مسئله اول –

قسمت ۱) میدانیم که KL Divergence دو توزیع، اگر توزیع اول را p و توزیع دوم را q بنامیم، برابر است با:

$$D_{KL}(p||q) = \int p(x)\log\left(\frac{p(x)}{q(x)}\right)dx = E_p[\log(p) - \log(q)]$$

توزیع های گاوسی چند متغیره گفته شده سوال، به شکل زیر هستند:

$$p(x) = \frac{1}{(2\pi)^{\frac{k}{2}}\sqrt{|\Sigma_1|}} exp\left(-\frac{1}{2}(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1)\right)$$

$$q(x) = \frac{1}{(2\pi)^{k/2}\sqrt{|\Sigma_2|}} exp\left(-\frac{1}{2}(x-\mu_2)^T\Sigma_2^{-1}(x-\mu_2)\right)$$

لذا داریم:

$$D_{KL}(p||q) = E_p[\log(p) - \log(q)]$$

$$= E_p\left[\frac{1}{2}\log\frac{|\Sigma_2|}{|\Sigma_1|} - \frac{1}{2}(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1) + \frac{1}{2}(x-\mu_2)^T\Sigma_2^{-1}(x-\mu_2)\right]$$

$$= \frac{1}{2}E_p\left[\log\frac{|\Sigma_2|}{|\Sigma_1|}\right] - \frac{1}{2}E_p[(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1)] + \frac{1}{2}E_p[(x-\mu_2)^T\Sigma_2^{-1}(x-\mu_2)]$$

$$= \frac{1}{2}\log\frac{|\Sigma_2|}{|\Sigma_1|} - \frac{1}{2}E_p[(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1)] + \frac{1}{2}E_p[(x-\mu_2)^T\Sigma_2^{-1}(x-\mu_2)]$$

حال میدانیم:

$$(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1) = tr\{(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1)\} = tr\{(x-\mu_1)^T(x-\mu_1)\Sigma_1^{-1}\}$$

پس:

$$\frac{1}{2}E_p[(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1)] = \frac{1}{2}E_p[tr\{(x-\mu_1)^T(x-\mu_1)\Sigma_1^{-1}\}]$$

$$= \frac{1}{2} tr\left\{E_p[(x-\mu_1)^T(x-\mu_1)\Sigma_1^{-1}]\right\} = \frac{1}{2} tr\left\{E_p[(x-\mu_1)^T(x-\mu_1)]\Sigma_1^{-1}\right\}$$

$$= \frac{1}{2} tr\{\Sigma_1\Sigma_1^{-1}\} = \frac{1}{2} tr\{I_k\} = \frac{k}{2}$$

هم چنین میدانیم:

$$E_p[(x-\mu_2)^T\Sigma_2^{-1}(x-\mu_2)] = (\mu_1-\mu_2)^T\Sigma_2^{-1}(\mu_1-\mu_2) + tr\{\Sigma_2^{-1}\Sigma_1\}$$

لذا با جمع تمام این نتایج داریم:

$$D_{KL}(p||q) = \frac{1}{2}\log\frac{|\Sigma_2|}{|\Sigma_1|} - \frac{1}{2}E_p[(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1)] + \frac{1}{2}E_p[(x-\mu_2)^T\Sigma_2^{-1}(x-\mu_2)]$$

$$= \frac{1}{2}\left[\log\frac{|\Sigma_2|}{|\Sigma_1|} - k + (\mu_1-\mu_2)^T\Sigma_2^{-1}(\mu_1-\mu_2) + tr\{\Sigma_2^{-1}\Sigma_1\}\right]$$

قسمت ۲) داریم:

$$D_{KL}(p||q) = \int p(x)\left(\frac{\log p(x)}{\log q(x)}\right) = \int p(x)(\log p(x) - \log q(x))\ dx$$

$$= \int p(x)\log p(x)\,dx - \int p(x)\log q(x)\ dx$$

$$\xRightarrow{q(x)=\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}}$$

$$D_{KL}(p||q) = \int p(x)\log p(x)\,dx + \int p(x)\left[\frac{1}{2}\log 2\pi + \log\sigma + \frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right],$$

$$\frac{\partial}{\partial\mu}D_{KL}(p||q) = \int p(x)\left[\frac{\mu-x}{\sigma}\right] \xrightarrow{\sigma=1\ because\ cov\ matrix\ is\ I} \mu^* = \frac{\int xp(x)\,dx}{\int p(x)dx} = E_p[x]$$

مسئله دوم –

قسمت ۱) *autoencoder* یک شبکه عصبی *unsupervised* است که یاد میگیرد چگونه به صورت بهینه داده ورودی را فشرده سازی کند و سپس آن را دوباره بازسازی کند.مزیت های این شبکه، اولا این است که برای کاهش بعد و فشرده سازی مفید است ثانیا برای گرفتن نویز از ورودی به کار میرود.از طرف دیگر، میتوانیم از آن برای استخراج ویژگی های مهم داده ورودی استفاده کنیم و به این طریق به نوعی برای *feature engineering* نیز به کار میرود.یکی دیگر از مزیت های آن، این است که ما به وسیله آن میتوانیم یک *non-linear transformation* از داده ورودی را یاد بگیریم و نیازی هم به لایه های *dense* ندارد.بهتر است برای کاهش بعد از آن استفاده کنیم نه از *PCA* که یک *transformation* بزرگ و سنگین است.از طرفی میتوان از لایه های از پیش آموزش داده شده برای *transfer learning* استفاده کند.از این شبکه عصبی، در رنگ آمیزی تصاویر یا حذف *watermark* از تصویر نیز استفاده میشود.

قسمت ۲) در ابتدا یادآوری میکنیم که PCA چگونه کار میکند.اگر x همان داده ورودی و z کاهش یافته شده ی داده ورودی باشد، از z برای بازسازی ورودی استفاده میشود(x hat) لذا:

$$z = B^T x \,, \hat{x} = Bz$$

حال در یک *linear autoencoder* ما از *x* به *z* میرسیم و از *z* به *x bar* . چون حرف از خطی بودن زده شده، یعنی تابع فعالساز مان یک تابع خطی است. میتوانیم *linear autoencoder* را به صورت زیر بنویسیم:

$$\hat{x} = W_2 W_1 x$$

وزن های W_1 و W_2 به ترتیب وزن های لایه اول و دوم است.حال اگر فرض کنیم:

$$W_1 = B \,, W_2 = B^T$$

حال برای *linear autoencoder* داریم:

$$z = W_1 x = Bx \,, \hat{x} = W_2 W_1 x = W_2(W_1 x) = W_2 z = B^T z$$

لذا میبینیم که دقیقا کاری که در *PCA* انجام دادیم نیز در *linear autoencoder* دارد انجام میشود.

مسئله سوم – طبق فرض سوال داریم:

$$E_{z_i \sim q(.|x)}\left[\hat{L}(x)\right] = E_{z_i \sim q(.|x)}\left[\log\left(\frac{1}{M}\sum_{i=1}^{M}\frac{p_\theta(x|z_i)p(z_i)}{q(z_i|x)}\right)\right]$$

$$\xrightarrow{Jensen\ inequality} E_{z_i\sim q(.|x)}\left[\log\frac{1}{M}\sum_{i=1}^{M}\frac{p_\theta(x|z_i)p(z_i)}{q(z_i|x)}\right]$$
$$\leq \log\left(\frac{1}{M}E_{z_i\sim q(.|x)}\left[\sum_{i=1}^{M}\frac{p_\theta(x|z_i)p(z_i)}{q(z_i|x)}\right]\right)$$
$$=\log\left(\frac{1}{M}\left[\sum_{i=1}^{M}\left(\int p_\theta(x|z_i)dz_i\right)\right]\right)=\log p_\theta(x)$$

لذا ثابت شد که *L bar* یک *biased estimator* از $\log p_\theta(x)$ است زیرا همیشه کوچک تر مساوی آن است.

حال اگر *M* به سمت بی نهایت برود، طبق قانون اعداد بزرگ داریم:

$$\lim_{M\to\infty}\left[\log\frac{1}{M}\sum_{i=1}^{M}\frac{p_\theta(x|z_i)p(z_i)}{q(z_i|x)}\right]=\log\left[E_{z_i\sim q(.|x)}\left[\frac{p_\theta(x|z_i)p(z_i)}{q(z_i|x)}\right]\right]=\log p_\theta(x)$$

لذا در این حالت، یک *unbiased estimator* داریم.

مسئله چهارم –

قسمت ۱) وقتی متغیر های پنهان گسسته(مثلا کتگوریکال) باشند، در هنگام *back propagation* بخاطر مشتق ناپذیری آن ها، مشکل پیش می آید.لذا باید این متغیر های کتگوریکال را به شکل توزیع پیوسته در بیاوریم تا بتوانیم از آن ها مشتق بگیریم.

راه پیشنهادی روش *gubmel-softmax* است که یک توزیع گامبل ایجاد میشود(بوسیله نمونه گیری از توزیع یکنواخت و *transform* کردن آن ها به توزیع گامبل) و آن ها با *logit* های شبکه ترکیب میشوند، به این صورت متغیر ها مشتق پذیر میشوند. از طرفی باید تابع *argmax* را هم تخمین بزنیم که ایده آن هم استفاده از *softmax* با رویکرد *simulated annealing* است. یعنی ابتدا با مقدار بزرگ *hyperparameter* آن که لامبدا نام دارد شروع میکنیم و به مرور آن را کوچک میکنیم تا همگرا شود. با استفاده از این روش ها میتوانیم متغیر های پنهان گسسته هم داشته باشیم.

منبع : E. Jang *et al.*, “Categorical reparameterization with gumbel-softmax,”

قسمت ۲) با توجه به مقاله β-VAE: LEARNING BASIC VISUAL CONCEPTS WITH ACONSTRAINED VARIATIONAL FRAMEWORK

متوجه میشویم که مسئله کلی بهینه سازی در *beta-vae* مسئله ی زیر است:

$$\max_{\varphi,\theta} E_{x\sim D}[E_{q_\varphi(z|x)}[\log p_\theta(x|z)]] \quad subject\ to\ D_{KL}(q_\varphi(z|x)||p(z)) < \epsilon$$

که در این مسئله *z* ها همان *generative latent factor* ها هستند که قرار است *observed data x* را تولید کنند.

حال اگر مسئله بهینه سازی بالا را تحت شرایط *KKT* به صورت لاگرانژین بازنویسی کنیم:

$$\mathcal{F}(\theta,\varphi,\beta,x,z) = E_{q_\varphi(z|x)}[\log p_\theta(x|z)] - \beta\left[D_{KL}(q_\varphi(z|x)||p(z)) - \epsilon\right]$$

در این رابطه، بتا همان ضریب *KKT* است که نوعی ضریب رگولاریزیشن است که وظیفه اش محدود کردن ظرفیت اطلاعات پنهان است.

تابع هزینه این شبکه با کمی تغییر در رابطه لاگرانژین آن برابر است با :

$$L(\theta,\varphi,\beta,x,z) = E_{q_\varphi(z|x)}[\log p_\theta(x|z)] - \beta D_{KL}(q_\varphi(z|x)||p(z))$$

اگر ما مقدار بتا را برابر با ۱ قرار دهیم، به تابع هزینه *ELBO* در *VAE* معمولی میرسیم.هرچه مقدار بتا بیشتر شود، پنالتی فاصله متغیر های پنهان کمتر شده و متغیر های پنهان در فضای *latent* میتوانند از هم فاصله بگیرند و *coding* بهتر صورت بپذیرد.با اینکار میتوانیم *disentangled representation of independent visual data generative factors* را یاد بگیریم و درجه *disentangled* را برای مدل تنظیم کنیم.

مسئله پنجم –

قسمت ۱) بخش الف) اگر مولد ثابت باشد و تمیز دهنده بهینه عمل کند، لذا مقدار بهینه تمیز دهنده برابر است با:

$$D^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)}$$

حال برای پیدا کردن نقطه ماکسیمم تابع هزینه داریم:

$$\min_g V(D^*,G) = \int \left[p_{data}(x)\log D^*(x) + p_g(x)\log\left(1 - D^*(x)\right)\right]dx$$

$$= \int \left[p_{data}(x)\log\frac{p_{data}(x)}{p_{data}(x)+p_g(x)} + p_g(x)\log\left(1 - \frac{p_{data}(x)}{p_{data}(x)+p_g(x)}\right)\right]dx$$

حال رابطه Jensen shanon را برای توزیع های p_{data} و p_g مینویسیم:

$$D_{JS}(p_{data}||p_g) = \frac{1}{2}D_{KL}\left(p_{data}\ ||\ \frac{p_{data}(x)+p_g(x)}{2}\right)$$

$$+\frac{1}{2}D_{KL}\left(p_g\,||\,\frac{p_{data}(x)+p_g(x)}{2}\right)$$

$$=\frac{1}{2}\left(\int p_{data}(x)\log\frac{2p_{data}(x)}{p_{data}(x)+p_g(x)}dx\right)+\frac{1}{2}\left(\int p_g(x)\log\frac{2p_g(x)}{p_{data}(x)+p_g(x)}dx\right)$$

$$=\frac{1}{2}\left(\log 2+\int p_{data}(x)\log\frac{p_{data}(x)}{p_{data}(x)+p_g(x)}dx\right)$$

$$+\frac{1}{2}\left(\log 2+\int p_g(x)\log\frac{p_g(x)}{p_{data}(x)+p_g(x)}dx\right)$$

$$=\frac{1}{2}\left(\log 4+\min_g V(D^*,G)\right)$$

$$\rightarrow \min_g V(D^*,G)=2D_{JS}(p_{data}||p_g)-2\log 2$$

قسمت ۱) بخش ب) همان طور که در بخش قبل دیدیم، $\min_g V(D^*,G)=2D_{JS}(p_{data}||p_g)-2\log 2$ است پس اگر میخواهیم تابع $V(D^*,G)$ را بر حسب g کمینه کنیم، باید تا حد امکان $D_{JS}(p_{data}||p_g)$ را کمینه کنیم، که کمترین حالت آن وقتی است که $p_{data}=p_g$ که در این حالت $D_{JS}(p_{data}||p_g)=0$ میشود لذا مقدار بهینه تابع میشود:

$$=\min_g V(D^*,G)=-2\log 2$$

قسمت ۱) بخش پ) در مقالات مختلف، نسبت های مختلفی بیان شده است برای مثال به ازای هربار آپدیت مولد، ۲ یا ۳ یا ۵ بار تمیز دهنده را آپدیت کنیم.خواسته این سوال کمی برایم واضح نیست. اما اینگونه برداشت کردم که برای رسیدن به نتایج بخش های الف و ب ، چه نسبتی در نظر بگیریم، یعنی باید تمیزدهنده خیلی خوب عمل کند، برای این کار، میتوانیم به ازای هربار آپدیت مولد، آنقدر تمیزدهنده را آپدیت کنیم که به یک local optimal برسد و سپس با آپدیت دوباره مولد، سراغ تمیز دهنده برویم.

قسمت ۲) بخش الف) داریم:

$$\frac{\partial}{\partial a}\log\big(1-D(x)\big)=\frac{\partial}{\partial a}\log\big(1-\sigma(a)\big)=\frac{\partial}{\partial a}\log\left(1-\frac{1}{1+\exp(-a)}\right)$$
$$=\frac{\partial}{\partial a}\log\left(\frac{1}{1+\exp(a)}\right)=\frac{\partial}{\partial a}-\log(1+\exp(a))=-\frac{\exp(a)}{1+\exp(a)}=-\sigma(a)$$

قسمت ۲) بخش ب) طبق گفته بخش الف سوال، اگر دامنه توزیع شبکه مولد و داده ها همپوشانی نداشته باشند،اگر تمیز دهنده بهینه عمل کند$D(x)=\sigma(a)$ به سمت صفر میل میکند لذا گرادیان ناپدید میشود به مرور زمان و بنابراین گرادیانی که برای مولد ارسال میشود صفر میشود و مولد آموزش نمیبیند.

قسمت ۲) بخش پ) در این صورت:

$$\frac{\partial}{\partial a}-\log\big(D(x)\big)=\frac{-\exp\,(-a)}{1+\exp\,(-a)}=\sigma(a)-1$$

لذا در شرایط گفته شده قسمت قبل، در این حالت اگر $D(x)=\sigma(a)$ به صفر میل کند، گرادیان ارسالی برای مولد ۱- میشود و مشکل محو شدن گرادیان حل میشود.

قسمت ۳) بخش الف) نقاط زینی تابع *f(x,y)* یعنی نقاطی که در شرایط زیر صدق کنند:

$$\frac{\partial f}{\partial x}=0,\quad \frac{\partial f}{\partial y}=0,\quad \frac{\partial^2 f}{\partial x^2}\frac{\partial^2 f}{\partial y^2}-\left[\frac{\partial^2 f}{\partial x\partial y}\right]^2<0$$

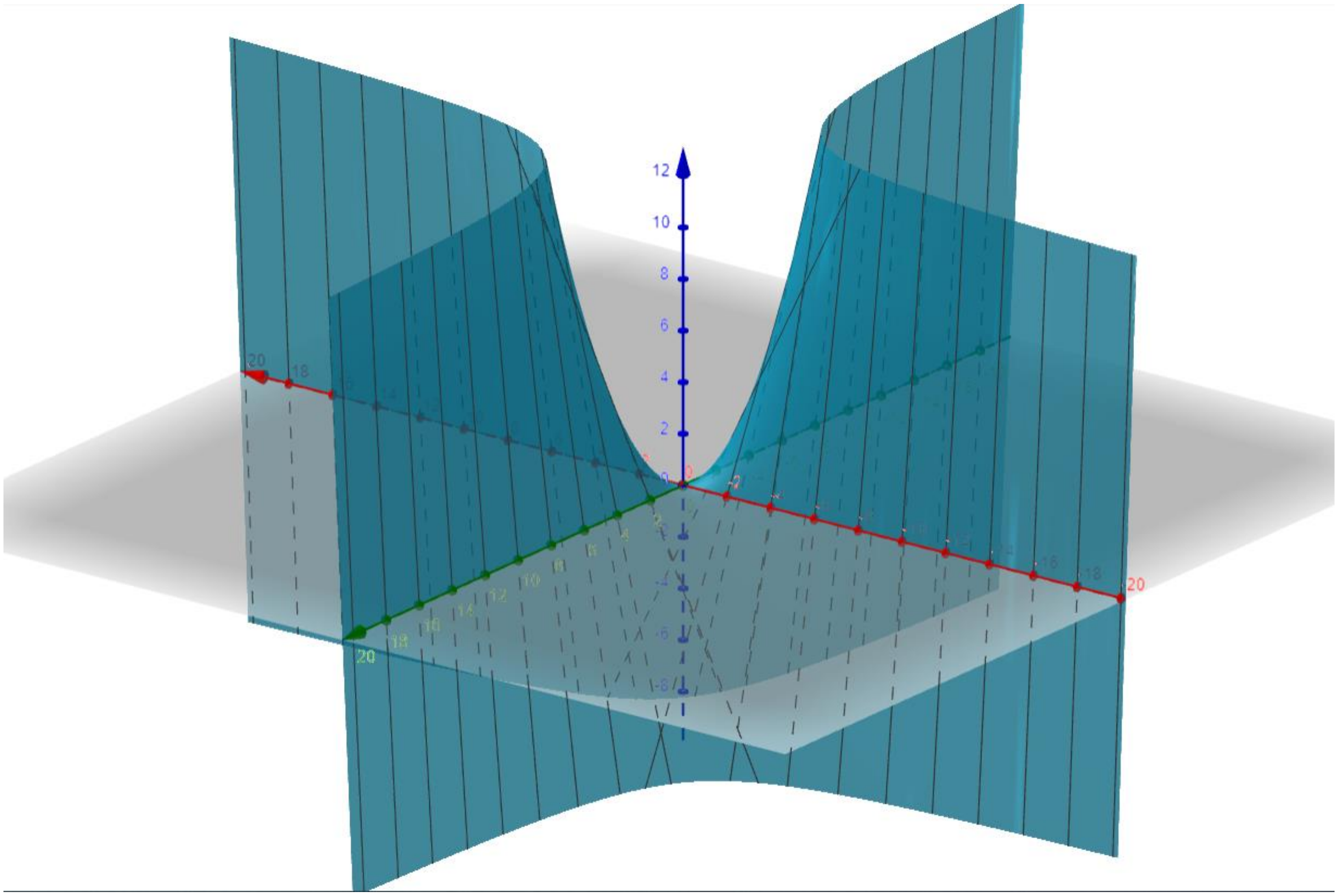
حال برای *f=xy* حساب میکنیم:

$$\frac{\partial xy}{\partial x}=0\rightarrow\frac{\partial xy}{\partial x}=0\rightarrow y=0$$
$$\frac{\partial xy}{\partial y}=0\rightarrow\frac{\partial xy}{\partial y}=0\rightarrow x=0$$

$$\frac{\partial^2 xy}{\partial x^2}\frac{\partial^2 xy}{\partial y^2}-\left[\frac{\partial^2 xy}{\partial x\partial y}\right]^2<0\xRightarrow{(x,y)=(0,0)}\frac{\partial^2 xy}{\partial x^2}\frac{\partial^2 xy}{\partial y^2}-\left[\frac{\partial^2 xy}{\partial x\partial y}\right]^2=0*0-1^2=-1<0$$

لذا تنها نقطه زینی این رویه، نقطه *(0,0)* است.

قسمت ۳) بخش ب) رویه تابع *f=xy* به صورت زیر است:



اگر فرض کنیم از نقطه *(1,1)* شروع کرده و نرخ آپدیت ها در هر *iteration* مثلا *0.2* باشد، اتفاق زیر می افتد:

$$itr\ 1\text{: } (1{,}1) = 1,$$

$$itr\ 2\text{: } (1.2\ , 0.8) = 0.96,$$

$$itr\ 3\text{: } (1.4\ , 0.6) = 0.84,$$

$$itr\ 4\text{: } (1.6\ , 0.4) = 0.64,$$

$$itr\ 5\text{: } (1.8\ , 0.2) = 0.36,$$

$$itr\ 6\text{: } (2{,}0) = 0$$

از اینجا به بعد ، چون *y* سعی میکند منفی شود و حاصل کل را *min* کند، *x* سعی میکند به سمت منفی ها بیاید تا وقتی منفی شد، حاصل را مثبت کند و جلوی *y* را بگیرد:

$$itr\ 7\text{: } (1.8, -0.2) = -0.36,$$

$$itr\ 8\text{: } (1.6\ , -0.4) = -0.64,$$

$$itr\ 9\text{: } (1.4\ , -\ 0.6) = -0.84,$$

$$itr\ 10\text{: } (1.2\ , -\ 0.8) = -0.96,$$

$$itr\ 11: (1\ , -1) = -1,$$

$$itr\ 12: (0.8, -1.2) = -0.96,$$

$$itr\ 13: (0.6, -1.4) = -0.86,$$

$$itr\ 14: (0.4, -1.6) = -0.64,$$

$$itr\ 15: (0.2, -1.8) = -0.36,$$

$$itr\ 16: (0, -2) = 0$$

حال از اینجا به بعد رویه برعکس میشود یعنی x سعی میکند توی منفی ها زیاد شود و تابع را بیشینه کند و y سعی میکند به سمت مثبت ها برود و باعث شود جلوی x را بگیرد. همانطور که میبینیم، دائما مقدار تابع هدف، بین ۱- تا ۱ تناوب میکند و هیچ گاه هم این تابع همگرا نمیشود.

قسمت ۴) داریم:

$$\mathcal{L}_{MLE}(\theta) = E_{x\sim p_{Data}}[-\log p_G(x)] \rightarrow \nabla_\theta \mathcal{L}_{MLE}(\theta) = E_{x\sim p_{Data}}[-\nabla_\theta \log p_G(x)]$$

$$\mathcal{L}_{MLE-GAN}(\theta) = E_{x\sim p_G}[f(x)] = \int p_G(x) * f(x)dx$$

$$\rightarrow \nabla_\theta \mathcal{L}_{MLE-GAN}(\theta) = \int \nabla_\theta p_G(x) * f(x)dx = E_{x\sim p_g}[\nabla_\theta \log p_G(x) f(x)],$$

$$\mathcal{L}_{MLE}(\theta) = \nabla_\theta \mathcal{L}_{MLE-GAN}(\theta) \rightarrow E_{x\sim p_{Data}}[-\nabla_\theta \log p_G(x)] = E_{x\sim p_g}[\nabla_\theta \log p_G(x) f(x)]$$

$$\rightarrow f(x) = \frac{-p_{Data}}{p_g} \rightarrow D^* = \frac{p_{Data}}{p_g + p_{Data}} \overset{\overline{p_g}}{\Rightarrow} D^* = \frac{\frac{p_{Data}}{p_g}}{\frac{p_g}{p_g} + \frac{p_{Data}}{p_g}}$$

$$\rightarrow D^* = \frac{f(x)}{f(x) - 1} \rightarrow f(x) = e^{-\alpha}$$

قسمت ۵) بخش الف) داریم:

$$KL(p_g||p_{Data}) = \int_{-2}^{+2} p_g(x)\log\frac{p_g(x)}{p_{Data}(x)}dx$$

$$= \int_{-2}^{-1} p_g(x)\log\frac{p_g(x)}{p_{Data}(x)}dx + \int_{-1}^{1} p_g(x)\log\frac{p_g(x)}{p_{Data}(x)}dx + \int_{1}^{2} p_g(x)\log\frac{p_g(x)}{p_{Data}(x)}dx$$

$$= \int_{-2}^{-1} 0 * \log\frac{0}{1}dx + \int_{-1}^{1} 0 * \log\frac{0}{0}dx + \int_{1}^{2} 1 * \log\frac{1}{0}dx = 0 + 0 + \infty = \infty$$

$$KL(p_{Data}||p_g) = \int_{-2}^{+2} p_{Data}(x)\log\frac{p_{Data}(x)}{p_g(x)}dx$$

$$= \int_{-2}^{-1} p_{Data}(x)\log\frac{p_{Data}(x)}{p_g(x)}dx + \int_{-1}^{1} p_{Data}(x)\log\frac{p_{Data}(x)}{p_g(x)}dx + \int_{1}^{2} p_{Data}(x)\log\frac{p_{Data}(x)}{p_g(x)}dx$$

$$= \int_{-2}^{-1} 1 * \log\frac{1}{0}dx + \int_{-1}^{1} 0 * \log\frac{0}{0}dx + \int_{1}^{2} 0 * \log\frac{0}{1}dx = \infty + 0 + 0 = \infty$$

$$JS(p_{data}||p_g) = \frac{1}{2}KL\left(p_{data} \,||\, \frac{p_{data}(x) + p_g(x)}{2}\right) + \frac{1}{2}KL\left(p_g \,||\, \frac{p_{data}(x) + p_g(x)}{2}\right)$$

$$= \frac{1}{2}\left(\int_{-2}^{-1} \log 2\, dx\right) + \frac{1}{2}\left(\int_{1}^{2} \log 2\, dx\right) = \log 2$$

قسمت ۵) بخش ب) چون فاصله Jensen ثابت است، لذا گرادیان صفر به مولد میرسد و آپدیت نمیشود.

قسمت ۵) بخش پ) مشکل صفر شدن گرادیان از جایی نشات میگیرد که دامنه توزیع مولد و تمیزدهنده مستقل است و فاصله آن ها همیشه ثابت میشود. حال با اضافه کردن نویز، این مستقل بودن دامنه ها را برهم میزنیم و فاصله دیگر ثابت نمیشود و گرادیان صفر به مولد نمیرسد.

قسمت ۶) بخش الف) دو شرطی که باید در این توابع صدق کند:

$$condition\ 1\colon \int \gamma(x,y)dy = p(x),$$
$$condition\ 2\colon \int \gamma(x,y)dx = p(y)$$

قسمت ۶) بخش ب) اگر روش جا به جایی را به شکل زیر در نظر بگیریم:

$$\gamma(x,y) = \delta(x+y)$$

داریم:

$$L = \int_1^2 \int_{-2}^{-1} \delta(x-y)|x-y|dxdy = \int_1^2 \int_{-2}^{-1} \delta(x+y)(y-x)dxdy$$
$$= \int_1^2 \int_{-2}^{-1} -2x\ dxdy = 3$$

قسمت ۶) بخش پ) در این فاصله ، فاصله همیشه ثابت نمیشود و گرادیان صفر به مولد نمیرسد.از طرفی فاصله بی نهایت هم نمیشود.لذا معیار بهتری است.