به نام خدا

یادگیری عمیق، تکلیف پنجم
مهدی کافی ۹۹۲۱۰۷۵۳

مسئله‌ی ۱. (۱۰ نمره)

(آ) یکی از راه‌های معمول برای تخمین گرادیان یک امید ریاضی، استفاده از رابطه‌ی زیر است:

$$\nabla_\theta \mathbb{E}_{z\sim q_\theta(z)}\left[f\left(z\right)\right] \approx \frac{1}{N}\sum_{i=1}^{N} f\left(z^i\right).\nabla_\theta \ln q_\theta\left(z^i\right)$$

که در آن هر z^i نمونه‌ی مستقلی از توزیع $q_\theta\left(z\right)$ می باشد. درستی این رابطه را نشان دهید و بیان کنید که چطور می توانیم از آن در VAE استفاده کنیم. با مراجعه به این مقاله مشکلی که در استفاده از این روش وجود دارد را بیان کنید.

(ب) به صورت شهودی بیان کنید که روش Reparameterization چگونه می‌تواند این مشکل را حل کند؟

(ج) در بسیاری از موارد تابع خطای رمزگشای VAE را خطای MSE در نظر میگیریم. این در حالی است که هدف ما بیشینه کردن تابع $\mathbb{E}_{z\sim q_\theta(z|x)} \ln p_\theta\left(x|z\right)$ می باشد. در چه صورتی و با چه فرض هایی این دو کار معادل یکدیگر هستند؟

آ) در مواردی که نمی‌توانیم مقدار گرادیان امید ریاضی را به صورت مستقیم محاسبه کنیم، می‌توانیم این مقدار را تخمین بزنیم. برای تخمین می‌توانیم به صورت زیر عمل کنیم.

$$\begin{aligned}\nabla_\theta E_q[f(z)] &= \nabla_\theta \int_z f(z)q(z)dz \\ &= \int_z f(z)\nabla_\theta q(z)dz \\ &= \int_z f(z)q(z)\nabla_\theta \ln\left(q(z)\right)dz \\ &= E_q[f(z)\nabla_\theta \ln\left(q(z)\right)]\end{aligned}$$

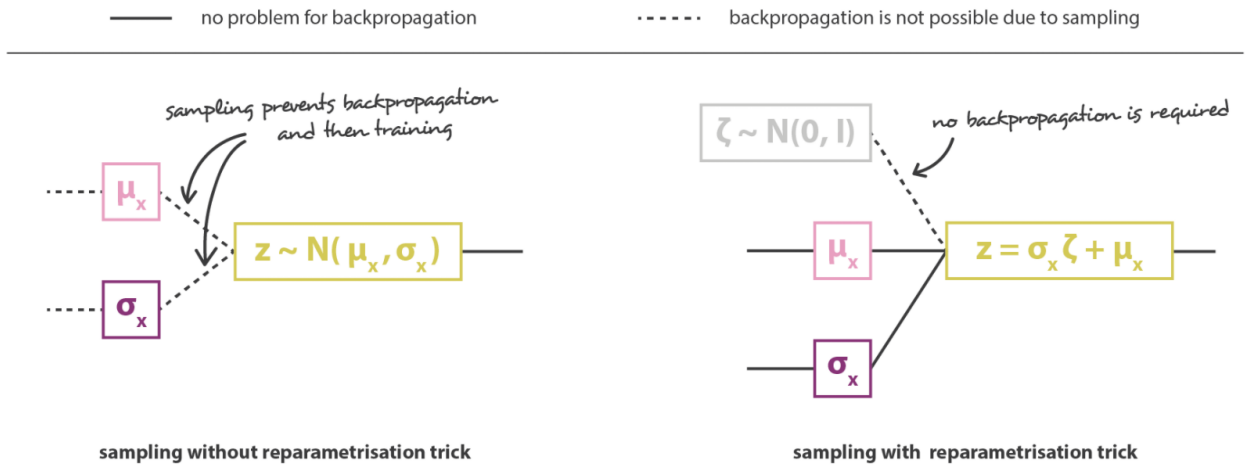
می‌توانیم امید ریاضی به دست آمده در رابطه بالا را با استفاده از روش Monte Carlo به صورت زیر بنویسیم.

$$\nabla_\theta E_q[f(z)] = \frac{1}{N}\sum_{i=1}^{N} f\left(z^i\right).\nabla_\theta \ln\left(q_\theta\left(z^i\right)\right)$$

در VAEمی‌دانیم که در لایه latent تلاش داریم که توزیع شناخته شده‌ای (معمولا توزیع گاوسی نرمال) را ایجاد کنیم و سپس از این توزیع نمونه گیری کنیم و برداری مانند z بسازیم. سپس با داشتن بردار z به عنوان ورودی بخش decoder، ورودی اولیه را بازسازی کنیم و سپس با محاسبه خطا و انتشار آن به سمت ابتدای مدل، شبکه‌ها را آموزش دهیم. می‌دانیم که در بخشی از این فرایند بردار z را از توزیعی نمونه برداری کردیم که امکان محاسبه مشتق برای این مرحله وجود ندارد. بنابراین می‌توانیم با استفاده از روش بالا، مقدار گرادیان امید این توزیع را محاسبه و در شبکه پخش کنیم تا آموزش شبکه امکان پذیر شود. با توجه به مقاله توصیه شده، می‌دانیم که مشکل این روش این است که واریانس گرادیان تخمین زده شده می‌تواند بسیار بزرگ باشد و با داشتن N نمونه از یک توزیع X، مقدار کوواریانس میانگین برابر با مقدار $\bar{Cov(X)} = Cov(X)/N$ خواهد بود. مقادیر روی قطر ماتریس کوواریانس

$Cov(X)$ همان واریانس گرادیان تخمین زده شده هستند که می‌دانیم می‌توانند بسیار بزرگ باشند. حال به منظور اینکه این مقادیر را در $Cov(\bar{X})$ کمتر از مقدار خاصی نگه داریم، به نمونه‌های بسیار زیادی نیاز داریم و الگوریتم کند می‌شود.

ب) می‌دانیم در میانه راه VAE نیاز به نمونه گیری از توزیع $p(z|x)$ داریم که x داده‌های ورودی و $p(z)$ توزیع داده‌های کد شده هستند. در ضمن می‌دانیم که عمل نمونه برداری، مشتق پذیر نیست و نیاز داریم که به طور مثال از روشی مانند روش بالا استفاده کنیم که دیدیم این روش هم مشکلاتی دارد. روش دیگری که می‌توان از آن استفاده کرد، Reparameterization است. این روش به این صورت عمل می‌کند که از یک توزیع گاوسی استاندارد نمونه برداری می‌کند و سپس با ضرب کردن بردار واریانس توزیع میانی در این نمونه و جمع کردن حاصل با میانگین توزیع میانی، نمونه برداری را شبیه سازی می‌کند. حال مسیر محاسبه گرادیان از نمونه برداری مستقل می‌شود و به راحتی می‌توانیم گرادیان خطا نسبت به بردارهای واریانس و میانگین را محاسبه کنیم و شبکه را آموزش دهیم.



ج) با فرض‌های زیر این دو کار معادل یکدیگر خواهندبود.

- فرض می‌کنیم که توزیع داده‌های کد شده، یک توزیع نرمال استاندارد است. $p(z) \sim N(0, I)$
- توزیع likelihood که توزیع داده‌های تولید شده با داشتن داده‌های کد شده است، نیز یک توزیع نرمال است که میانگین آن حاصل تابع f بر داده کد شده و ماتریس کوواریانس آن یک ماتریس قطری حاصل از ضرب ثابت مثبت c در ماتریس همانی است. $p(x|z) \sim N(f(z), cI)$
- همچنین برای تخمین توزیع داده‌های کد شده با داشتن داده‌های واقعی $p(z|x)$ از توزیع q_x استفاده می‌کنیم و فرض می‌کنیم که این توزیع نیز یک توزیع گاوسی به صورت روبرو است. $q_x(z) = N(g(x), h(x))$

با فرض‌های گفته شده، هدف از VAE این است که encoder و decoderای بسازیم و تابع fای را انتخاب کنیم که مقدار امید log-likelihood توزیع شرطی x با داشتن zای که از توزیع q(x) اصلی آمده باشد را بیشینه کند. به عبارت دیگر با داشتن یک x می‌خواهیم که احتمال برابری داده تولید شده $\hat{x}$ با x را بیشینه کنیم در حالیکه مقدار z را از توزیع $q^*_x(z)$ نمونه برداری شده باشد و سپس داده $\hat{x}$ از توزیع $p(x|z)$ نمونه برداری شده باشد. در اینصورت دو عبارت گفته شده با یکدیگر برابر هستند.

$$f^* = \arg\max_{f \in F} \mathbb{E}_{z \sim q_x^*}(\log p(x|z))$$

$$= \arg\max_{f \in F} \mathbb{E}_{z \sim q_x^*}\left(-\frac{||x - f(z)||^2}{2c}\right)$$

مسئله‌ی ۲. (۱۵ نمره)

(آ) در یک VAE اگر داده ورودی از نوع باینری باشد (تصویری با پیکسل های ۰ و ۱)، می‌توان به جای توزیع گاوسی چندمتغیره روی خروجی کدگشا، از توزیع برنولی چندمتغیره استفاده کرد. توزیع خروجی کدگشا را به شکل برنولی چندمتغیره در نظر بگیرید و تابع فعال سازی آخرین لایه کدگشا را sigmoid در نظر بگیرید. اثبات کنید که در تابع هزینه این شبکه یک جمله ای به شکل Binary Cross Entropy ظاهر می شود.

(ب) تکنیک Reparameterization روی بسیاری از توزیع‌های پیوسته قابل اعمال است. تحقیق کنید که چگونه می‌توان از این تکنیک برای یک توزیع categorical استفاده کرد؟

(ج) یکی از نسخه‌های تغییریافته VAE مقاله مربوط به beta-VAE می باشد. در این روش یک ضریب beta ای پشت یکی از توابع هزینه قرار میگیرد. درباره این روش تحقیق کنید و بیان کنید:

۱) با انجام چه روندی از محاسبات، این ضریب در تابع هدف این روش ظاهر می‌شود؟

۲) هذف از افزودن ضریب beta چیست و اضافه کردن آن چه تاثیری روی ویژگی‌های فضای نهان یادگرفته‌شده توسط مدل دارد؟

آ)

مسئله‌ی ۳. (۱۰ نمره)

(آ) همانطور که می‌دانیم، دو مدل GAN و VAE از مهم ترین و شناخته شده ترین مدل‌های generative در یادگیری عمیق می‌باشند. یکی از مهم‌ترین کاربردهای آن‌ها، تولید تصاویر و data augmentation می‌باشد. در این صورت، با فرض استفاده از مجموعه داده یکسان و فرآیند آموزش نسبتا کامل، آیا به طور کلی کیفیت تصاویر تولید شده توسط یکی از این مدل‌ها بر دیگری برتری دارد؟ لطفا پاسخ خود را با دلیل و در صورت نیاز اثبات ریاضی تشریح نمایید. می‌توانید از این مقاله استفاده نمایید.

(ب) تابع ReLU یکی از پرکاربردترین توابع فعالسازی مورد استفاده در شبکه های یادگیری عمیق می‌باشد، اما در برخی کاربری‌های خاص همانند بخش Generative در روش GAN می‌تواند منجر به ایجاد مشکل در فرآیند آموزش شود، به عبارت دیگر شبکه مولد ما آموزش نخواهد دید. در این مورد استثنا، توصیه می‌شود به جای ReLU ، از Leaky ReLU به عنوان تابع فعالسازی استفاده گردد. لطفا علت بروز مشکل به هنگام استفاده از ReLU را به طور کامل توضیح داده و تشریح کنید که به چه علت استفاده از Leaky ReLU می‌تواند مشکل را حل کند.

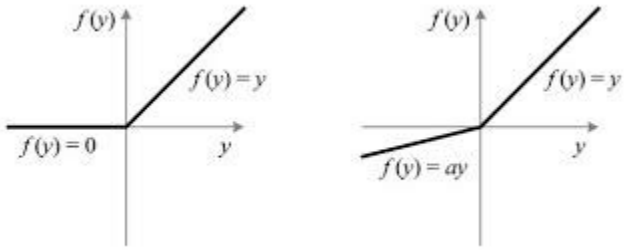
آ) به طور معمول تصاویری که توسط GANها تولید می‌شوند، تصاویری با لبه‌های تیزتری نسبت به تصاویر تولیدی VAEها هستند. تصاویر تولید شده توسط VAEها، تصاویری مات هستند. می‌دانیم که VAEها برای تولید داده‌های جدید نیاز دارند که در لایه میانی یک توزیع (معمولا گاوسی استاندارد) را تولید نمایند و سپس از این توزیع با روشی مثل reparameterization نمونه برداری کنند و داده‌های جدید تولید کنند. نیاز به رساندن توزیع لایه میانی به توزیعی prior یکی از عواملی است که باعث می‌شود تصاویر تولید VAEها کمی مات شوند. این موضوع در تابع هزینه آنها نیز مشهود است. زیرا که تابع هزینه آنها از دو بخش reconstruction loss و KL divergence تشکیل شده است که بخش اول سعی می‌کند که تصاویر تولیدی را نزدیک به

تصاویر اولیه کند و تصاویر شفاف تولید کند و بخش دوم سعی می‌کند که توزیع میانی را به توزیع prior نزدیک کند و تصاویر تولیدی مات می‌شوند و بین این دو بخش یک trade-off برقرار است.

$$\left(\mathbb{E}_{z\sim q_x}\left(-\frac{||x-f(z)||^2}{2c}\right)-KL(q_x(z),p(z))\right)$$

همچنین دلیل دیگر مات شدن تصاویر خروجی VAEها این است که معمولا به منظور سادگی ماتریس کوواریانس لایه میانی را یک ماتریس قطری در نظر می‌گیرند (در واقع تمام ابعاد مستقل از یکدیگر در نظر گرفته می‌شوند) و این موضوع نیز باعث می‌شود که تصاویر تولیدی دقیق نباشند. GANها این مشکلات را ندارند و آزادانه در فضای فرضیه تلاش برای ساختن تصاویر با کیفیت می‌کنند و در نهایت تصاویری شفاف‌تر تولید می‌کنند.

ب) در بخش Discriminator شبکه‌های GAN باید از LeakyReLU استفاده کنیم زیرا که در صورت استفاده از تابع فعال سازی ReLU، ممکن است که خروجی تمامی نورون‌های Discriminator در یکی از لایه‌های آن صفر شود و در این صورت حالتی ایجاد می‌شود که به آن dying state گفته می‌شود. در این حالت مقدار گرادیان برابر با صفر می‌شود و Generator که برای آموزش نیاز به مقادیر گرادیان دارد، فقط مقدار صفر می‌بیند و آموزشی صورت نمی‌گیرد زیرا که شبکه مولد برای آموزش به گرادیان حاصل از شبکه تمیز دهنده نیاز دارد و در حالت dying state آموزش امکان پذیر نیست. تابع فعال سازی LeakyReLU از این اتفاق جلوگیری می‌کند به این صورت که در صورتیکه ورودی آن مثبت باشد همان مقدار را تولید می‌کند و در صورتیکه ورودی منفی باشد به جای صفر، مقداری مانند α را در ورودی ضرب کرده و حاصل ضرب را در خروجی تولید می‌کند با این کار باعث می‌شود که مقدار گرادیان‌ها برای شبکه مولد صفر نشوند و این شبکه بتواند آموزش ببیند.



مسئله‌ی ۴. (۱۵ نمره)

(آ) یکی از مشکلات شایع در شبکه های GAN مشکل Mode Collapse می باشد که باعث می شود شبکه GAN به یک حالت عدم آموزش رسیده و به طور متوالی خروجی های یکسان تولید کند. این مشکل را به طور کامل تشریح کرده و راه حل های احتمالی به منجر به غلبه بر این مشکل می شوند را بیان نمایید.

(ب) معماری کلی و تابع هزینه مدل W-GAN را تشریح نمایید و تفاوت های آن با مدل پایه GAN را توضیح دهید. آیا تابع هزینه این مدل کمکی به برطرف شدن مشکل Mode Collapse خواهد کرد؟ توضیح دهید.

آ) در شبکه‌های GAN انتظار داریم که با دادن ورودی‌های متفاوت خروجی‌های متفاوتی از شبکه generator دریافت کنیم. اما مشکلی شایع در شبکه‌های GAN وجود دارد که باعث می‌شود شبکه generator حتی با گرفتن ورودی‌های متفاوت، همواره خروجی‌هایی یکسان یا بسیار شبیه و با کیفیت تولید کند به نحوی که شبکه discriminator را فریب دهد. بنابراین شبکه generator همواره خروجی یکسان و با کیفیت تولید می‌کند و شبکه discriminator را فریب می‌دهد اما پس از مدتی شبکه discriminator متوجه می‌شود که خروجی‌های یکسان تولید شده توسط generator همگی تقلبی هستند و تمام ورودی‌های از آن نوع را تقلبی شناسایی می‌کند حتی اگر یک نمونه از آن نوع از داده‌های واقعی آمده باشد. در این حالت generator وارد mode دیگری می‌شود و شروع به تولید داده‌هایی دیگر، شبیه و با کیفیت می‌کند و اتفاقات گفته شده تکرار می‌شوند. می‌دانیم که تابع هزینه مدل generator به صورت زیر است.

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} log(1 - D(G(z^{(i)})))$$

در حالیکه داده‌های تولیدی به x^* همگرا شوند به نحوی که به راحتی discriminator را فریب دهند، این داده‌ها از ورودی z مستقل می‌شوند.

$$x^* = argmax_x D(x)$$

بنابراین گرادیان نسبت به z برابر با 0 می‌شود.

$$\frac{\partial J}{\partial z} \approx 0$$

بنابراین شبکه مولد آموزش نمی‌بیند. هنگام آموزش شبکه تمیز دهنده، این شبکه داده‌های یکسان تولیدی از شبکه مولد را شناسایی می‌کند و شبکه مولد را به سمت یک mode دیگر هدایت می‌کند.

برخی از راه حل‌های پیشنهادی برای حل مشکل mode collapse در زیر آورده شده است.

- مدل AdaGAN؛ این مدل چندین GAN را به روش boosting آموزش می‌دهد.
- مدل VEEGAN؛ شبکه‌ای در این مدل وجود دارد که برعکس مولد عمل می‌کند و داده‌های تولیدی را به نویز اولیه تبدیل می‌کند.
- مدل Wasserstein GAN
- مدل Unrolled GAN

ب) فاصله Wasserstein یک معیار برای محاسبه فاصله بین توزیع است که از دو معیار JS و KL معیاری نرم‌تر و بهتر است. به طوریکه برای دو توزیع P و Q فاصله بسیار بهتری را نسبت به روش‌های JS و KL محاسبه می‌کند.

$$\forall (x,y) \in P, x = 0 \text{ and } y \sim U(0,1)$$
$$\forall (x,y) \in Q, x = \theta, 0 \leq \theta \leq 1 \text{ and } y \sim U(0,1)$$

در صورتیکه مقدار θ مخالف صفر باشد، معیارهای گفته شده فاصله دو توزیع را به صورت زیر محاسبه می‌کنند.

$$D_{KL}(P\|Q) = \sum_{x=0, y\sim U(0,1)} 1 \cdot \log \frac{1}{0} = +\infty$$
$$D_{KL}(Q\|P) = \sum_{x=\theta, y\sim U(0,1)} 1 \cdot \log \frac{1}{0} = +\infty$$
$$D_{JS}(P,Q) = \frac{1}{2}\left(\sum_{x=0, y\sim U(0,1)} 1 \cdot \log \frac{1}{1/2} + \sum_{x=0, y\sim U(0,1)} 1 \cdot \log \frac{1}{1/2}\right) = \log 2$$
$$W(P,Q) = |\theta|$$

و در حالیکه مقدار θ برابر با صفر باشد، نیز فاصله‌ها به صورت زیر محاسبه می‌شوند.

$$D_{KL}(P\|Q) = D_{KL}(Q\|P) = D_{JS}(P,Q) = 0$$
$$W(P,Q) = 0 = |\theta|$$

دیده می‌شود که معیار Wasserstein نرم‌تر و دقیقتر از دو روش دیگر، فاصله را محاسبه می‌کند. ایده WGANها استفاده از این فاصله است. این فاصله به صورت زیر تعریف می‌شود.

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

نکته‌ای که لازم است به آن توجه کنیم این است که عبارت بالا قابل محاسبه نیست زیرا که نمی‌توانیم مقدار infimum را برای تمام γها محاسبه کنیم ولی می‌توانیم با استفاده از دوگان Kantorovich-Rubinstein عبارت را به صورت زیر بازنویسی کنیم.

$$W(\mathbb{P}_r, \mathbb{P}_g) = \sup_{||f||_L \leq 1} \mathbb{E}_{x\sim\mathbb{P}_r}[f(x)] - \mathbb{E}_{x\sim\mathbb{P}_g}[f(x)]$$

اگر خانواده‌ای از توابع f_w را در نظر بگیریم که پیوسته K-Lipschitz هستند، خواهیم داشت.

$$W(\mathbb{P}_r, \mathbb{P}_g) \propto \max_{w \in W} \mathbb{E}_{x \sim \mathbb{P}_r}[f_w(x)] - \mathbb{E}_{z \sim Z}[f_w(g_\theta(z))]$$

در شبکه‌های WGAN به جای شبکه تمیزدهند از یک منقتد (critic) استفاده می‌شود، که فاصله wasserstein را تخمین می‌زند و تابع هزینه آن به صورت زیر است.

$$L_{critic}(w) = \max_{w \in W} \mathbb{E}_{x \sim \mathbb{P}_r}[f_w(x)] - \mathbb{E}_{z \sim Z}[f_w(g_\theta(z))]$$

تفاوت تمیزدهند و منتقد در اینجاست که تمیزدهنده سعی می‌کرد داده‌های اصلی و تقلبی را از یکدیگر تشخیص دهد در حالیکه منتقد فاصله توزیع‌های تقلبی و واقعی را محاسبه می‌کند.

مولد در WGAN نیز سعی می‌کند که فاصله بین این دو توزیع را کاهش دهد و تابع هزینه آن به صورت زیر است.

$$L_{gen}(w) = \min_{\theta} \mathbb{E}_{x \sim \mathbb{P}_r}[f_w(x)] - \mathbb{E}_{z \sim Z}[f_w(g_\theta(z))] = \min_{\theta} -\mathbb{E}_{z \sim Z}[f_w(g_\theta(z))]$$

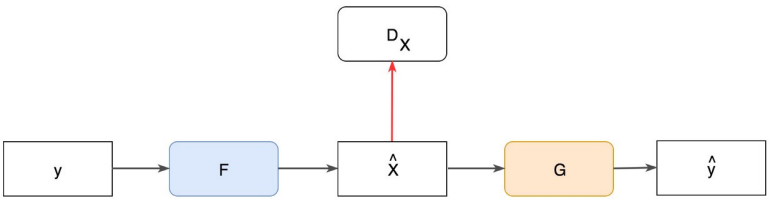
می‌دانیم که WGANها در برطرف کردن مشکل mode collapse بسیار موفق هستند.

مسئله‌ی ۵. (۵ + ۱۰ نمره)

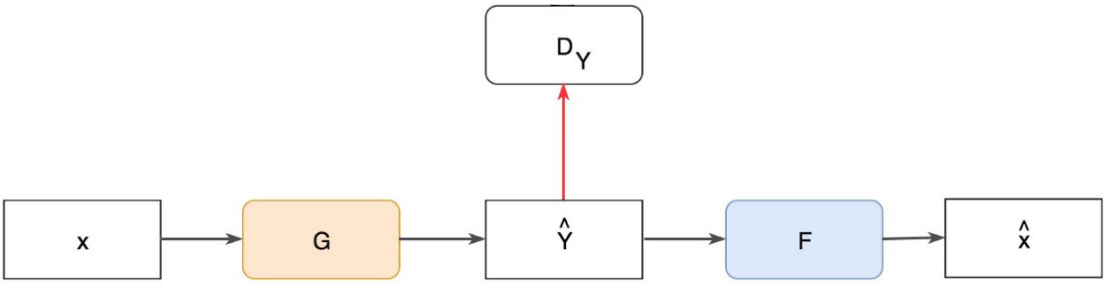
(آ) یکی از چالش های مهم که مدل های GAN تاثیر به سزایی در برطرف شدن آن ها دارند موضوع image2image translation می باشد. یکی از مدل هایی که به طور خاص برای این امر توسعه داده شده است مدل Cycle GAN می باشد. معماری دوگانه این مدل را تشریح کرده و مزایای آن را در ارتباط با چالش مذکور نسبت به مدل پایه GAN بیان نمایید.

(ب) فرض کنید تحت شرایطی بسیار خاص از شما درخواست شده با تعداد داده محدود (در حدی که معماری پایه GAN به سختی به کمک آن train می شود) یک مدل Cycle GAN را train نمایید. در صورتی که الزام به استفاده از Cycle GAN باشد، راه حل پیشنهادی شما چیست؟ (دقت شود سوال ابتکاری بوده و پاسخ کاملا یکتا ندارد، اما بایستی هر راه حل پیشنهادی با تحلیل کامل و حتی استفاده از روابط ریاضی در صورت نیاز تشریح گردد).

آ) شبکه‌های Cycle GAN به این منظور استفاده می‌شوند که به طور مثال تصویری از یک اسب و پس زمینه آن داریم و تمایل داریم که تنها اسب در تصویر با گورخر جایگزین شود بدون اینکه پس زمینه تغییر کند. داده‌های در اختیارمان نیز فقط تصاویری از اسب و تصاویری از گورخر است که لزوما ارتباطی با یکدیگر ندارند. در این شرایط ما می‌توانیم شبکه GANای تولید کنیم که با گرفتن یک تصویر اسب، یک تصویر از گورخر تولید کند و همچنین می‌توانیم شبکه GANای تولید کنیم که با دریافت تصویر گورخر، تصویر اسب تولید کند و تصاویر خروجی این شبکه‌ها لزوما ارتباطی با تصاویر ورودی ندارند.بخش‌های مولد این شبکه‌ها را شبکه‌های G و F می‌نامیم. ایده‌ای که برای رسیدن به هدف اولیه مطرح می‌شود این است که تصویر خروجی از شبکه مولد G (مطلوب است که تصویر گورخر باشد) را به عنوان ورودی به شبکه مولد F (شبکه مولدی که با گرفتن تصویر گورخر، تلاش به تولید تصویر اسب می‌کند) بدهیم و خروجی این شبکه را که تمایل داریم اسب باشد را با ورودی اولیه که تصویر اسب با پس زمینه بود، مقایسه کنیم برای این مقایسه می‌توانیم به طور مثال از تابع هزینه MSE استفاده کنیم. در همین حین تلاش می‌کنیم که با استفاده از بخش تمیز دهنده هر کدام از شبکه‌های GAN گفته شده، کیفیت تصاویر تولیدی بخش مولد هر کدام را بهبود ببخشیم. بنابراین اگر y بیانگر تصویر اولیه اسب با پس زمینه باشد، $\hat{x}$ تصویر تولیدی گورخر باشد و $\hat{y}$ تصویر اسب تولید شده از روی $\hat{x}$ باشد؛ تصویر زیر بیانگر توضیحات بالا است.



همچنین مراحل گفته شده را برای شبکه دیگر نیز تکرار می‌کنیم به طوریکه با تصویر گورخر به عنوان ورودی شروع می‌کنیم، تصویر اسب تولید می‌کنیم و دوباره تلاش می‌کنیم که تصویر گورخر ابتدایی با پس زمینه را بازسازی کنیم.



برای تابع هزینه Cycle GAN نیاز داریم تا تمام موارد گفته شده را در نظر داشته باشیم. در ابتدا تابع هزینه برای نزدیک کردن تصاویر اولیه با پس‌زمینه و تصاویر خروجی با پس‌زمینه را به صورت زیر تعریف می‌کنیم.

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

می‌بینیم که در این تابع، فاصله بین تصویر اولیه اسب با پس‌زمینه یعنی y با تصویر تولید شده از روی گورخر تولید شده از روی تصویر اسب اولیه یعنی $G(F(y))$ محاسبه شده است. همچنین برای تصاویر گورخر نیز کار مشابهی را انجام داده‌ایم و سعی در کم کردن این فاصله داریم.
سپس برای هر کدام از شبکه‌های GAN نیاز است که تابع هزینه متداول آنها را محاسبه کنیم که این تابع هزینه برای بخش مولد و تمیزدهنده در زیر آمده است.

$$\mathcal{L}_{\text{GAN}}(G, D, X, Y)$$
$$\textit{For } G, \text{ minimize } \mathbb{E}_{x\sim p_{\text{data}}(x)}[(D(G(x)) - 1)^2]$$
$$\textit{For } D, \text{ minimize } \mathbb{E}_{y\sim p_{\text{data}}(y)}[(D(y) - 1)^2] + \mathbb{E}_{x\sim p_{\text{data}}(x)}[D(G(x))^2]$$

سپس تابع هزینه کل معماری CycleGAN ترکیبی از تمام توابع هزینه گفته شده است.

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

مدل پایه GAN توانایی انجام نیاز مطرح شده را ندارد و نمی‌تواند به طور مثال در یک تصویر، فقط اسب را با گورخر جابجا کند و پس‌زمینه را بدون تغییر بازسازی کند. مدل پایه همانطور که بیان شد، می‌تواند با گرفتن تصویر ورودی اسب در یک پس‌زمینه، یک تصویر گورخر که ربطی به اسب و پس‌زمینه آن ندارد تولید کند.

ب)