

Redukovano GAN obučavanje

Kosta Grujić Momčilo Knežević

Matematički fakultet

Sažetak

Generativne suparničke mreže su postale veoma popularan generativan model naročito u oblasti 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 potreba za velikim skupom podataka za 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 u kojoj je meri ovo izraženo, izmenićemo poznatu DCGAN arhitekturu uvodeći spektralnu normalizaciju [4] i redukujući skup podataka CelebA [9]. Obučavanje kvantifikujemo poznatim metrikama IS (eng. *inception score*) [7] i FID (eng. *Fréchet Inception distance*) [8]. Očekujemo da je moguće dobiti reprezentativno obučen model upotrebom spektralne normalizacije na redukovanim skupom podataka.

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 *diskriminadora*. Diskriminator posmatramo kao preslikavanje $D : \mathcal{X} \rightarrow [0, 1]$, dok generator kao $G : \mathcal{Z} \rightarrow \mathcal{X}$, 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 uporedno, 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 gradijentata [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 pogodnjem 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 diskriminatore. S tim u vezi, zavisnosti od domena primene se mogu dizajnirati odgovarajući modeli. Kako se CelebA sastoji od slike, to se opredeljujemo za arhitekturu baziranu na konvolutivnim neuronskim mrežama. Koristićemo DCGAN arhitekturu [3], pre svega zbog povoljnog vremena obučavanja.

Prvo ćemo objasnit strukturu diskriminatora ove arhitekture. Sastoje 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 diskriminatore 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 $\mathbb{R}^{1024 \times 4 \times 4}$. Aktivaciona funkcija poslednjeg sloja je hiperbolički tangens.

Funkcija greške je binarna unakrsna entropija. Iz formule 2 se vrlo lako dobija oblik funkcije greške za obe komponente.

2.3 Lipšic neprekidnost generatora

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

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

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 diskriminatore ima oblik:

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

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

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

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 funkcija¹. Tada je gradijent iz formule 5 ograničen. Formula 1 postaje:

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

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 $\int_x p_{\mathcal{M}}(y|x) dp_g(x)$.

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

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

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.

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

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

$$p_r \stackrel{s.s.}{=} p_g \iff (\forall k \in \mathbb{N}) \int_{\mathcal{X}} p_r(x) x^k dx = \int_{\mathcal{X}} p_g(x) x^k dx. \quad (8)$$

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 eksplisitno 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 eksplisitno 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 Freševim rastojanjem:

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

gde je tr operator traga matrice.

Konačno, definišemo 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 SN-DCGAN [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. Implementacija se na nalazi na GitHub repozitorijumu².

²<https://github.com/4eyes4u/RI-GAN>

Skup CelebA ima 202599 instanci. U našim eksperimentima je korišćeno 89931, 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 u trajanju 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.

U tablici 1 su prikazane vrednosti IS i FID metrika za sve konfiguracije.

model	IS	FID
DCGAN	1.93	27.72
SN(1)-DCGAN	1.84	24.61
SN(2)-DCGAN	1.80	24.54
SN(3)-DCGAN	1.80	23.39

Tablica 1: Poređenje metrika obučenih modela. SN(k)-DCGAN označava SN-DCGAN sa vrednošću parametra k .

4 Rezultati

Na osnovu tablice 1 primećujemo sledeće:

- Modeli su praktično identični prema IS metrići. To možemo objasniti izraženom robusnošću Inception modela kao i glavnim nedostakom IS metrike – ne uzima u obzir stvarne podatke. Za očekivati je da su \mathcal{M} raspodele svih modela veoma slične jer raspodele p_g u odnosu na \mathcal{M} predstavljaju slične raspodele ljudskih lica.
- Modeli se razlikuju prema FID metrići. Budući da FID metrika koristi p_r za očekivati je da postoji razlika u kvalitetu modela ukoliko spektralna normalizacija ima uticaj na obučavanje. Kako je na osnovu pojedinačnih primera jasno da to jeste slučaj, FID metrika to i potvrđuje. SN(3)-DCGAN očito jeste najbolji model, što svakako i treba da bude imajući u vidu da vrši najprecizniju aproksimaciju l_2 norme.

- Na slici 1 su istaknuti nasumično odabrani primeri iz odgovarajućih p_g raspodela. Primetan je definitivan uticaj spektralne normalizacije u obučavanju što je u saglasnosti sa FID metrikom. Slike koje DCGAN generiše su primetno isprane i lošije od ostalih modela. SN(k)-DCGAN modeli uglavnom generišu kvalitetne slike, uz poneke izuzetke u vidu neprirodnih boja ili distorzije lica.



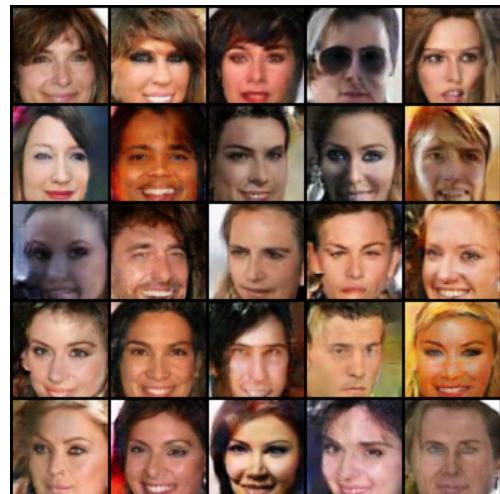
(a) DCGAN



(b) SN(1)-DCGAN



(c) SN(2)-DCGAN



(d) SN(3)-DCGAN

Slika 1: Poređenje rezultata nakon obučavanja. SN(k)-DCGAN označava SN-DCGAN sa vrednošću parametra k .

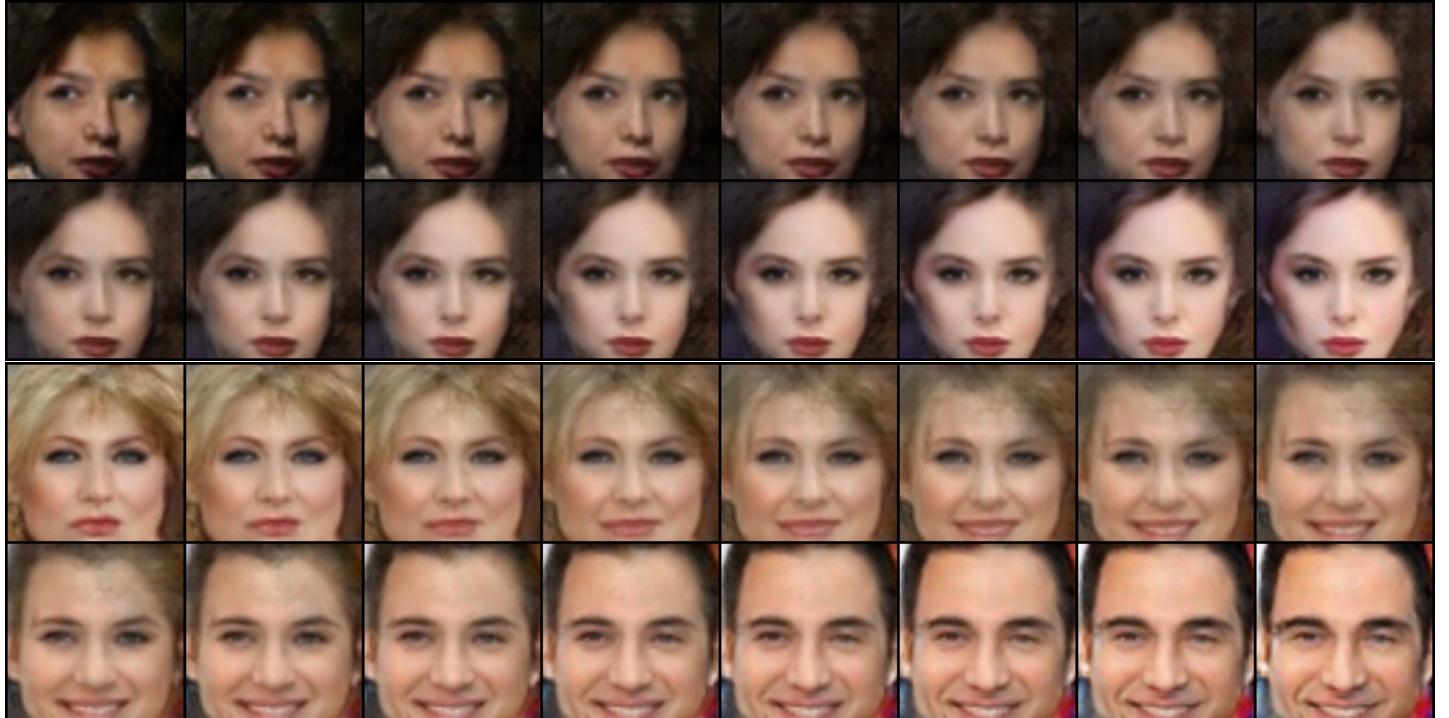
5 Zaključak

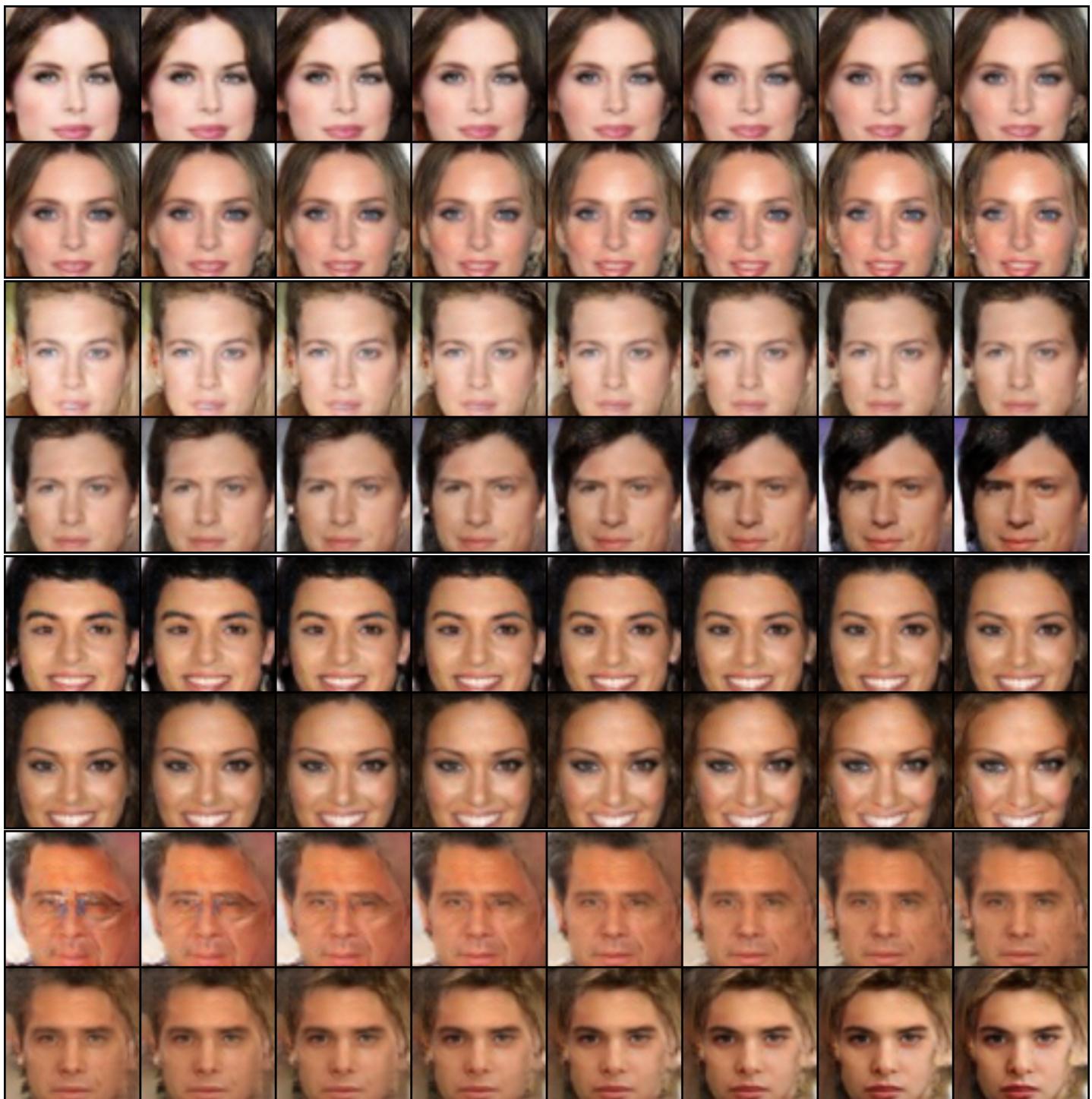
Na osnovu prikazanih rezultata uticaj spektralne normalizacije je neupitan. Takođe, pokazuje se da je moguće dobiti konkurentne modele sa redukovanim skupom za obučavanje što i jeste bio cilj ovog rada.

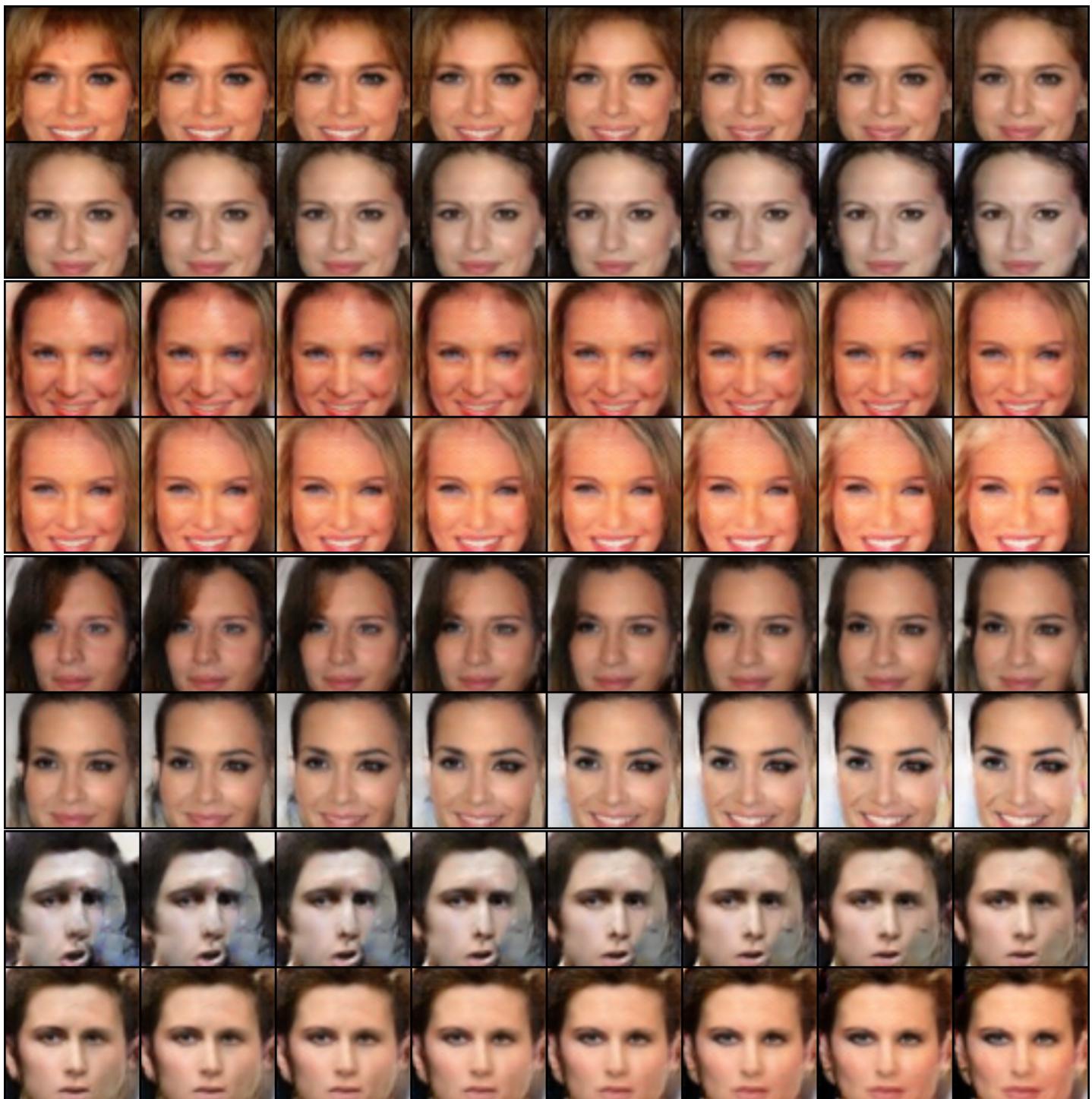
Interesantno je proveriti uticaj spektralne normalizacije na preprilagođenost modela. Mišljenja smo da spektralna normalizacija ne dovodi do preprilagođavanja jer isključivo utiče na glatkoću mnogostrukosti funkcije greške.

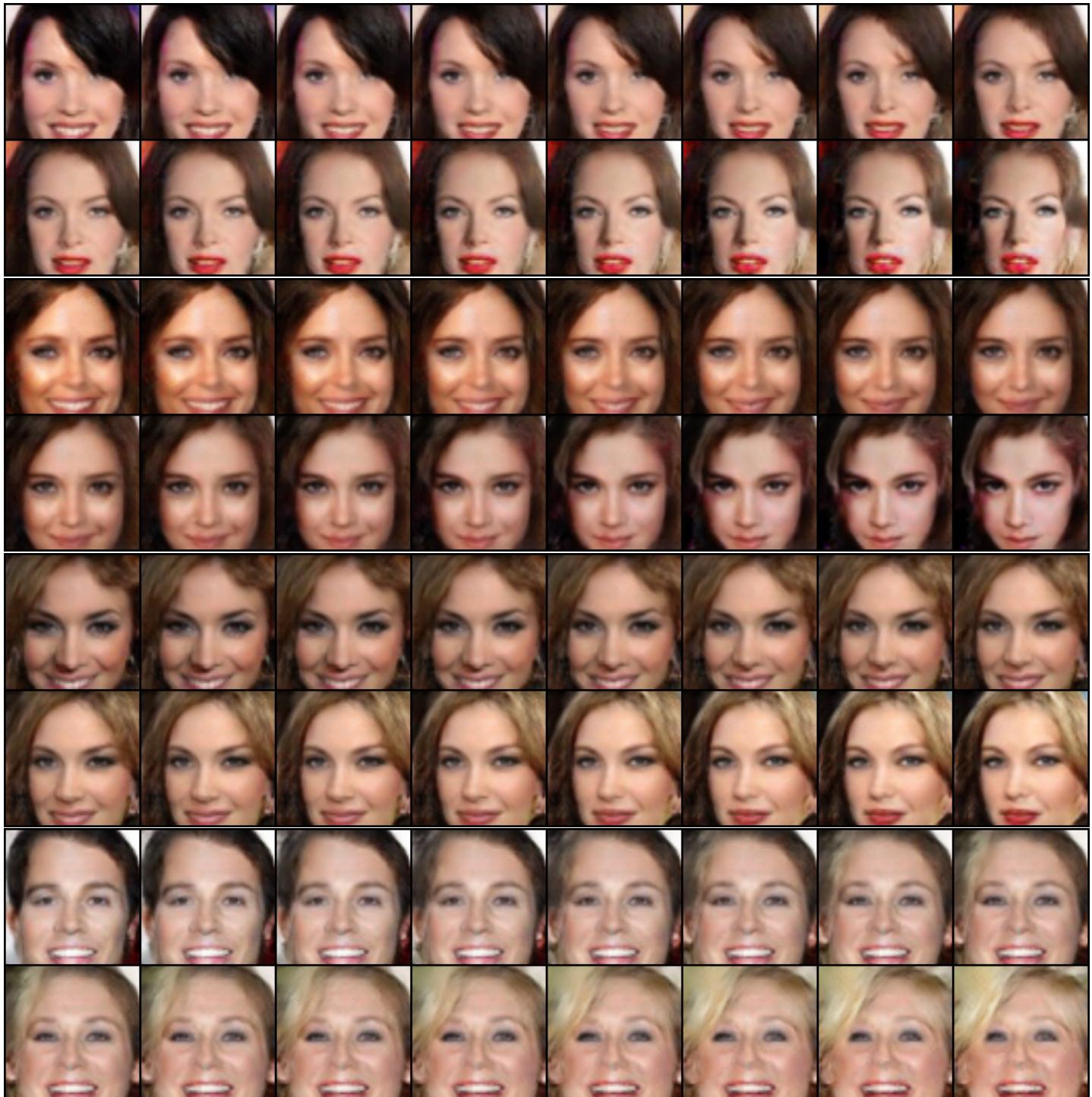
Dodatak A

Ovde prikazujemo nekoliko primera interpolacije upotrebom SN(3)-DCGAN modela. Slike su interpolirane upotrebom linearne interpolacije u 16 koraka. Prikazano je ukupno 14 interpolacija od kojih svaka zauzima 2 reda. Slike su izvorno rezolucije 64×64 , ali su ovde prikazane u rezoluciji 128×128 kao rezultat bilinearne interpolacije.





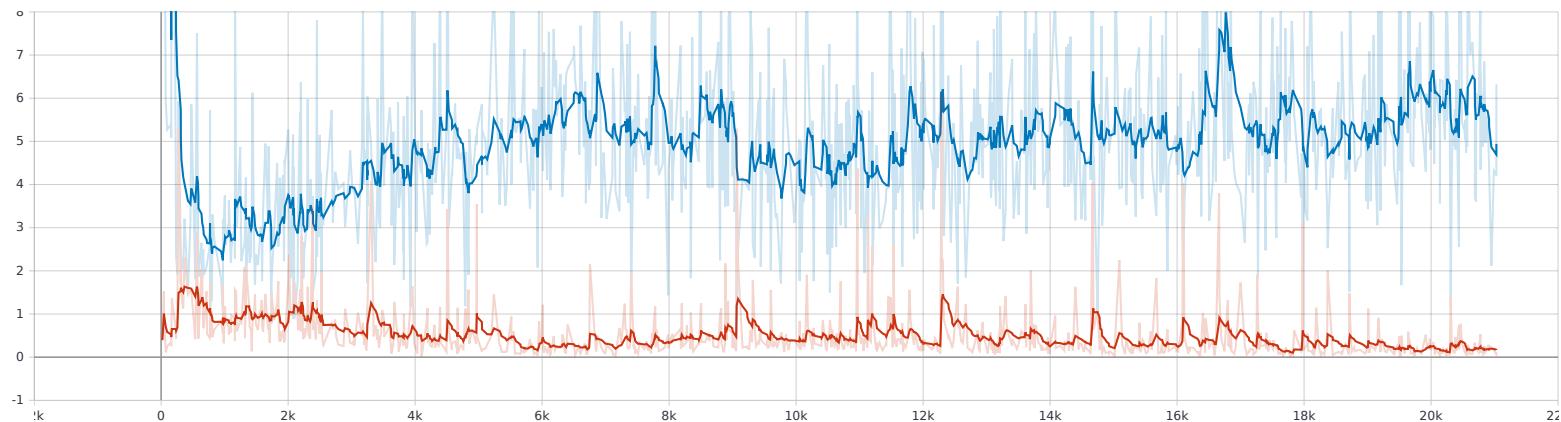




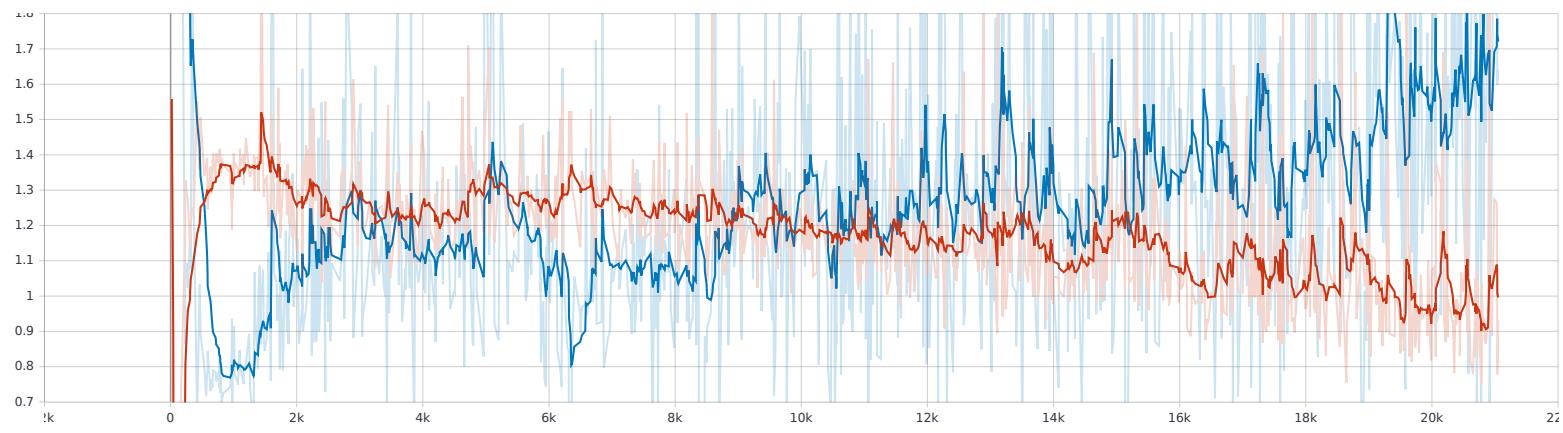
Dodatak B

Ovde prikazujemo funkciju greške svih modela tokom obučavanja. Da bi prikazali usrednjenu funkciju greške, grafici su čuvani u TensorBoard formatu i ufinjeni faktorom 0.85. Možemo primetiti sledeće:

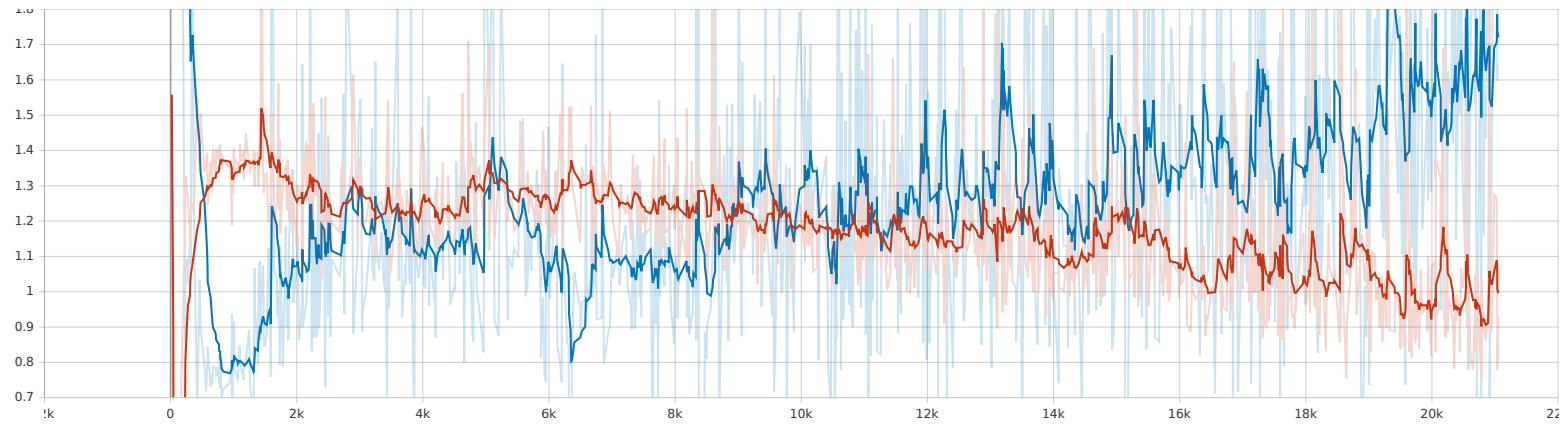
- U slučaju DCGAN modela brzo dolazi do stagnacije diskriminatora, čija je funkcija greške približno 0. Tada gradijent iz 5 praktično ne postoji i generator nije u stanju da napreduje.
- Funkcija greške diskriminatora u slučaju SN(k)-DCGAN modela evidentno opada tokom treninga, ali na asimptotski povoljniji način. Usled takvog ponašanja diskriminatora gradijent iz 5 neće ni nestati ni eksplodirati te se generator daleko bolja obučava.
- Bez obzira da li se spektralna normalizacija upotrebljava ili ne, primetan je značaj šum u obučavanju.



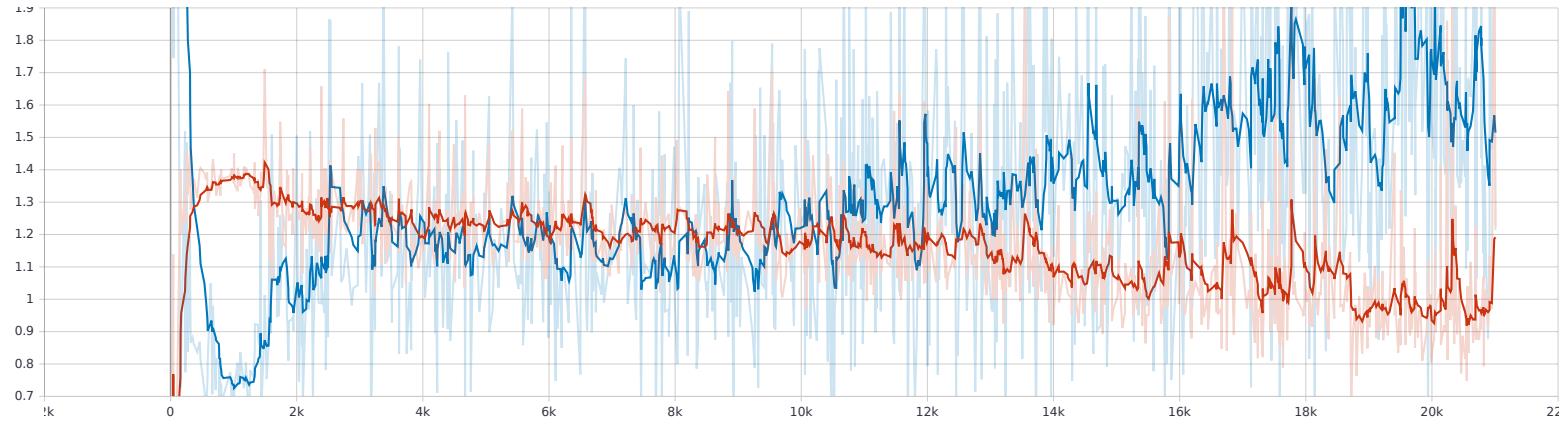
Slika 2: Funkcija greške DCGAN modela. Generator i Diskriminator.



Slika 3: Funkcija greške SN(1)-DCGAN modela. **Generator** i **Diskriminator**.



Slika 4: Funkcija greške SN(2)-DCGAN modela. **Generator** i **Diskriminator**.



Slika 5: Funkcija greške SN(3)-DCGAN modela. **Generator** i **Diskriminator**.

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. *arXiv 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.