Redukovano GAN obučavanje

Kosta Grujčić, Momčilo Knežević

Matematički fakultet

Sažetak

Generativne suparničke mreže su postale veoma popularan generativan model naročito oblasti u računarskog vida. Nastale su mnogobrojne arhitekture, posebno dizajnirane za različite domene primene. Problem sa ovom vrstom modela je otežana konvergencija, izražena osetljivost na hiperparametre i veliki broj instanci skupa podataka nad kojim se vrši obučavanje. Odlučili smo da proverimo ovu tezu obučavajući DCGAN model nad redukovanim CelebA skupom podataka uvodeći dodatne uslove koji pomažu boljoj konvergenciji.

1 Uvod

Generativne suparničke mreže (GAN) [1] uživaju veliku popularnost kao generativni modeli. Nalaze uspešnu primenu u različitim domenima. Dizajnirane su mnogobrojne arhitekture poput [3], [5], [17], [18], [19]. Zajednička osobina većine GAN arhitektura je otežana konvergencija usled neograničenog gradijenta, te se nastoji uvođenju restrikcija [14]. Usled navedenog problema, potrebno je vršiti iscrpnu pretragu hiperparametara i koristiti veliki skup podataka za obučavanje.

Da bi ispitali kojoj u je meri ovo izraženo, izmenićemo poznatu DCGAN arhitekturu uvodeći spektralnu normalizaciju [4] i redukujući skup podataka CelebA [9]. Takođe, proverićemo da li spektralna normalizacija ima kontra-efekat – preprilagođavanje. Obučavanje kvantifikujemo poznatim metrikama IS (eng. *inception score*) [7] i FID (eng. *Fréchet Inception distance*) [8].

2 Pređašnji rad

U ovom poglavlju ćemo opisati princip rada GAN-ova, GAN arhitekturu koju koristimo i metrika kojima ćemo kvantifikovati obučenost modela.

2.1 Generativne suparničke mreže

Prvo dajemo neformalnu ideju. Osnovni cilj je naučiti raspodelu podataka tako da je moguće vršiti uzorkovanje, čime se dobija mogućnost generisanja podataka koji zapravo ne postoje. GAN arhitektura podrazumeva upotrebu dva modela. Prvi je zadužen za učenje pomenute raspodele, dok drugi treba da razlikuje stvarne i generisane podatke. Suparničkim učenjem ova dva modela se stvarna raspodela podataka sve bolje aproksimira.

Neka je $\mathcal{X} \subset \mathbb{R}^{c \times d \times d}$ prostor c-kanalnih slika dimenzija $d \times d$. Bilo koji skup slika možemo predstaviti nekom raspodelom p_r nad $\mathcal{X}$. Skup nad kojim obučavamo model možemo definisati kao prost slučajan uzorak iz raspodele p_r. Cilj je naučiti parametrizovanu raspodelu p_g koja aproksimira p_r.

GAN se sastoji od *generatora* i *diskriminatora*. Diskriminator posmatramo kao preslikavanje $D : \mathcal{X} \to [0, 1]$, dok generator kao $G : \mathcal{Z} \to \mathcal{X}$, gde je $\mathcal{Z}$ latentni prostor. Pomenuti latentni prostor služi za uzorkovanje šuma na osnovu kojeg generator pokušava da generiše elemente raspodele p_r. Ako takvo uzorkovanje šuma vršimo iz raspodele p_z definisane nad $\mathcal{Z}$, tada raspodelu p_g možemo definisati kao $G(p_z)$.

Napomenimo da male vrednosti diskriminatora odgovaraju generisanim podacima, dok veće stvarnim. Prema tome, generator nastoji generisanju podataka za koje diskriminator daje vrednosti bliske 0, dok diskriminator za takve ulaze nastoji da se sve viši približi vrednosti 1.

Obučavanje diskriminatora i generatora se vrši uporedo, usled čega se može videti kao min-max igra. Takav proces se može formalizovati:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_r}[\log D(x)] + \mathbb{E}_{z \sim p_z}[\log(1 - D(G(z)))]. \quad (1)$$

Primetimo da ne postoji ograničenje za izbor modela D i G. Međutim, pretpostavljajući njihovu diferencijabilnost i parametrizovanost, obučavanje se može vršiti propagacijom unazad i time u velikoj meri automatizovati.

U početnim iteracijama obučavanja može doći do zasićenja gradijenata [1]. Usled lošeg generatora, diskriminator je u stanju vrlo precizno da razlikuje lažne i prave podatke. S tim u vezi, optimizacija iz formule 1 se može zapisati u numerički pogodnijem obliku:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_r}[\log D(x)] - \mathbb{E}_{z \sim p_z}[\log D(G(z))]. \quad (2)$$

2.2 DCGAN

Generativne suparničke mreže ne podrazumevaju nikakav konkretan model ni za generator ni za diskriminator. S tim u vezi, zavisnosti od domena primene se mogu dizajnirati odgovarajući modeli. Kako se CelebA sastoji od slika,

to se opredeljujemo za arhitekturu baziranu na konvolutivnim neuronskim mrežama. Koristićemo DCGAN arhitekturu [3], pre svega zbog povoljnog vremena obučavanja.

Prvo ćemo objasniti strukturu diskriminatora ove arhitekture. Sastoji se 5 blokova koji vrše decimaciju prostornih dimenzija ulaznog tenzora za faktor 2. Svaki blok se sastoji od konvolutivnog sloja praćenog BN slojem [12], osim u poslednjem bloku gde ne postoji sloj normalizacije. Aktivaciona funkcija u svakom konvolutivnom sloju je ReLU [13], osim u poslednjem gde je sigmoid. Konvolutivni slojevi imaju redom $128, 256, 512, 1024, 1$ kanala. Decimacija je postignuta postavljanjem pomeraja konvolutivnog sloja na 2. Ulazni tenzor je trokanalna RGB slika, koja se prethodno normalizuje sa $\hat{\mu} = 0.5$ i $\hat{\sigma} = 0.5$.

Generator u odnosu na diskriminator ima inverznu strukturu. Koristi 4 ekspanziona bloka i transponovane konvolutivne slojeve umesto običnih. Broj kanala u konvolutivnim slojevima je redom $512, 256, 128, 3$. Pre nego što se latentni šum prosledi prvom konvolutivnom sloju, vrši se projektovanje potpuno povezanim slojem u prostor $\mathbb{R}^{1024\times4\times4}$. Aktivaciona funkcija poslednjeg sloja je hiperbolički tangens.

2.3 Lipšic neprekidnost generatora

Pretpostavimo da je diskriminator D neuronska mreža koja za ulaz $x \in \mathcal{X}$ ima sledeći oblik:

$$f(x, \theta) = \left(\prod_{i=1}^{L} a_i(\beta_i + W_i) \right)(x), \tag{3}$$

gde je $\theta = \{W_i\}_{i=1}^{L}$ skup parametara koji se obučavaju, $W_i \in \mathbb{R}^{d_i \times d_{i-1}}$ i $\{a_i\}_{i=1}^{L}$ aktivacione funkcije koje se primenjuju član po član. Napomenimo da d_0 predstavlja broj kanala ulaza, dok je $d_L = 1$.

Dokazano je [1] da za fiksiran generator G, optimalan diskriminator ima oblik:

$$D_G^*(x) = \frac{p_r(x)}{p_r(x) + p_g(x)} = \sigma(f^*(x)), \tag{4}$$

gde je σ sigmoid, a $f^*(x) = \log p_r(x) - \log p_g(x)$. Tada dobijamo:

$$\nabla_x D_G^*(x) \propto \nabla_x f^*(x) = \frac{1}{p_r(x)} \nabla_x p_r(x) - \frac{1}{p_g(x)} \nabla_x p_g(x). \tag{5}$$

Primetimo da gradijent $\nabla_x f^*(x)$ nije ograničen. Iz tog razloga dolazi do nepredvidivih oscilacija tokom obučavanja i česte divergencije. Zato je neophodno nametnuti dodatne uslove za f. Na osnovu [14], diskriminator obučavamo

u prostoru K-Lipšic neprekidnih funkcija[1]. Tada je gradijent iz formule 5 ograničen. Formula 1 postaje:

$$\min_G \max_{\|f\|_{\text{Lip}\leq K}} V(D,G), \tag{6}$$

gde je $\|f\|_{\text{Lip}\leq K}$ skup svih K-Lipšic neprekidnih funkcija u metričkom prostoru $(\mathcal{X}, l_2)$.

Spektralna normalizacija (SN) [4] se zasniva na uslovu $\|W_i\|_2 = 1$. Kako je $\|W_i\|_2 = \sqrt{\lambda_{\max}(W_i^T W_i)}$, to je neophodno odrediti najveću sopstvenu vrednost matrice $W_i^T W_i$. Pomenuti postupak je moguće sprovesti numerički.

U odnosu na standardnu DCGAN implementaciju, u našim eksperimentima je SN korišćena umesto BN i to isključivo kod diskriminatora.

2.4 IS

Adekvatno obučen GAN poseduje sigurnost i varijabilnost. To znači da generator može dati svojstvene podatke, kao i da ravnomerno pokriva čitav kombinatorni prostor.

Prepostavimo da svakoj instanci skupa podataka x možemo pridružiti oznaku y. Skup podataka takve oznake ne mora da sadrži, ali ih je moguće uvesti dodatnim modelom $\mathcal{M}$ (kao što je Inception [10]) koji je treniran kao klasifikacioni model na nekom drugom skupu podataka (kao što je ImageNet [11]). Zato možemo posmatrati uslovnu raspodelu $p_{\mathcal{M}}(y|x)$. Marginalna raspodela $p_{\mathcal{M}}(y)$ se dobija kao $\int_x p_{\mathcal{M}}(y|x)\mathrm{d}p_g(x)$.

Prema tome, $p_g(y|x)$ treba da odgovara Dirakovoj δ funkciji, a $p_g(y)$ ravnomernoj raspodeli. Konačno, IS definišemo:

$$\text{IS}(p_g) = \exp(\mathbb{E}_{x\sim p_g}[D_{KL}(p_{\mathcal{M}}(y|x)||p_{\mathcal{M}}(y))]), \tag{7}$$

gde je D_{KL} Kulbek-Lajblerova divergencija.

Ispostavlja se da ova metrika dobro korelira sa ljudskom odlukom [7]. Međutim, IS ni na koji način ne koristi p_r. Takođe, skup podataka nad kojim je $\mathcal{M}$ treniran možda nema ni približno sličnu raspodelu kao p_r usled čega GAN model biva proglašen lošim.

2.5 FID

Kao što smo već rekli, ideja obučavanja GAN-a je što bolja aproksimacija p_r raspodelom p_g. Ukoliko p_r predstavlja raspodelu podataka iz realnog sveta

[1] $(\exists K > 0)(\forall x, y \in \mathcal{D})\big\|f(x) - f(y)\big\| \leq K \cdot \|x - y\|$

(poput slika ljudi), tada možemo pretpostaviti ograničenost njenog nosača. U tom slučaju važi:

$$p_r \overset{s.s.}{=} p_q \iff (\forall k \in \mathbb{N}) \int_{\mathcal{X}} p_r(x) x^k \mathrm{d}x = \int_{\mathcal{X}} p_g(x) x^k \mathrm{d}x. \tag{8}$$

Drugim rečima, jednakost momenata povlači skoro sigurnu jednakost raspodela. U praksi je desnu stranu ekvivalencije 8 skoro nemoguće pokazati. Zato se ona aproksimira za vrednosti do nekog k. Mi ćemo aproksimaciju vršiti za $k = 2$.

Kako raspodele p_r i p_g nisu eksplicitno poznate, uvodimo operator ϕ tako da jednakost $\phi(p_r)$ i $\phi(p_g)$ odgovara jednakosti odgovarajućih raspodela. Kao i kod IS, koristimo Inception model čiji će neki unutrašnji sloj predstavljati operator ϕ. Inception model obučen nad ImageNet skupu podataka poseduje kvalitetnu aproksimaciju raspodele slika iz stvarnog sveta čiji duboki slojevi poseduju vizuelne karakteristike visokog nivoa. Iz tog razloga, za očekivati je da forsiranjem sličnosti $\phi(p_r)$ i $\phi(p_g)$ postižemo sličnost p_r i p_g.

Međutim, raspodele $\phi(p_r)$ i $\phi(p_g)$ takođe nisu eksplicitno poznate. Poznato je da za fiksirane momente prva dva reda normalna raspodela ima maksimalnu entropiju. Zato pretpostavljamo: $\phi(p_r) = \mathcal{N}(\mu_r, \Sigma_r)$ i $\phi(p_g) = \mathcal{N}(\mu_g, \Sigma_g)$. Razliku ovih raspodela merimo Frešeovim rastojanjem:

$$F(\phi(p_r), \phi(p_g)) = \left\|\mu_r - \mu_g\right\| + \mathrm{tr}\left(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{1/2}\right), \tag{9}$$

gde je tr operator traga matrice.

Konačno, definišemo $\mathrm{FID}(p_r, p_g)$ kao $F(\phi(p_r), \phi(p_g))$ [8]. Primetimo da za razliku od IS, FID koristi p_r.

3 Eksperimenti

Koristili smo dve arhitekture, DCGAN [3] i DCGAN-SN [4]. Za obučavanje je korišćena `Python` biblioteka `PyTorch` [15]. Svi eksperimenti su vršeni upotrebom servisa Google Colaboratory na kom smo imali pristup Nvidia T4 i Nvidia P100 grafičkim procesorima.

CelebA ima 202599 instanci. U našim eksperimentima je korišćeno približno 90000, što predstavlja redukciju od oko 55%. Ovaj skup podataka se sastoji od slika lica poznatih ličnosti različite boje kože, odeće i dodatnih karakteristika poput naočara, nakita ili šminke.

Sve slike su rezolucije 64×64. Obučavanje je vršeno ne više od 30 epoha. Korišćen je Adam optimizator za generator i diskriminator. U slučaju spektralne normalizacije, broj iteracija potrebnih za aproksimiranje najveće sopstvene vrednosti (k) je iz skupa $\{1, 2, 3\}$. Svi ostali parametri su korišćeni

u skladu sa [3]. Svi modeli su obučavani po 3 puta za svaku konfiguraciju i uzeta je srednja vrednost.

Dužina trajanja epohe je redom 14 i 5 minuta na T4 i P100.

3.1 Uticaj spektralne normalizacije

U ovom eksperimentu utvđujemo da li i koliko spektralna normalizacija poboljšava generator.

Pri istim uslovima poredimo rezultate modela sa i bez spektralne normalizacije. Obučavanje vršimo za sve dopustive vrednosti parametra k.

3.2 Merenje preprilagođenosti

U ovom eksperimentu utvrđujemo da li je došlo do preprilagođavanja modela i u kojoj meri.

Skup podataka S podelimo na dva podskupa S_{train} i S_{val}. Metriku koju koristimo za ocenu kvaliteta obučenosti modela označimo sa d. Skup slika generisanih od strane generatora označimo sa S_g. Model obučavamo nad S_{train}, dok se S_{val} ne koristi tokom obučavanja. Ukoliko model dobro generalizuje, tada $\frac{d(S_{train},S_g)}{d(S_{val},S_g)} \to 1$, dok u suprotnom $\frac{d(S_{train},S_g)}{d(S_{val},S_g)} \to 0$. Skup S delimo u razmeri $30:70$ i $10:90$.

4 Rezultati

5 Zaključak

Literatura

[1] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio. Generative Adversarial Nets. *arXiv preprint arXiv:1406.2661*, 2014.

[2] Qiantong Xu, Gao Huang, Yang Yuan, Chuan Guo, Yu Sun, Felix Wu, Kilian Weinberger. An empirical study on evaluation metrics of generative adversarial networks. *arXiv preprint arXiv:1806.07755*, 2018.

[3] Alec Radford, Luke Metz, Soumith Chintala. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *arXiv preprint arXiv:1511.06434*, 2016.

[4] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, Yuichi Yoshida. Spectral Normalization for Generative Adversarial Networks. *arXiv preprint arXiv:1802.05957*, 2018.

[5] Martin Arjovsky, Soumith Chintala, Léon Bottou. Wasserstein GAN. *arXiv preprint arXiv:1701.07875*, 2017.

[6] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, Aaron Courville. Improved Training of Wasserstein GANs. *aXiv preprint arXiv:1704.00028*, 2017.

[7] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen. Improved Techniques for Training GANs. *arXiv preprint arXiv:1606.03498*, 2016.

[8] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernard Nessler. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. *NIPS*, 2017.

[9] Ziwei Liu, Ping Luo, Xiaogang Wang, Xiaoou Tang. Deep Learning Face Attributes in the Wild. *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[10] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich. Going Deeper with Convolutions. *arXiv preprint arXiv:1409.4842*, 2014.

[11] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael

Bernstein, Alexander C. Berg, Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *arXiv preprint arXiv:1409.0575*, 2014.

[12] Sergey Ioffe, Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv preprint arXiv:1502.03167*, 2015.

[13] Vinod Nair, Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. *ICML: Proceedings of the 27th International Conference on International Conference on Machine Learning*, 2010.

[14] Guo-Jun Qi. Loss-Sensitive Generative Adversarial Networks on Lipschitz Densities. *arXiv preprint arXiv:1701.06264*, 2017.

[15] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *arXiv preprint arXiv:1912.01703*, 2019.

[16] Diederik P. Kingma, Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[17] Mehdi Mirza, Simon Osindero. Conditional Generative Adversarial Nets. *arXiv preprint arXiv:1411.1784*, 2014.

[18] Tero Karras, Samuli Laine, Timo Aila. A Style-Based Generator Architecture for Generative Adversarial Networks. *arXiv preprint arXiv:1812.04948*, 2018.

[19] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, Timo Aila. Analyzing and Improving the Image Quality of StyleGAN. *arXiv preprint arXiv:1912.04958*, 2019.