تمرین ۳ دیپ لرنینگ

۱.۱) $$\nabla_\phi E_{q_\phi(z)}[f(z)] = \nabla_\phi \int f(z)\, q_\phi(z)\, dz$$

$$= \int f(z)\, \nabla_\phi q_\phi(z) \cdot dz \overset{*}{=} \int f(z)\, q_\phi(z) \nabla_\phi \ln q_\phi(z)\, dz$$

$$= \underset{q_\phi(z)}{E}[f(z)\, \nabla_\phi \ln q_\phi(z)] \underset{\text{طبق قانون اعداد بزرگ}}{\approx} \frac{\sum_{i=1}^{n} f(z^i)\, \nabla_\phi q_\phi(z^i)}{n}$$

$$* \quad \nabla_\phi \ln q_\phi(z) = \nabla_\phi q_\phi(z) \times \frac{1}{q_\phi(z)}$$

حل $$L = \sum_i E_z[\log p_\theta(x^{(i)}|z)] - D_{KL}(q_\phi(z|x^{(i)}) \| p_\theta(z))$$

$$\underset{f(z)}{=} \sum_i E_q(f(z)) + \Sigma_i H(q(z|x))$$

که $\nabla_\phi H$ قابل محاسبه است، و $\nabla_\phi f$ هم در بالا توضیح دادیم چگونه به دست می‌آید پس کل ELBO را تخمین زدیم

۲.۱) روش بالا یک ایراد دارد آن هم این است که واریانس f اگر زیاد باشد

به ازای n تا sample برای $cov(\bar{X}) = \frac{cov(X)}{n}$ داریم

Date: / / Subject:

متغیرِ تصادفی Xها این نتیجه را می‌دهد که اگر COV(X)

عدد خیلی بزرگی باشد تعداد sample های زیادی نیاز است تا

$\mathrm{cov}(\bar{x})$ معقول باشد

حالا برای f یک تابع کمکی تعریف می‌کنیم بنام g

$$\hat{f} = f - a(g - E[g])$$

$$\mathrm{var}\,\hat{f} = \mathrm{var}f - 2a\,\mathrm{cov}(f,g) + a^2\,\mathrm{var}(g)$$

که برحسب a مشتق می‌گیریم تا مینیمم $\mathrm{var}\,\hat{f}$ را برحسب a به دست

آوریم.
$$a = \frac{\mathrm{cov}(f,g)}{\mathrm{var}(g)}$$

a را دوباره وارد معادله می‌کنیم ⇐ $\frac{\mathrm{var}f}{\mathrm{var}f} = 1 - \mathrm{corr}(f,g)^2$

⇐ هرچه corr ، f و g بیشتر باشد واریانس $\hat{f}$ کاهش

بیشتری داشته. ما فقط کافی است یک ~~f~~ g که خیلی با f correlation

دارد پیدا کنیم و $E_q(\hat{f})$ را به جای $E_q(f)$ تخمین بزنیم

با این کار، یک approximation از $E_q(f)$ خواهیم داشت ولی در عوض

تخمین موثرتر که در $E_q(\hat{f})$ خیلی دقیق‌تر خواهد بود

۳.) $\sum(\hat{x} - x^{(i)})^2$ که $\hat{x}$ از توزیع $p(x|z)$

می‌آید که z خودش از توزیع $q(z|x)$ می‌آید

این در واقع یک تخمینی از $E_{\hat{x}\sim p(\hat{x}|z),\, z\sim q(z|x)} (\hat{x}-x)^2$

است که این معادل است با $E_{z\sim q(z|x)}\left[\int (\hat{x}-x)^2 \log p(\hat{x}|z) \cdot d\hat{x}\right]$

که ~~[illegible]~~ این کار یعنی به ازای $\hat{x}$ های نزدیک x

تلاش می‌کند که $\log p(\hat{x}|z)$ بزرگ کند زیرا به ازای آن‌ها فقط به ازای

$(\hat{x}-x)^2$ دارد هزینه کمی می‌دهد ولی در عوض به ازای $\hat{x}$ های دور از x

$\log(p(x|z))$ را ~~[illegible]~~ تلاش می‌کند کوچک کند چون $(x-\hat{x})^2$ که بزرگ است هزینه ده

حالا تابع ما چه بود $\max E_{q(z|x)}[\log p(x|z)]$ که دارد تلاش می‌کند

$\log(x|z)$ را زیاد کند که اگر $p(x|z)$ تابعی پیوسته باشد

این در معادل خواهد بود

۴.) term دوم که KL divergense است فاصله

$q(z|x)$ ، $p(z)$ را نشان می‌دهد معمولاً $p(z)$ را یک چیز ساده

مثل $N(0,1)$ می‌گیرند حالا این باعث می‌شود که q تلاش کند

Date: / / Subject:

ساختار خیلی پیچیده ای به خود نگیرد و به آن ساختار ساده نزدیک باشد => ما با این جمله داریم روی q و محدودیت اعمال می کنیم => فضای z خیلی باز نخواهد بود که هر شکلی به خود بگیرد => این کار جلوی overfit را خواهد گرفت زیرا احتمالاً اگر می خواست خیلی overfit کند باید z با فضای خیلی پیچیده ای کار کند این کار خیلی شبیه regularization term در رگرسیون است

البته کارهایی مثل dropout هم موثر خواهد بود

$$p(x_1, x_2, z) = p(x_1)\, p(z)\, p(x_2 | x_1, z) \quad (1.1.2$$

$$\Rightarrow \int p(x_1, x_2, z) \, dz = p(x_1) \int p(x_2 | x_1, z)\, p(z) \, dz$$

$$\Rightarrow \frac{p(x_2, x_1)}{p(x_1)} = \int p(x_2 | x_1, z)\, p(z) \, dz$$

$$\Rightarrow p(x_2 | x_1) = \int p(x_2 | x_1, z)\, p(z) \, dz$$

=> همان conditional vae است

$$E_{z\sim q(z|x_1,x_2)}[\log(x_2|x_1,z)] - D_{KL}(q(z|x_1,x_2)||p(z))$$

را باید مینیمم کرد   فقط به جمع شود

$$\frac{p(x_1,x_2,z)}{p(x_2|x_1,z)} = p(x_1)p(z)$$ ([illegible]

$\Rightarrow p(x_1,z) = p(x_1)p(z) \Rightarrow$ x_1, z مستقل اند

که می‌توان جای $q(z|x_1,x_2)$ یکجوری از $q(z|x_2)$ expectet

گرفت که فرقی نمی‌کند

$p(x_1,x_2,z) = p(z)p(x_1|z)p(x_2|z)$ (۲.۱.۳

پس $p(x_1,x_2,z) = p(x_2|z)p(x_1,z)$

$\Rightarrow p(x_2|x_1,z) = p(x_2|z) \Rightarrow$

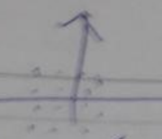
$$\int p(x_2|x_1,z)p(z)\,dz = \int p(x_2|z)p(z)\,dz$$

$\Rightarrow p(x_2|x_1) = p(x_2) \Rightarrow$ $p(x_1x_2)$ مستقل اند x_1,x_2

فقط کافی است رابطه را برای x_2 نوشت؟

$\hookrightarrow E_{z\sim q(z|x_2)}[\log p(x_2|z)] - D_{KL}(q(z|x_2)||p(z))$

Date:



Subject:

۱.۱.۲) من اول با h=16 شروع کردم و کیفیت عکس‌ها واقعاً
فاجعه است با h=64 تغییر دادم کیفیت پیشرفت قابل توجهی کرد
بعد با h=128 عوض کردم خیلی تغییر نکرد در کیفیت در کل ۳۰
ایپاک آموزش دادم و کیفیت به میزان قابل توجهی از GAN بهتر بوده است
همانطور که گفته شده برای اینکه ۲.۱.۱ بهتر بشود صرفاً کافی است به x ها
conditional شود نکته‌ای که هم کلاس هم اشاره شد E log(P(x|z)) را
log P(x|z) تخمین می‌زنیم روی batchsize تا تمام جمع بشود تخمینی از
$E_{q(z|x)}(\log P(x|z))$ خواهیم بود چون هر z از توزیع q(z|x) می‌آید

۲.۱.۲) برای اینکه GAN را بتوانیم آموزش دهم از dropout استفاده کردم (دقت برای G)
خروجی discriminator باید بین ۰ و ۱ باشد پس sigmoid می‌گذاریم و از طرفی
چون relu خوب train نمی‌شد در GAN تمام activation های
hidden ها را LeakyRelu گذاشتم GAN را هم ۳۰۰ ایپاک
گذاشتم Gain شد

۳.۱.۱) شهود ساختن vae-gan این است که میخواهیم

decoder یک generator است و یک discriminator میگذاریم که $\hat{x}$ را از

x تشخیص دهد بنابراین ترکیب ۲ به loss های یک عبارت

loss gan اضافه شده که $\log(Dis(x)) + \log(1-dis(\hat{x}))$

است اضافه شده از detail مقاله را نگاه کردم

هر ۲ loss تعریف کرده

$$L_{Prior} = D_{KL}(q(z|x)||p(z))$$

$$L^{Dis}_{likelihood} = -E_{q(z|x)}[P(Dis(x)|z)]$$

$$LGAN = \log(Dis(x)) + \log(1-Dis(\hat{x})) + \log(1-Dis(x_p))$$

که x_p از Decoder با z رندوم تولید شده

عوض کنیم

و اول پارامترهای enc را به هدف مینیمم کردن $L_{prior} + L^{Dis}_{like}$ است

عوض کنیم

دوم پارامترهای dec را به هدف مینیمم کردن ~~SL~~ $L^{Dis}_{like} - L_{GAN}$ را

عوض کنیم

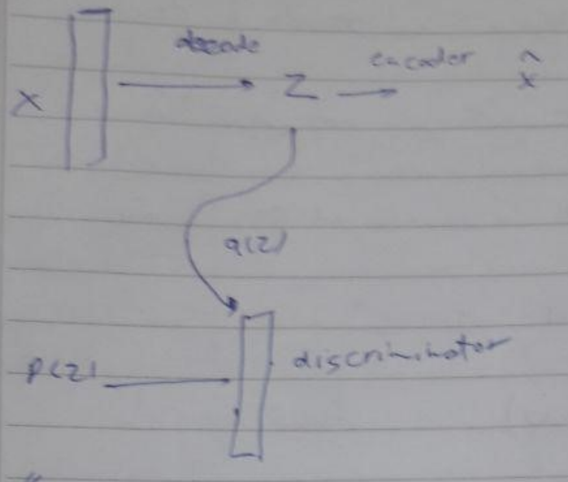
بعد پارامترهای dis را به هدف ماکسیمم کردن L_{GAN}

ولی این کارها در عمل دیده شده بهتر جواب میدهد

تا اینکه کل‌ها را ماکسیمم کردن $L_{GAN} + L_{Prior} + L^{Dis}_{Like}$

است همان loss vae است به loss discriminator

(۲-۱-۳



همان autoencoder عادی را هم از decoder به discriminator می‌دهیم

تا q(z) را از p(z) تشخیص دهد

distribution of data

$$q(z) = \int q(z|x)\, p_d(x)\, dx$$

یعنی طوری $\hat{x}$ سعی می‌کند به x نزدیک شود با این شرایط که

q(z) هم به p(z) نزدیک شود این شباهت زیاد که به vae دارد زیرا

$$E_x[KL(q(z|x) \| p(z))] = -E_x[H(q(z|x))] + E_{q(z)}[-\log p(z)]$$

که این اولی واریانس q(z|x) را [illegible]

و دومی cross entropy ، p(z), q(z)

Date: / / Subject:

۱ در آ عبارت را با شبیه کردن p(z) و q(z) جابه جای کنیم

کنیم که $E_{q(z|x)}(\log p(x|z))$ هم مانند همان $(\hat{x}-x)^2$ است

پس در کل شبیه vae می‌شود

۳.۱۲) aae از لینک های hwalsuklee برداشتم

عکسهایی که گذاشته برای z ، greed شده روی فضا z را اجرا کرد

داده و عکس های متفاوت را گرفته از آنجا که مشکل گرد بود تغییر

ندادم و

vae-gan از لینک های jlindsey15 برداشتم

برای همه ی مدل ها batch-size را ۲۵۶ گرفتم

و لایه پنهان استفاده شده در تمام مدل‌ها را ۱۲۸ گرفتم

و همینطور ۱۰۰ ایپاک برای همه پیش رفتم

$$E_{x\sim P_{data}} \log D(x) + E_{z\sim p(z)} \log(1-D(G(z))) \quad (۱.۱.۴$$

$$= E_{x\sim P_{data}} \log D(x) + E_{x\sim P_g} \log(1-D(x))$$

$$= \int P_{data}(x) \log D(x)\, dx + \int P_g(x) \log(1-D(x))\, dx$$

برای یک x در x0 حاصل از تابع برحسب P(x0) مشتق بگیریم

$$\Rightarrow \frac{\nabla L}{\nabla D(x_0)} = \int \frac{P_{data}(x_0)}{D(x_0)} - \frac{P_g(x_0)}{1-D(x_0)} \cdot dx_0$$

که D(x0) به x وابسته نیست و اصلا انتگرال می رود

$$\Rightarrow$$

$$\frac{D^*(x_0)}{1-D^*(x_0)} = \frac{P_{data}(x_0)}{P_g(x_0)}$$

برای هر x

ماکسیمم در حالتی است که D(x) چنان باشد

$$\Lsh \quad D^*(x) = \frac{P_{data}(x)}{P_g(x)+P_{data}(x)}$$

$$\min_{\theta_g} E_{z\sim p(z)}\left[\log(1-D^*(G_{\theta_g}(z)))\right] \qquad (٢.١.۴$$

فقط عبارت دوم می ماند

$$\Lsh \min_{P_g} \underset{x\sim P_g}{E}\left[\log(1-D^*(x))\right]$$

$$= \min_{P_g} \int P_g(x)\log(1-D^*(x)) \cdot dx$$

در هر x ثابت

loo

Date: / / Subject:

$D^*_g = \frac{1}{2}$ ⟸ $P_g = P_{data}$ اگر ~~[illegible]~~ [illegible]

$$E_{x\sim P_g}[\log(1-D^*_g(x))] = E \log 1/2$$

$$= \log 1/2 = -\log 2$$

حال برای P_gهای دیگر

$$L = -\log 4 + KL\left(P_{data} \| \frac{P_{data}+P_g}{2}\right)$$

$$+ KL\left(P_g \| \frac{P_{data}+P_g}{2}\right) = -\log(4) + 2JSD(P_{data} \| P_g)$$

که چون فاصله‌ها منفی نمی‌شوند آن می‌تواند کمینه باشد

است پس مینیمم L همان $-\log(4)$ است

⟸ $P_g = P_{data}$ حالت بهینه است

۳.۱۴) ایده‌ی اصلی که به کار رفته این است که وقتی یک تصویر با اندازه‌ی بزرگ داریم اول ~~resolution~~ تصویر را کم کنیم و یک generator و discriminator با لایه‌های کم و لایه‌های کوچک را روی آن learn کنیم و در طول زمان ~~[illegible]~~ آخرین لایه را در واقع کرده و سعی کنیم تصویر با resoloution دو برابر را

iman

Date: / /

Subject:

با دیگر بپره . یکی دیگر از چالش های train gan این است که

ممکن است result ها variation کمی داشته باشند که

گفته شده مقاله با روشی به اسم minibatch discrimination ، که در واقع

feature های هر عکس را در می آورد و به فضای دیگری map

می کند بعد این feature ها برای کل minibatch را concatenate

می کند و از جمع این اطلاعات discriminator را update می کند

روشی که در این مقاله ارائه داده که بسیار ساده است این است که

یک statistic از feature های هر عکس در می آورد و با هم جمع

می کند feature را با آن concatenate می کند این باعث می شود که

یک اطلاعات از کل عکس های minibatch را داشته باشد این کار

که روی لایه آخر بهترین جواب را گرفته ، statistic استفاده شده

standard deviation یک سطر روی فضای دو بعد آن است که بهترین نتیجه را گرفته

یک روش دیگر که پیشنهاد شده orthogonize کردن فضای feature

های generator است

Date: / /

Subject:

GAN ها معمولاً بصورت "unhealthy competition" پیشرفت شدید پیدا می‌کنند

و گرادیان ها کشیده می‌شود معمولاً برای اینکار از batch normalization استفاده می‌شود

این مقاله اشاره می‌کند که batch normalization برای کنترل واریانس بکار می‌رود

از این رو از روش های دیگری برای اینکار استفاده کرده است یکی اینکه

بجای اینکه از weigth initialization با دقت زیاد استفاده کنند هر weigth

را در هر مرحله تقسیم بر یک عدد ثابت می‌کنند دلیل آن این است که

optimizer های که استفاده می‌شود یعنی Adam و RMSprop معمولاً

گرادیان را نرمالایز می‌کنند از این رو weigth initialization تأثیرش کمرنگ می‌شود

کار دیگری که می‌کند این است که pixelwise ، عکس را نرمالایز می‌کند یعنی

عکس را در هر pixelو آن تقسیم بر یک میانگین ای از pixelهای کنارش

به دست آورده می‌کند