# Generative Adversarial Network

# s využitím konvolučnej neuréonovej siete

#### Ladislav Kováč

Katedra kybernetiky a umelej inteligencie Technická univerzita v Košiciach Košice,Slovensko ladislav,kovac@student,tuke.sk

Abstract—V tejto eseji by som chcel popísať umelú neurónovú sieť Generative Adversarial Network.

V úvode predstavím základný princíp a najcitovanejšie články. V druhej sekcii bližšie popíšem aplikácie a rôzne metódy reprezentácie modelov. V tretej časti článku popíšem topológiu a detailnejšie modely ako aj učenie. V nasledujúcej časti popíšem hyperparametre a parametre neurónovej siete z časti tri. V piatej časti popíšem L funkciu celého systému.

Keywords—deep learning; generative adversial network; convolutial neural network; computer vision

## I. ÚVOD

Za myšlienkou Generative Adversarial Network (GAN) stojí Ian Goodfellow, ktorý tento framework predstavil vo svojej práci pre Open AI [1]. Umelé neurónové siete sú dobre v predikcii, klasifikácií, no vedcov zaujímalo či sa vie nejaký typ umelej neurónovej siete vytvoriť "sama zo seba". Intuícia GAN je v učení neurónovej sieti zo známych objektov a z nich následne generovanie nových objektov.

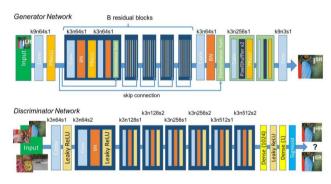
Tento framework sa skladá z dvoch systémov. A to systému, ktorý generuje objekty od nuly (Generative) - Generator, a systému, ktorý rozhoduje o tom či je vytvorený nový objekt podobný známemu objektu (Adversarial) - Discriminator. Tieto systémy kooperujú spolu počas celého procesu trénovania. Pri stabilnom procese trénovania by mal systém, ktorý generuje objekty, byť schopný pri každej epoche vytvárať objekty podobnejšie už známym.

Ďalším míľnikom je vo vývoji a výskume GAN je typ konvolučnej neurónovej siete, ktorá sa nazýva deep convolutional generative adversial networks (DCGANs)[2].Ako prvým sa podarilo natrénovať tento systém s použitím konvolučnej neurónovej siete, ktorá je učená nekontrolovaním učením. V využitím obrázkov na vstupe. Kde dokázal, že konvolučná neurónová sieť sa nemusí učiť iba kontrolovaným učením.

# II. ŠPECIFIKÁCIA OBLASTI APLIKÁCIE A POPIS PROBLÉMU

Primárne uplatnenie GAN našlo v počítačovom videní a počítačovej grafike. Pomocou týchto sieti vieme rekonštruovať digitálne obrázky zo šumu, zlepšovať kvalitu obrázku.

Jednou zo sľubných oblastí kde majú GA siete dobré výsledky je zlepšovanie kvality obrázku pri ich viacnásobnom zväčšení. Takáto hlboká neurónová sieť sa nazýva SRGAN, super-resolution GAN [3]. Jej topológia je veľmi rozsiahla a náročná. Topológia Discriminator-a obsahuje viacero konvolučných vrstiev a aktivačných funkcii Leaky ReLU. Generátor je taktiež konvolučná neurónová sieť ktorá, má na vstupe obrázok malej kvality a na výstupe obrázok super-resolved.

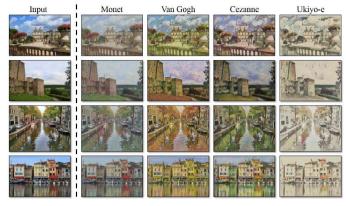


Obrázok 1 Podrovná topológia sytému SRGAN [3].

Výsledký v niektorých ukazoteloch neboli na úrovní SRResNet-u, state-of-the-art v tejto oblasti, ale v hlavnom ukazovatelovi, mean optimal error, bola najlepšia z pomedzi všetkých sieti aj konvenčných metód.

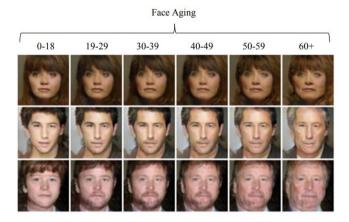
Pomocou GAN vieme upravovať obrázok na rôzne umelecké štýly (CycleGan) [4]. Topológia CycleGAN-u je o niečo zložitejšia. Celá štruktúra obsahuje dva systémy GAN a každý z nich obsahuje dva modely Generator-a (G a F) a jeden Discriminator-a (D).

Na začiatku sa rekonštruuje pôvodný obrázok pomocou G na obrázok nejakého stýlu napríklad Van Gogh-a. Následne sa vygenerovaný obrázok porovna v D s reálnými obrázkami Van Gogh-a. Kedže chceme použiť rekonštruovaný Van Gogh-ov štýl na náš pôvodný obrázok tak sa pridáva ešte jeden generator (F), kde na jeho výstupe je pôvodný obrázok už so štýlom Van Gogh-a. V druhom systéme sú navzájom vymenené generátory.



Obrázok 2 Výstupy CycleGAN s použitím rôznych vstupov a štýlov [4].

Ďalšou zaujímavou aplikáciou je Face Aging [5]. Kde GAN vytvára sekvenciu obrázkov ľudských tvári a predpovedá ako budú vyzerať v ďalších rokov života.



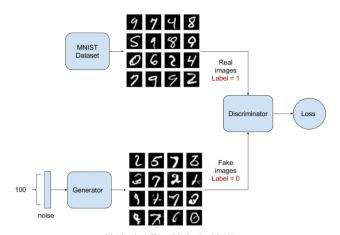
Obrázok 3Výstup Face Aging-u [5].

Jednou z prvých aplikácii s použitím textu je syntetizácia obrázku z textu [6]. Vedcom sa podarilo vytvoriť pomocou GAN obrázok s rozlýšením 256x256 pixelov len z textového popisu. Pričom je to lepší výsledok ako state-of-the-art metódy, ktoré generujú obrázky s polovičným rozlíšenim.

GAN sa môže aplikovať taktiež na rekonštrukciu tváre (TP GAN) z rôznych uhlov pričom neurónová sieť pozná len jednu fotografiu tváre [7].

## III. TOPOLÓGIA A POPIS SIETE

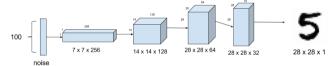
Tento systém sa skladá z dvoch hlavných podsystémov. A to z hlbokých sieti, ktoré sa nazývajú Generator a Discriminator.



Obrázok 4 Topológia GAN [8].

#### A. Generator

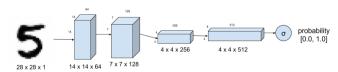
Model Generator je v princípe autoencoder, ktorý má na vstupe náhodné zašumené dáta vytvorené pomocou neakej distribúcie. Napríklad jednou z možností je pomocou normálneho rozdelenia vytvoriť zašumený vektor hodnôt, ktorá bude reprezentovať vstup. A na výstupe sa bude snažíť rekonštruovať digitálny obraz, pomocou upsampling-u. Tento model môže vyzerať takto:



Obrázok 5 Model Generator-a [8].

# B. Discriminator

Druhým podsystémom je Discriminator. Discriminator je klasická Konvolučná neurónová sieť, kde vstup reprezentuje digitálny obraz a výstup je apriorna pravdepodobnosť aký je obraz "real" resp. "fake". V poslednej vrstve tak figuruje sigmoidálna aktivačná funkcia, ktorej výstup reprezentuje "real" obraz a to hodnotou 1, resp. "fake" = 0. Takže úlohou tejto siete je vedieť rozpoznať či vstup je reálny z datasetu alebo vytvorený pomocou modelu Generator.



Obrázok 6 Konvolučná neurónová sieť, ktorá reprezentuje model Discriminator-a [8].

Najzaujímavejšou časťou tohto systému je samotný spôsob trénovania.

Na začiatok začneme pomocou malej vzorky z trénovacích dát (mini-batch) trénovať Discriminator.Kde na vstup privedieme obrazky z datasetu a na výstupe očakávame hodnotu, ktorú nazveme *real loss*. Následne necháme vygenerovať vstup Generator-a a na výstupe očakávamé obrázok. Tento obrázok nastavíme na vstup nášho Discriminator-a a očakávame pravdepodobnosť *fake loss*. V princípe pri jednej epoche sa Generator spustí raz ale Discriminator dva-krát

Loss funkcia Discriminator-a predstavuje vzťah:

# real loss - fake loss

Čo predstavuje ako odlišné sú si vygenerované dáta voči reálnym z datasetu. Tieto hodnoty sú znovu vrátane do svojich pôvodných modelov, v ktorých sa následne upravujú parametre podľa nich.

Týmto spôsobom sa snaží Generator oklamať Discriminator, že aktuálny vstup sú realne obrázky z datasetu.

Ak sa parametre nastavujú správne tak hodnota fake loss má narastať a real loss zase klesať. Pokiaľ je real loss rovná 0.5. Čoho dôsledkom je to, že generátor je dokázal naučiť generovať podobné dáta ako na nachádzajú v datasete. Je nutné podotknúť, že každý model má vlastnú Loss funkciu.

#### IV. HYPERPARAMETRE A PARAMETRE

# A. Hyperparametre GAN

Neurónová sieť GAN je veľmi náročná na stabilitu trénovania.

Parametre učenia jednotlyvých systémov sú veľmi malé. Rádovo sú to hodnoty 1x10<sup>-6</sup> a menšie.

Kvôli penalizácii velkých synaptických váh sa v tomto prípade môže použiť weight decay.

Váhy sa menia po 64 vzorkách – bath size. Vstupný vektor (šum) do Generátora má veľkosť 1x100.

Počet epoch je na pomery klasických neurónových sieti aj konvolučných hlbokých sieti veľmi veľký. A to v rádovo niekoľko desiatok tisíc.

## B. Parametre Discriminator-a

Vstupny obraz 28x28x1

Filter 1 5x5x64, stride 1 – 64 príznakových platní Filter 2 5x5x126, stride 2 – 128 príznakových platní Filter 3 5x5x256, stride 2 – 256 príznakových platní Filter 4 5x5x512, stride 1 – 512 príznakových platní

# C. Parametre Generator-a

Filter 1 5x5x128, padding Filter 2 5x5x64, padding Filter 3 5x5x32, padding Filter 4 5x5x1, padding

## V. VÝBER L FUNKCIE A JEJ MATEMATICKÝ POPIS

Kedže každý systém má vlastnu L funkciu, ktorá môže byť rozna, tak som sa rozhodol popísať L funkciu celého systému, ktorá sa nazýva teoretická loss funkcia. Je to vlastne minimax pravidlo, ktoré sa používa v teórii hier.

Teoretická loss funkcia:

$$E(G, D) = \frac{1}{2} \mathbb{E}_{x \sim p_t} [1 - D(x)] + \frac{1}{2} \mathbb{E}_{z \sim p_z} [D(G(z))]$$
$$= \frac{1}{2} (\mathbb{E}_{x \sim p_t} [1 - D(x)] + \mathbb{E}_{x \sim p_g} [D(x)])$$

z – zašumený vektor.

G(z) – výstup generátor-a.

x – trénovacia vzorka.

D(x) – výstup discriminator-a pre reálnu trénovaciu vzorku.

D(G(z)) – výstup discriminator-a pre vstup od generátor-a.

Optimalizačním kritériom je nájsť také parametre sme minimalizovali túto funkciu podľa G a maximalizovali podľa D

Pre Discriminator platí:

D(x) – má byť maximálne.

D(G(z)) – má byť minimálne.

Pre Generator platí:

D(G(z)) – má byž maximálne.

#### REFERENCES

- GoodFellow, I. (2014). Generative Adversarial Nets. Available at: https://arxiv.org/pdf/1406.2661.pdf [Accessed 12 May 2019].
- [2] Radford, A., Metz, L. and Chintala, S. (2016). Unsupervised representation learning with deep convolutional generative adversarial networks. *ICLR* 2016. [online] Available at: https://arxiv.org/pdf/1511.06434.pdf [Accessed 12 May 2019].
- [3] Ledig, C., Theis, L., Huszar, F. and Caballero, J. (2017). Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. [online] Available at: https://arxiv.org/pdf/1609.04802.pdf [Accessed 12 May 2019].
- [4] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks", in IEEE International Conference on Computer Vision (ICCV), 2017.
- [5] Antipov, G., Baccauche, M. and Degelay, J. (2018). Face Aging with Identity-Preserved Conditional Generative Adversarial Networks. 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
- [6] Zhang, H., Xu, T., Li, H., Zhang, S., Huang, X., Wang, X. and Metaxas, D. (2016). StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks. [online] Available at: https://arxiv.org/pdf/1612.03242v1.pdf [Accessed 12 May 2019].
- [7] Huang, R., Zhang, S., Li, T. and He, R. (2017). Beyond Face Rotation: Global and Local Perception GAN for Photorealistic and Identity Preserving Frontal View Synthesis. [online] Available at: https://arxiv.org/pdf/1704.04086.pdf [Accessed 12 May 2019].
- [8] Towards Data Science. (2019). GAN by Example using Keras on Tensorflow Backend. [online] Available at: https://towardsdatascience.com/gan-by-example-using-keras-ontensorflow-backend-1a6d515a60d0 [Accessed 4 May 2019].