

APLICAREA REȚELEI NEURONALE DE TIP DCGAN PENTRU GENERAREA LOGO-URILOR

Roman SOCHIRĂ, student, Facultatea de Științe Reale, Economice
și ale Mediului, Universitatea de Stat „Alecă Russo” din Bălți
Conducător științific: **Adela GOREA**, asist. univ.

Abstract: *This article examines the applicability of neural networks, particularly the DCGAN architecture, to logos generation. The model was trained on a dataset consisting of 22000 logo images giving a good result for the hardware capabilities provided for free by Google Colab. Essentially by increasing the training time and increasing the complexity of the model can achieve very good results.*

Keywords: *neural networks, DCGAN network, logos generating, tensorflow, google colab.*

În prezent, domeniul rețelelor neuronale este unul foarte activ și promițător fiind doar la începutul expansiunii sale. Rețelele neuronale sunt folosite în multe domenii din viața noastră de zi cu zi, acestea sunt diverse, începând cu recunoașterea fețelor omenești, pilotarea automată a mașinilor, terminând cu generarea de imagini și videoclipuri.

Rețelele neuronale artificiale de tip GAN (generativ-competitive) – sunt utilizate cu succes la generarea conținutului unic cum ar fi: generarea imaginilor, videoclipurilor, textului, și sunetului, ele au apărut relativ nu demult fiind introduse pentru prima dată de Ian Goodfellow în articolul său [6, p. 23].

Rețelele de tip GAN sunt o prescurtare din engleză pentru „Generative Adversarial Network”, care sunt un tip de rețea neuronală artificială formată din 2 părți un generator și un discriminator. Generatorul primește o intrare aleatoare și încearcă să genereze un rezultat care să semene cu datele de antrenament. Discriminatorul la rândul său primește atât date reale de antrenament cât și date generate de generator și încearcă să distingă între ele care este reală și care este generată [3, p. 1].

Familia rețelelor neuronale de tip GAN se poate bucura de numeroase arhitecturi care au drept scop îmbunătățirea rezultatelor și optimizarea antrenării. Printre acestea se numără și arhitectura DCGAN care oferă o abordare promițătoare pentru modelarea generativă a datelor de imagine și au arătat rezultate impresionante pe o varietate de seturi de date [2, p. 2].

1. Arhitectura DCGAN

În ultima perioadă au apărut numeroase variații ale rețelelor GAN printre acestea se numără GAN-uri condiționate care generează rezultate pe baza unei anumite condiții, GAN-uri Wasserstein care utilizează o funcție de pierdere diferită pentru a îmbunătăți stabilitatea în timpul antrenării, CycleGAN-urile, StarGAN-urile precum și altele [1, p. 7].

Pentru crearea rețelei neuronale de tip GAN a fost aleasă ca model arhitectura DCGAN.

Arhitectura DCGAN din engleză „Deep Convolutional GAN” (rețele generativ-competitive convoluționale adânci) a fost introdusă pentru prima oară în anul 2015 de către Alec Radford și oferea o stabilitate mai mare la antrenare și generare de imagini pentru rețelele de tip GAN [3, p. 2].

Principalele diferențe dintre un GAN standard și un DCGAN este utilizarea straturilor convoluționale în locul straturilor complet conectate și utilizarea normalizării loturilor atât în generator cât și în discriminator [4. p. 9].

Modelul creat este format dintr-un generator și un discriminator respectiv. Scopul generatorului este să genereze, pe bază unei intrări, o imagine și este format din următoarele straturi funcționale:

- *Intrarea* – este stratul care primește un vector din 100 valori, în intervalul 1 și -1, generate aleatoriu;
- *Dense* – este stratul care transformă valorile vectorului într-o reprezentare 3D numit tensor;
- *Conv2DTranspose* – sunt straturile care măresc rezoluția;
- *Ieșirea* – este ultimul strat care transformă tensor-ul rezultat într-un tensor de forma 32x32x3 care în continuare poate fi interpretat drept o imagine cu rezoluția 32x32 și 3 culori RGB.

Scopul discriminatorului este de a prezice pe baza unei imagini de intrare dacă aceasta este reală sau generată și este format din următoarele straturi funcționale:

- *Intrarea* – este stratul care primește drept intrare o imagine cu rezoluția 32x32 pixeli;
- *Conv2D* – sunt straturile care micșorează rezoluția;
- *Flatten* – este stratul care reduce dimensiunea tensorului din 3D în 1D;
- *Ieșirea* – este stratul care face predicția dacă imaginea de la intrare este reală sau generată (1 sau 0).

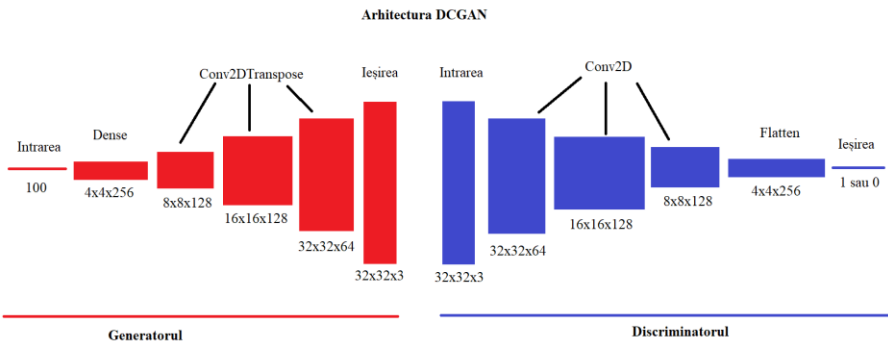


Fig. 1. Arhitectura rețelei neuronale DCGAN

Modelul rețelei neuronale din figura 1 a fost definit cu ajutorul limbajului de programare Python și folosind bibliotecile sale pentru învățarea automată Tensorflow și Keras. Ca mediu de dezvoltare s-a folosit Google Colab. În total modelul dat conține 1.5×10^7 parametri antrenabili pentru generator și 5×10^5 de parametri antrenabili pentru discriminator.

2. Pregătirea setului de date pentru antrenare

Pentru antrenarea modelului s-a folosit un set de date format din 22000 de imagini adunate din seturi de date mai mici din acces liber și verificate manual. Ulterior imaginile au fost prelucrate și standartizate în formatul PNG cu rezoluția 32x32 pixeli fiecare folosind pentru aceasta biblioteca limbajului Python PIL. Imaginile prealabil au fost sortate în mod aleatoriu folosind funcția Python random. Pentru a putea fi folosite la antrenare imaginile au fost transformate într-un tensor 4D cu dimensiunea 22000x32x32x3 folosind pentru aceasta biblioteca Numpy. La fel valorile pixelilor au fost normalizate în intervalul [1, -1] pentru a optimiza antrenarea. În figura 2, pentru reprezentare, au fost alese aleatoriu 100 de imagini din setul inițial.



Fig. 2. Exemple de logo-uri din setul de date

Setul de date colectat reprezintă imagini cu logo-uri reale care sunt sau au fost folosite de către diferite entități. Această colecție de logo-uri a fost creată special pentru acest model de rețea neuronală și a fost folosită pentru antrenarea discriminatorului.

3. Antrenarea modelului

Modelul a fost antrenat timp de 2 ore folosind procesorul grafic (GPU) Tesla T4 oferită gratuit de mediul cloud de dezvoltare Google Colab. Ca funcție de activare pentru neuronii straturilor adânci s-a folosit funcția *LeakyReLU*. Pentru ieșirea generatorului s-a aplicat funcția de activare *tanh* iar la discrimi-

nator funcția *sigmoid*. Ca funcție de pierdere pentru model s-a folosit entropia încrucișată binară (în engleză „binary crossentropy”). Ca optimizator s-a luat *Adam* cu rata de învățare 2×10^{-4} și metrica acuratețe (în engleză „accuracy”).

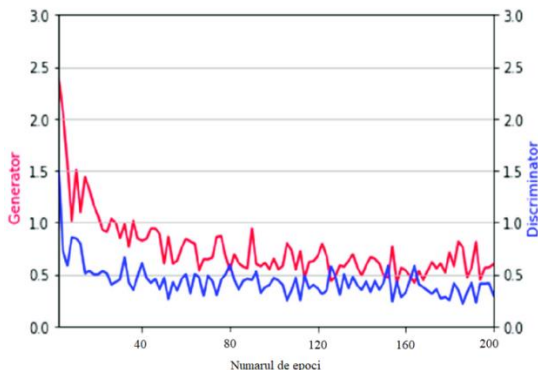


Fig. 3. *Curba de pierdere*

Modelul a fost antrenat timp de 200 de epoci ceea ce se vede în figura 3, logo-urile reale și cele generate au fost împărțite egal în loturi a câte 150 de imagini. În fiecare epocă modelul a parcurs câte 150 imagini, în total fiind parcurse 30000 de imagini dintre care 15000 imagini reale și 15000 imagini generate de generator. La final de antrenare, acuratețea de precizie pentru imaginile reale a reprezentat 99% pe când pentru cele generate 75%.

4. Rezultatele

După antrenarea timp de 200 de epoci, a fost salvat generatorul folosind funcția standard de salvare. Mărimea totală a fișierului în care se păstrează parametrii antrenați ai generatorului a fost de 5.6 MB. Pentru generarea logo-urilor s-a creat o funcție Python care primește numărul de exemple dorite împreună cu generatorul și întoarce imaginile cu rezoluția 32x32 pixeli în format PNG.

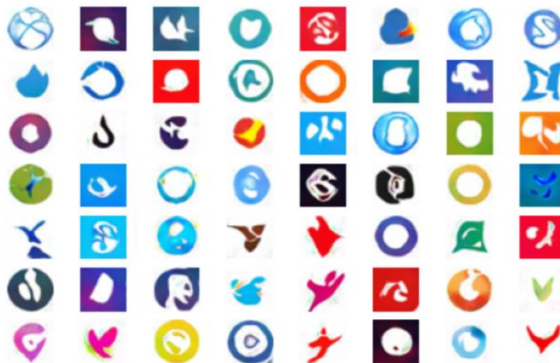


Fig. 4. *Exemple de logo-uri generate*

În figura 4, sunt reprezentate câteva exemple de logo-uri generate folosind generatorul antrenat. S-a observat că modelul generează logo-uri cu diverse figuri geometrice, la fel se remarcă și varietatea de culori care generate. Generarea unor astfel de logo-uri este destul de rapidă ceea ce se datorează mărimii mici a modelului, în total aproximativ 1.5×10^7 de parametri.

Concluzii:

Pe parcursul efectuării studiului, a fost creată o rețea neuronală de tip GAN, în mediul de dezvoltare Google Colaboratory folosind pentru aceasta limbajul Python și bibliotecile sale TensorFlow și Keras care generează imagini de logo-uri în formatul PNG cu rezoluția de 32x32 pixeli. Pentru atingerea acestui obiectiv, a fost efectuată analiza domeniului rețelelor neuronale de tip GAN și în special a arhitecturii DCGAN.

Rețele neuronale de tip GAN se descurcă destul de bine cu crearea de conținut nou și pot fi folosite la soluționarea problemelor legate de domeniul creației grafice, acestea elimină necesitatea elaborării manuale a logo-ului, acestea fiind generate în mod automat de model. Totodată ele pot fi folosite ca sursă de inspirație pentru designeri.

Articolul dat poate prezenta interes pentru cei ce sunt interesați de domeniile de aplicare a rețelelor creative de tip GAN, unul dintre acestea fiind generarea logo-urilor. În perspectivă aplicarea cercetată poate servi drept un punct de pornire spre îmbunătățirea ulterioară a acestui model pentru generare a logo-urilor.

Bibliografie:

1. ARJVOSKY, M., CHINTALA, S., BOTTOU, L. *Wasserstein GAN*. Articol științific. [online] [citată 23.03.2023]. Disponibil: <https://arxiv.org/pdf/1701.07875.pdf/>
2. DENTON, E., CHINTALA, S., SZLAM, A., FERGUS, R. *Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks*. Articol științific. [online] [citată 23.03.2023]. Disponibil: <https://arxiv.org/pdf/1506.05751.pdf/>
3. IAN J. Goodfellow. *Generative Adversarial Nets*. Articol științific. [online] [citată 23.03.2023]. Disponibil: <https://arxiv.org/pdf/1406.2661v1.pdf/>
4. KARRAS, T., AILA, T., LAINE, S., LEHTINEN, J. *Progressive Growing of GANs for Improved Quality, Stability, and Variation*. Articol științific. [online] [citată 26.03.2023]. Disponibil: <https://arxiv.org/pdf/1710.10196.pdf/>
5. RADFORD, A., METZ, L. *Unsupervised Representation Learning With Deep Convolutional Generative Adversarial Networks*. Articol științific. [online] [citată 26.03.2023]. Disponibil: <https://arxiv.org/pdf/1511.06434.pdf/>
6. YAU CHEONG, S. *Hands-On Image Generation With TensorFlow: A practical guide to generating images and videos using deep learning*. Sebastopol: O'Reilly Media, 2020. ISBN 978-183-88-2678-9