Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar

KÉPGENERÁLÁS DIFFÚZIÓS MODELLEKKEL

Mélytanulás projektmunka

Slepp csapat

Kányádi Richárd (EPI047)

Pataki Dávid (EWXZA3)

Tasi Zsombor (T0D8GA)

2024.12.13.

1 Feladat leírása

Image generation with diffusion models

Implement and train unconditional diffusion models, such as DDPM (Denoising Diffusion Probabilistic Model) or DDIM (Denoising Diffusion Implicit Model) for generating realistic images. Evaluate the capabilities of the models on two different datasets, such as CelebA and Flowers102.

Related GitHub repositories:

https://huggingface.co/blog/annotated-diffusion

https://github.com/huggingface/diffusers

https://keras.io/examples/generative/ddim/

Related papers:

https://arxiv.org/abs/2006.11239

https://arxiv.org/abs/2010.02502

2 Bevezető a témához

A feladatkiírásban említett két generatív modell (DDPM és DDIM) hasonló elveken működnek, viszont eltérő célokra helyezik a hangsúlyt és más optimalizációs stratégiával rendelkeznek. Mindkét modell diffúziós elvet követ, tehát fokozatosan a kép folyamatosan zajosodik, majd a betanított zajcsökkentési folyamat segítségével visszaállítja az eredeti adatra. A cél tehát iteratív módon képek generálása zajból. Főbb különbségek közé tartozik, hogy a diffúziós folyamatot érintő eltérés, a DDPM stochasztikus, míg a DDIM determinisztikus megközelítést alkalmaz. A DDPM esetén a kép visszaállításának folyamata sok lépésből áll, míg a DDIM jelentősen kevesebből, ezért az utóbbi lényegesen gyorsabb generálási sebességet biztosít és kevesebb számítási erőforrást igényel. A lassabb számításért cserébe a DDPM által generált képek minősége kiemelkedő és a zajokkal szemben is robosztusabb.

A félév során az idő szűkössége és az erőforrások hiánya miatt csak a DDIM modellt tudtuk saját módon implementálni. Kipróbáltunk továbbá egy GAN (Generative Adversarial Networks) modellt is.

Az irodalmak feldolgozásához és a feladat megértéséhez, valamint a megoldáshoz fordítót és ChatGPT-t használtunk. [1] [2]

3 Modellek

3.1 GAN modell

A GAN (Generative Adversarial Networks) modell két neurális hálózatból áll, amik egymás ellen „versenyeznek” [3]. Ez a két háló a Generátor és Diszkriminátor. Az előbbi feladata új, a valósághoz hasonló minták generálni, hogy a Diszkriminátor ne tudja megkülönböztetni a valódiaktól. Bemenetként egy véletlenszerűen generált zajt adunk, majd a kimenet pedig már a generált adat, esetünkben kép lesz. Utóbbi feladata megkülönböztetni a generált képeket a valós képektől. Bemenete tehát egy valódi vagy generált kép, kimenete pedig egy valószínűségi érték, hogy a bemenet milyen valószínűséggel valódi. A veszteségfüggvény egy min-max optimalizáció eredményeképpen írható fel. A modell előnyei közé tartozik széles körben alkalmazható és valósághű eredményeket hoz, viszont a tanítás idő és számításigényes.

3.2 Keras DDIM modell

Ez a kód egy diffúziós modell segítségével történő képgenerálást mutat be. [4] A célja az, hogy mesterséges neurális hálózatokat használva generáljon új képeket, amelyek hasonlóak az Oxford Flowers 102 adathalmaz képeire. A kód több részre bontható, amelyek közül a legfontosabbak a diffusion model architektúrája és működése.

Először meghatározza a TensorFlow-t, mint Keras backend-et. Ez biztosítja, hogy a Keras API TensorFlow-t használjon a számításokhoz. Majd megadjuk a használandó paramétereket. Képeket feldolgozzuk és előkészítjük a datasetet. Az adathalmazt tanító és validációs szettekre osztja, 80%-20%-os arányban. A KID (Kernel Inception Distance) metrika azt méri, mennyire hasonlóak a generált képek a valós adatokhoz. A Diffusion Model valósítja meg a diffúziós folyamatot, mint például a denoise, a zaj csökkentése adott zajarány mellett, a reverse_diffusion, ami a visszafelé haladó zajmentesítés és a generate, amivel új képeket fogunk generálni. A modell a tanítása során a zajos képekből való visszaállítást tanulja meg, majd ezeket a KID metrikával kiértékeljük.

3.3 Egyszerűsített DDIM modell

A CelebA képeket és a partícionálási fájlt itt is az általunk megosztott Google Drive-ról töltjük le. A torchvision.transforms segítségével 64x64 méretűre átméretezzük, majd tensor-okká alakítjuk és normalizáljuk. A CelebADataset osztály az adatok partícionálása alapján kezeli a tréning, validációs és teszt adathalmazokat. Az erőforrás-korlátok miatt a teljes adathalmazból kisebb, véletlenszerű mintákat veszünk (10,000 tréning, 2,000 validációs és teszt adat) és DataLoader-ekkel töltjük be.

A modell előkészítéseképp lineárisan növekvő béta ütemezést használunk, amely az időlépéseknél alkalmazott zaj mértékét szabályozza, valamint előszámításokat végzünk a diffúziós folyamat gyorsítására. Az egyszerűsített modell miatt az alkalmazunk lineáris béta ütemezés, míg a DDIM [5] (és számos továbbfejlesztett DDPM [6]) általában cosine ütemezést használ, amely jobb képminőséget eredményez. [7] Különböző hiperparaméterek is definiálunk.

A modellunk egy UNet architektúrán alapul, ami egy encoder-decoder alapú hálózat, konvolúciós és transzponált konvolúciós rétegeket tartalmaz. Az „időlépések” információját egy lineáris beágyazással integrálja a bemenetekhez.

A veszteségfüggvény az MSE (Mean Squared Error), ami az adott időlépésben a zajt próbálja prediktálni. Epochonként mentjük a modell aktuális állapotát, ezáltal checkpointokat alkalmazunk.

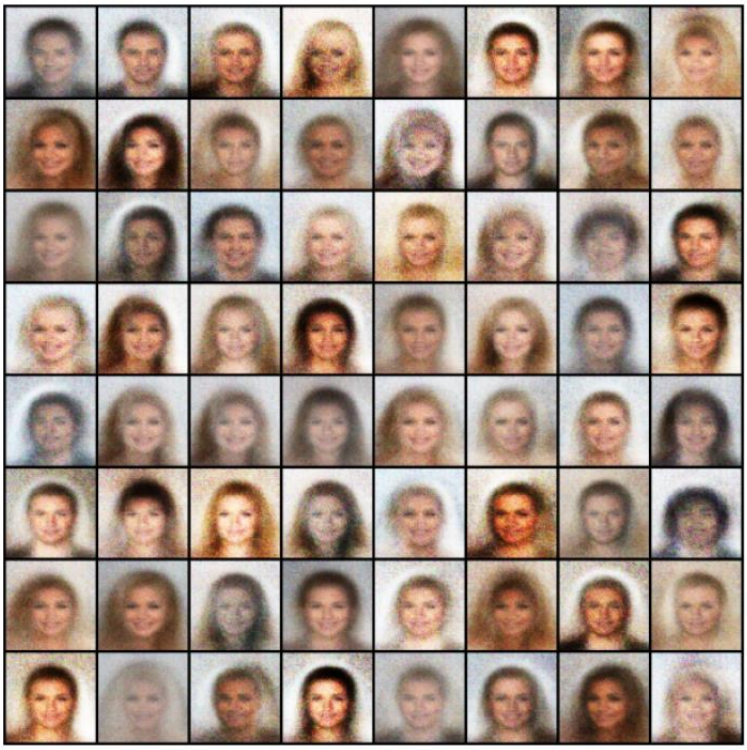
A notebook végén a diffúziós modell által generált képek kerülnek vizualizálásra, melyek a tanított UNet teljesítményét tükrözik a CelebA adathalmazon.

Összességében ez az implementáció egy egyszerű diffúziós modell, amely bár közel áll a DDIM működéséhez, de nem valósítja meg annak teljes funkcionalitását.

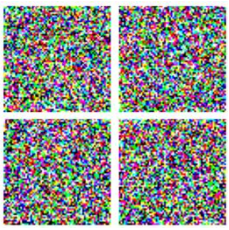
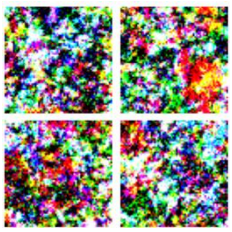
4 Eredmények

4.1 GAN modell

Csökkentettük a tanító adathalmaz méretét (10000 minta), hogy csökkentsük a modell tréningeléséhez szükséges időt. A hálót 5 epochon keresztül futtatuk, ami az alábbi arcok generálását eredményezte.



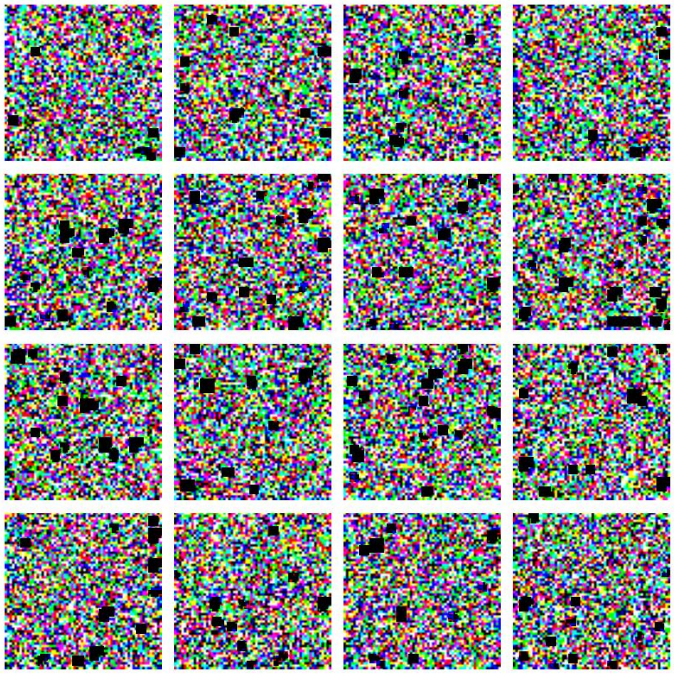
4.2 Keras DDIM modell

				
Epoch 1 után	**Epoch 5 után**	**Epoch 10 után**	**Epoch 15 után**	**Epoch 20 után**

A tanítást összesem 20 epochon keresztül futtattuk. Így változnak a train során a generált képek minőségei.

4.3 Egyszerűsített DDIM modell

Sajnos a modellünk nem lett elég komplex és az időhiány miatt a projekt leadási határidejéig nem tudtuk eleget tanítani, így az eredményünk ennél a modellnél nem túl fényes.



5 Irodalomjegyzék

[1] „DeepL Translate: The world’s most accurate translator”. Elérés: 2024. november 29. [Online]. Elérhető: https://www.deepl.com/translator

[2] „ChatGPT”. Elérés: 2024. december 13. [Online]. Elérhető: https://chatgpt.com

[3] I. Goodfellow *és mtsai.*, „Generative adversarial networks”, *Commun. ACM*, köt. 63, sz. 11, o. 139–144, okt. 2020, doi: 10.1145/3422622.

[4] K. Team, „Keras documentation: Denoising Diffusion Implicit Models”. Elérés: 2024. december 13. [Online]. Elérhető: https://keras.io/examples/generative/ddim/

[5] J. Song, C. Meng, és S. Ermon, „Denoising Diffusion Implicit Models”, 2022. október 5., *arXiv*: arXiv:2010.02502. doi: 10.48550/arXiv.2010.02502.

[6] J. Ho, A. Jain, és P. Abbeel, „Denoising Diffusion Probabilistic Models”, 2020. december 16., *arXiv*: arXiv:2006.11239. doi: 10.48550/arXiv.2006.11239.

[7] „The Annotated Diffusion Model”. Elérés: 2024. december 13. [Online]. Elérhető: https://huggingface.co/blog/annotated-diffusion