(z)))$$



شبه کد آموزش GAN

for number of training iterations **do**

for k steps **do**

- Sample minibatch of m noise samples $\{\mathbf{z}^{(1)}, \dots, \mathbf{z}^{(m)}\}$ from noise prior $p_g(\mathbf{z})$.
- Sample minibatch of m examples $\{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\mathbf{x})$.
- Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^m \left[\log D_{\theta_d}(x^{(i)}) + \log(1 - D_{\theta_d}(G_{\theta_g}(z^{(i)}))) \right]$$

end for

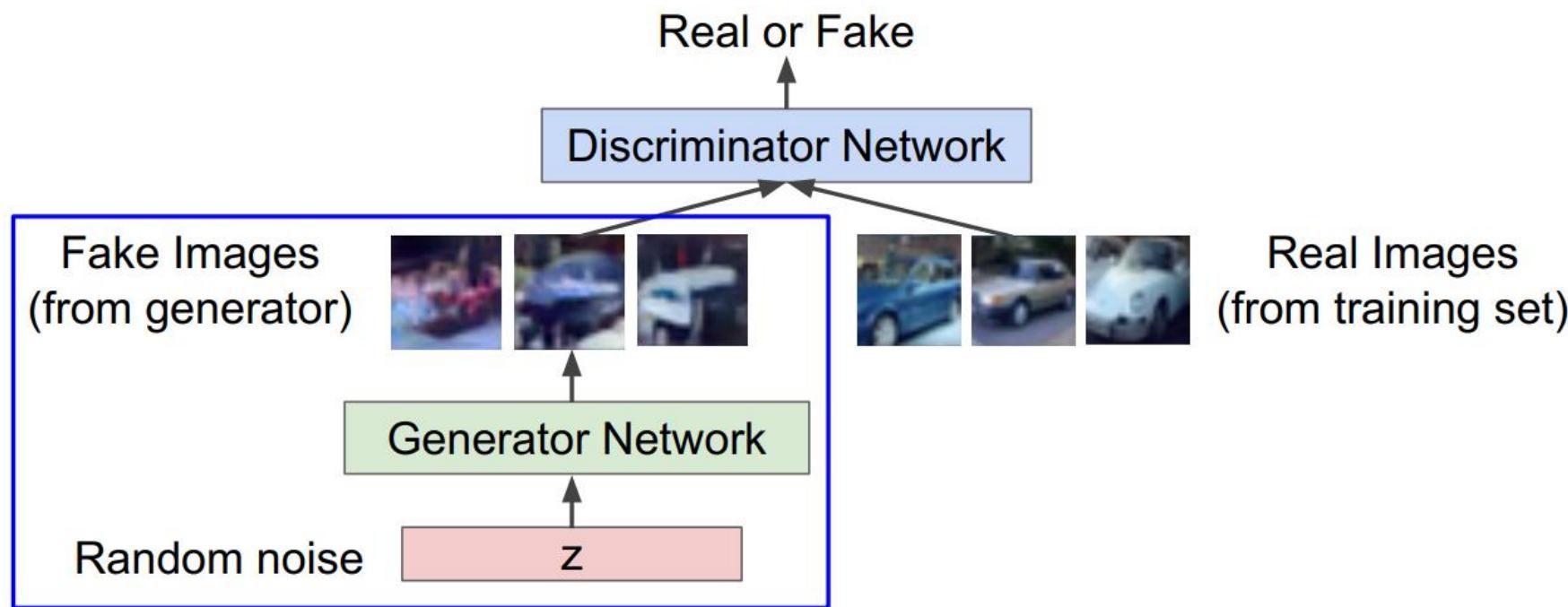
- Sample minibatch of m noise samples $\{\mathbf{z}^{(1)}, \dots, \mathbf{z}^{(m)}\}$ from noise prior $p_g(\mathbf{z})$.
- Update the generator by ascending its stochastic gradient (improved objective):

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^m \log(D_{\theta_d}(G_{\theta_g}(z^{(i)})))$$

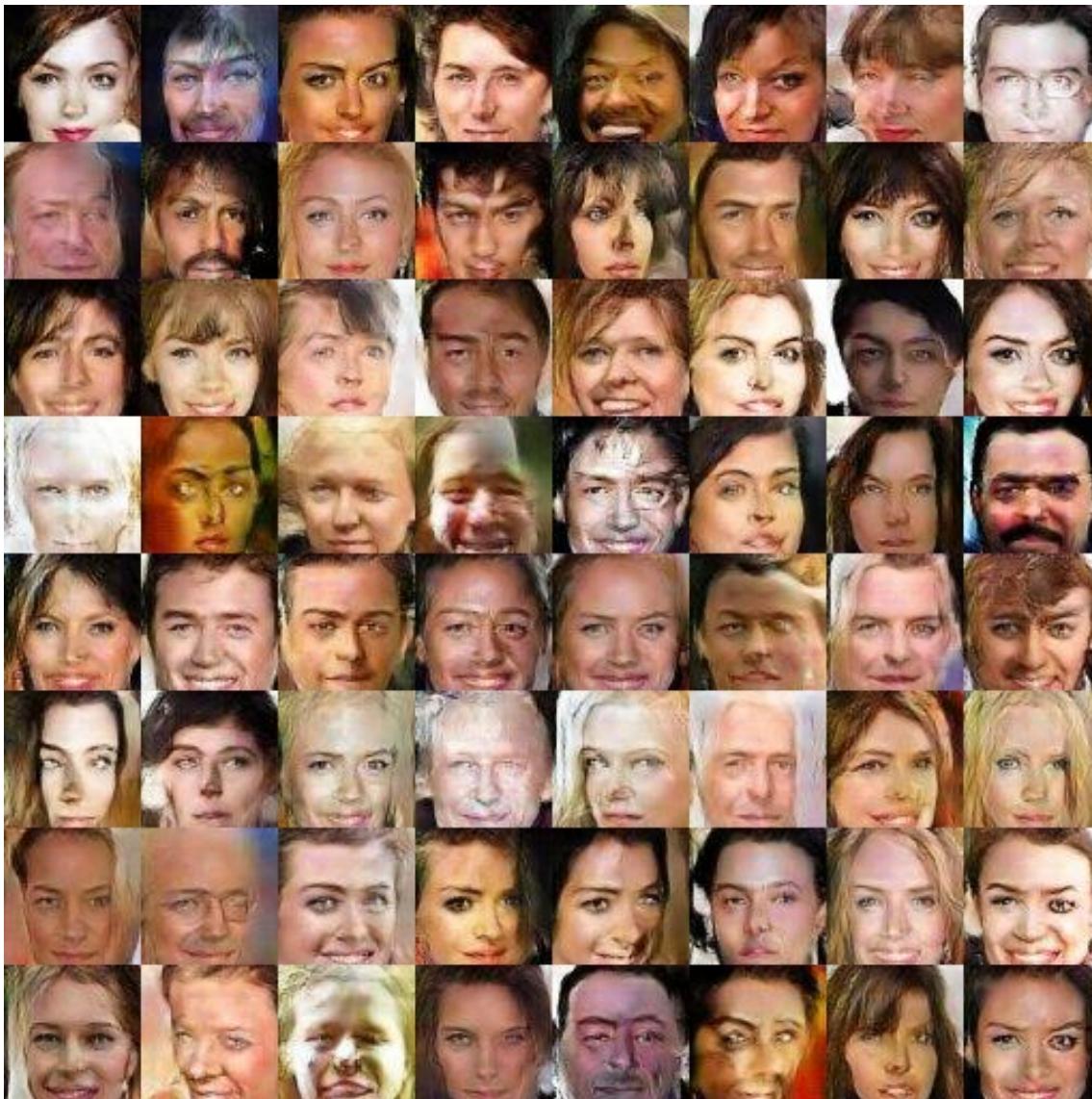
end for

GAN

- پس از آموزش، از شبکه مولد برای تولید تصاویر جدید استفاده می‌شود



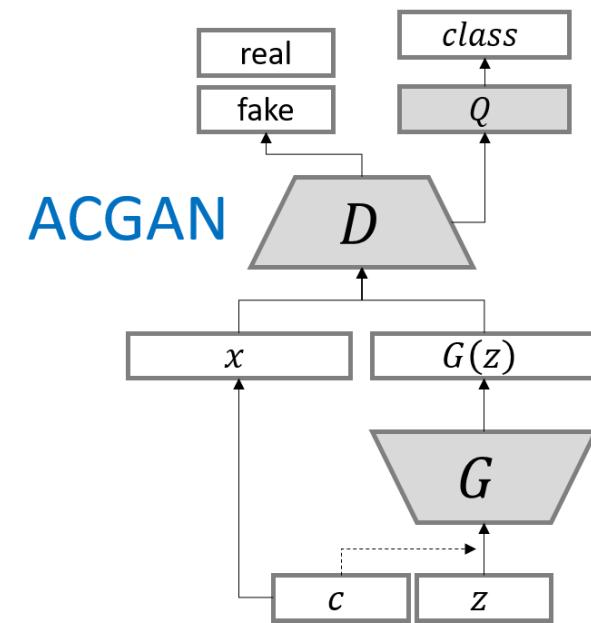
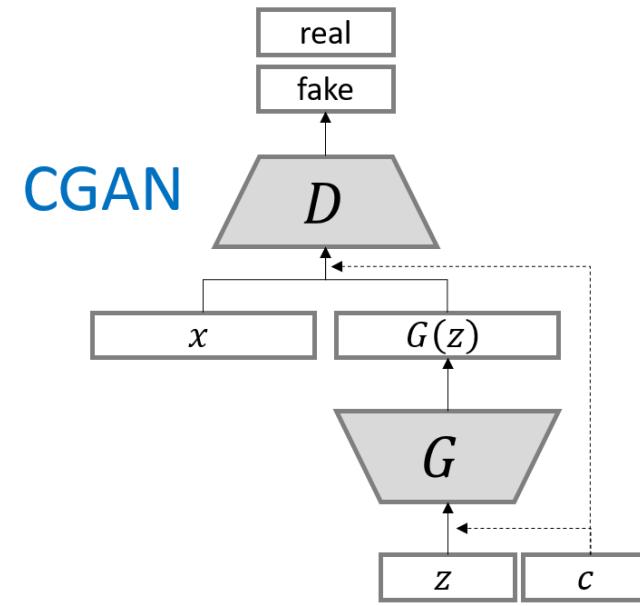
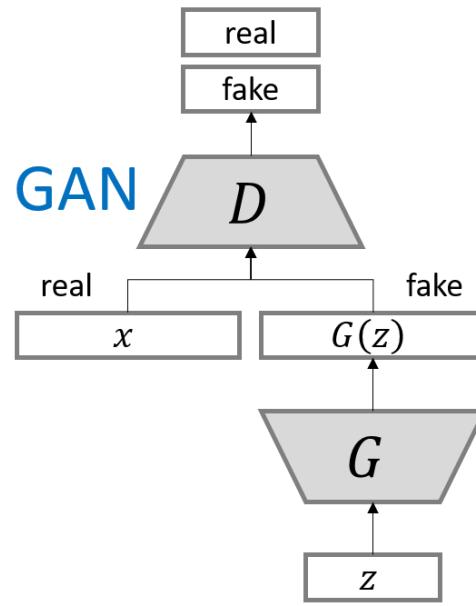
GAN



Karras, T., et. al., "A style-based generator architecture for generative adversarial networks." CVPR 2019.

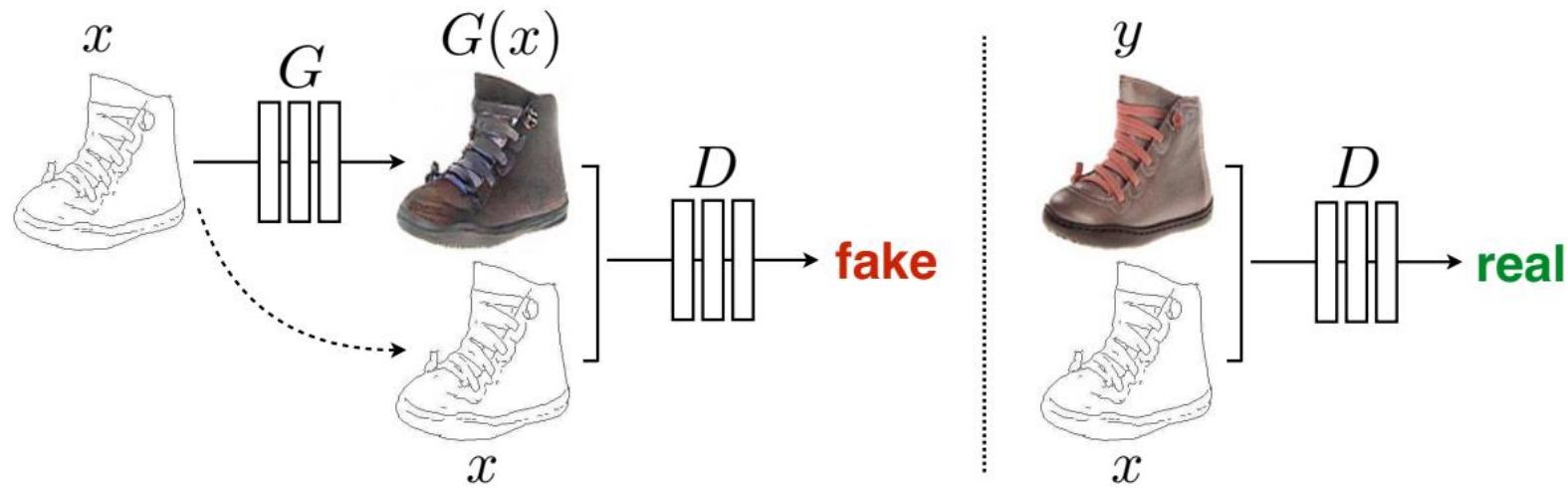
GAN شرطی

- ورودی شبکه مولد علاوه بر بردار تصادفی، می‌تواند یک بردار شرط هم باشد
- برای آنکه مطمئن شویم نمونه‌های ساختگی شرایط لازم را دارند، می‌توان از شبکه ممیز خواست تا علاوه بر واقعی/ساختگی بودن، دسته آن را نیز تخمین بزند



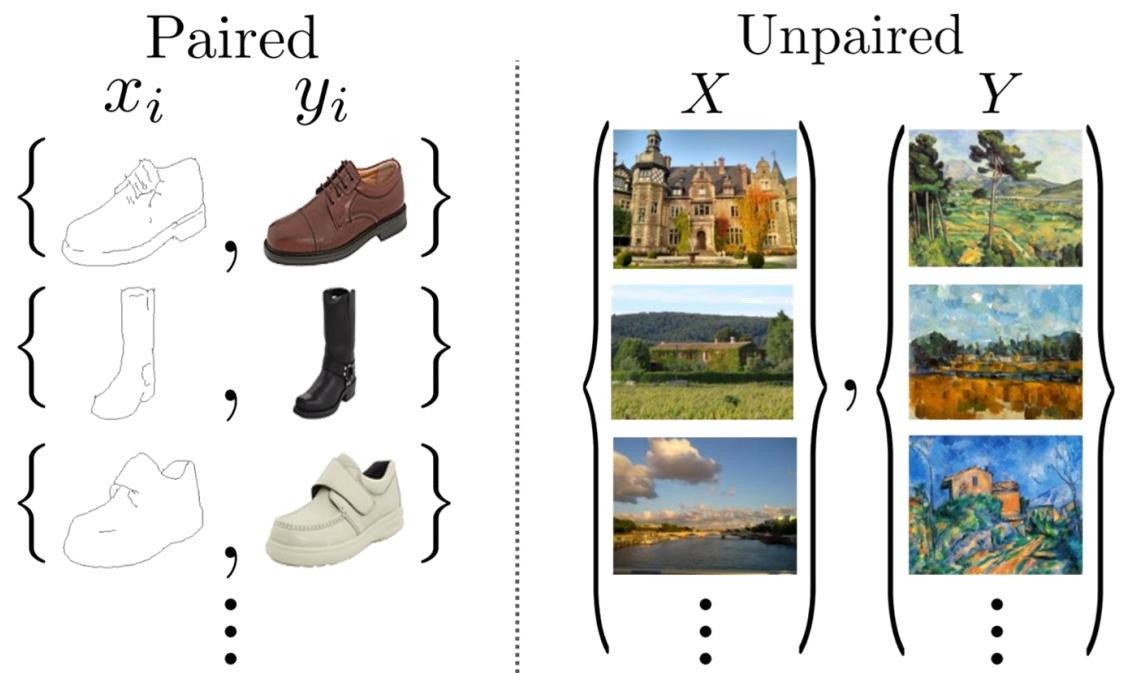
GAN شرطی

- برای آموزش چنین مدل‌هایی نیاز به جفت تصاویری داریم که ورودی و خروجی شبکه مولد را مشخص کنند
- مزیت این روش نسبت به مقایسه پیکسل به پیکسل خروجی شبکه مولد با خروجی مطلوب آن است که مقایسه بر حسب محتوا انجام می‌شود



CycleGAN

- تهیه کردن تصاویر جفت در بسیاری از کاربردها هزینه‌بر است و در برخی موارد غیرممکن است
- رویکرد CycleGAN استفاده از تصاویر غیرجفت برای آموزش مدل است (تعدادی تصویر از ورودی شبکه داریم و تعدادی تصویر از خروجی شبکه که با هم متناظر نیستند)
- چطور می‌شود چنین مدلی را آموزش داد؟



تصاویر فصل زمستان

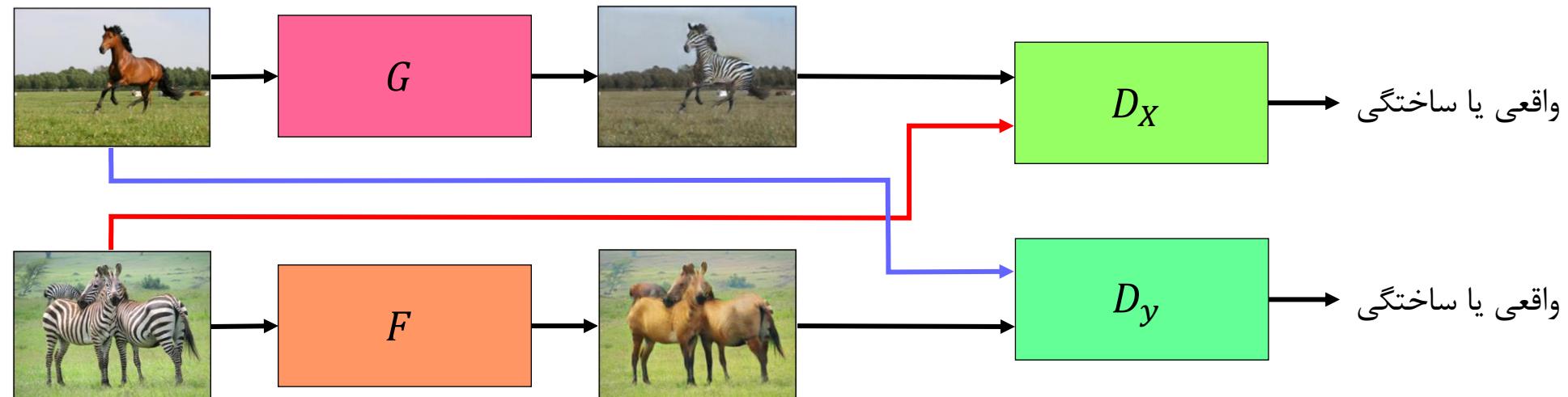


تصاویر فصل تابستان

CycleGAN

- ایده CycleGAN آن است که:

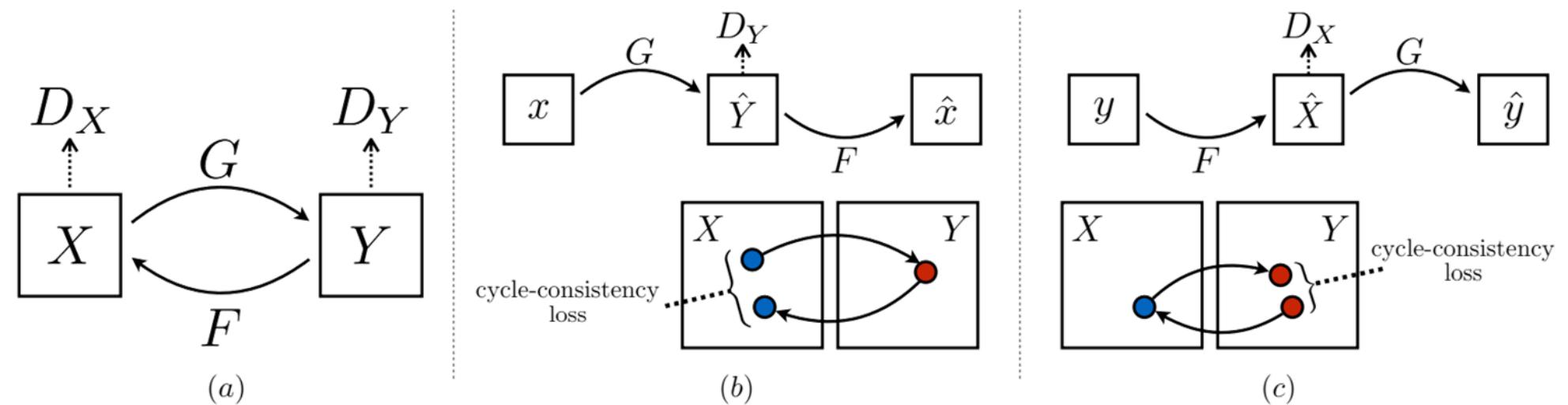
- دو شبکه مولد داشته باشد (شبکه G برای تبدیل از X به Y و شبکه F برای تبدیل از Y به X)
- دو شبکه ممیز داشته باشد (شبکه D_X برای تشخیص ساختگی بودن تصاویر حوزه X و شبکه D_Y برای تشخیص ساختگی بودن تصاویر حوزه Y)
- دو شبکه مولد باید معکوس یکدیگر باشند



CycleGAN

- ایده CycleGAN آن است که:

- دو شبکه مولد داشته باشد (شبکه G برای تبدیل از X به Y و شبکه F برای تبدیل از Y به X)
- دو شبکه ممیز داشته باشد (شبکه D_X برای تشخیص ساختگی بودن تصاویر حوزه X و شبکه D_Y برای تشخیص ساختگی بودن تصاویر حوزه Y)
- دو شبکه مولد باید معکوس یکدیگر باشند

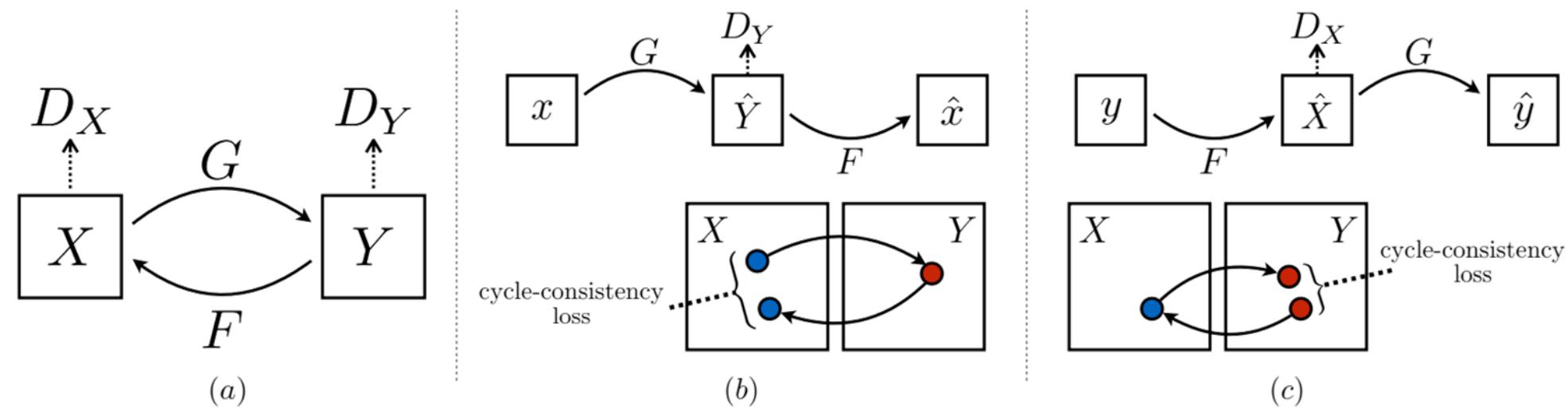


CycleGAN

$$\begin{aligned}\mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) = & \mathbb{E}_{y \sim p_{\text{data}}(y)} [\log D_Y(y)] \\ & + \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log(1 - D_Y(G(x)))]\end{aligned}$$

$$\begin{aligned}\mathcal{L}_{\text{cyc}}(G, F) = & \mathbb{E}_{x \sim p_{\text{data}}(x)} [\|F(G(x)) - x\|_1] \\ & + \mathbb{E}_{y \sim p_{\text{data}}(y)} [\|G(F(y)) - y\|_1]\end{aligned}$$

$$\begin{aligned}\mathcal{L}(G, F, D_X, D_Y) = & \mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) \\ & + \mathcal{L}_{\text{GAN}}(F, D_X, Y, X) \\ & + \lambda \mathcal{L}_{\text{cyc}}(G, F)\end{aligned}$$



خدا با چنان کن سرانجام کار
تو خشنود باشی و ما رستگار