Mélytanulás házi feladat

Dokumentáció

Képgeneráció diffúziós modellek használatával

Készítették:

Köpeczi-Bócz Gergely

Safár Gergő

Bevezetés

A mi projektfeladatunk egy diffúziós modell implementálása és tanítása volt azzal a céllal, hogy azt később képgenerálásra lehessen felhasználni.

A feladatkiírás szerint a megvalósításhoz olyan diffúziós modelleket volt célszerű felhasználnunk, mint a DDPM (Denoising Diffusion Probabilistic Model) vagy a DDIM (Denoising Diffusion Implicit Model). A kiírásban rögzített cél volt továbbá, hogy a generált képek realisztikus ábrázolást kell, hogy szemléltessenek.

A házi feladat készítése során több különböző tudományos írást, cikket és publikációt megvizsgáltunk és felhasználtunk projektünk megvalósításához. Ezek egy része a kiírásban lévő ajánlott irodalom részét képezte azonban akad közötte általunk talált forrás is. Ezek a Hivatkozások fejezetben találhatóak felsorolva.

Adatelőkészítés

Adathalmazok kiválasztása

Az első lépés a házi elkészítése során két megfelelő adathalmaz kiválasztása volt. Több ilyen adathalmaz megvizsgálása után arra jutottunk, hogy a Flowers102, valamint a OxfordIIITPet adathalmazokat fogjuk felhasználni a modelleken végzet tanítás során.

A kettő adatstruktúra közül a Flowers102 elég kicsi méretűnek mondható, viszont ez az adatset szerepelt a feladatkiírásban, mint ajánlott halmaz. Ezért is esett rá a választás.

Emellett szerettünk volna egy nagyobb adathalmazon is elvégezni a feladatot, ezzel is tesztelve, hogy milyen különbség mutatkozik az adathalmazok méreteinek kisebb mértékű változása esetén. Ezen okból esett a választásunk végül az OxfordIIITPet nevű adathalmazra második opció gyanánt.

Míg a Flowers102 adathalmaz 1020 mintából dolgozik, addig az OxfordIIITPet esetében ez a szám 3680. Ezt a nagyjából 3-szoros különbséget kellően nagynak éreztük ahhoz, hogy érdemes legyen a két adathalmazt egymással összevetni, hogy a projekt során melyikkel érhetünk el jobb eredményeket.

Mindkét adathalmaz megtalálható volt a torchvision library-ben, ami nagyban megkönnyítette az importálásukat a projektünkbe.

Az OxfordIIITPet adatstruktúra esetében cleansing-et is alkalmaztunk, ugyanis előfordultak olyan osztályok a halmazban, melyek elemszáma nem egész 100 volt. Összesen 5 osztály volt ilyen a 36-ból és bár ezek elemszáma sem maradt el sokkal 100-tól (nagyjából 93 és 98 között mozogtak), mégis úgy gondoltuk, hogy az eldobásukkal tisztábbá, ugyanakkor pedig nem számottevően kissebbé fog megváltozni az adathalmazunk.

Adathalmazok felosztása

Következőnek felosztottuk mindkét adathalmazt három különböző részre. Ezek voltak a tanító, validációs és teszt halmazok, amelyek az eredeti adatstruktúrák 70, 15, valamint 15 %-át tették ki.

Mivel a két adathalmaz mérete elégségesen kicsi (kevesebb, mint 10 000 sample) volt a memóriában való tárolásukhoz ezért azokat a program során oda töltöttük be és tároltuk.

Fontos megjegyezni, hogy az mindkét adathalmazban 256x256-ostól 1024x1024-es felbontásig voltak minták, amiket végül minden esetben32x32-es felbontásra csökkentettünk le. Erre a projekt második felében használt DDPM miatt volt szükség. Ugyanis ebben az esetben egy előre elkészített modellt finomítottunk tovább, aminek feltétele volt, hogy 32x32-es képekkel kellett dolgoznnunk.

Felhasznált modellek

Ezzel át is térnénk a használt modellek ismertetésére. Kezdésnek, azaz baseline modellnek mi az úgynevezett VAE-t (Variational Autoencoder) használtuk.

Baseline modell

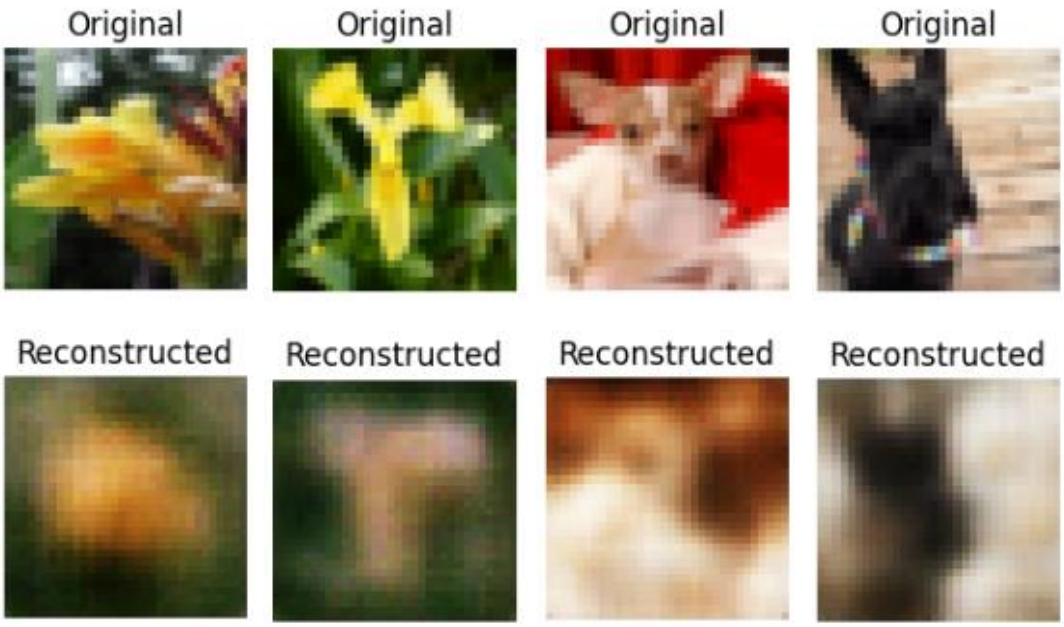
A VAE egy generatív neurális hálózati modell, amely a Bayes-statisztikán alapul, és hatékonyan képes tanulni az adatok rejtett, alacsony dimenziós reprezentációját. A VAE kiterjeszti a klasszikus autoencodert, amely az adatok kódolásával és visszafejtésével (dekódolásával) foglalkozik, azáltal, hogy a kódolt reprezentációt egy valószínűségi térben definiálja.

```python
# Encoder
self.enc = torch.nn.Sequential(
    torch.nn.Conv2d(3, 32, 4, 2, 1),
    torch.nn.ReLU(),
    torch.nn.Conv2d(32, 32, 4, 2, 1),
    torch.nn.ReLU(),
    torch.nn.Flatten()
)
```

```python
# Decoder
self.dec = torch.nn.Sequential(
    torch.nn.Linear(latent_dim, output_shape),
    torch.nn.ReLU(),
    torch.nn.Unflatten(1, (32, vmi, vmi)),
    torch.nn.ConvTranspose2d(32, 32, 4, 2, 1),
    torch.nn.ReLU(),
    torch.nn.ConvTranspose2d(32, 3, 4, 2, 1),
    torch.nn.Tanh()
)
```

1. ábra Az encoder és decoder felépítése a baseline (VAE) esetén

A modell megírása után következett a tanítási folyamat. Ezt először a Flowers102, majd az OxfordIIITPets adathalmazokon végeztük el. Végül plot-oltunk néhány képet az eredeti, valamint a VAE által rekonstruáltak közül, amelyek a következő képen láthatóak.



2. ábra A VAE által rekonstruált képek összehasonlítása az eredetiekkel mindkét adathalmaz esetében

Baseline kiértékelése

A generált képek pontosságának mérésére két közismert módszert is alkalmaztunk. Az egyik az úgynevezett Fréchet Inception Distance (FID), míg a másik az Inception Score (IS).

A FID egy széles körben használt metrika generatív modellek, többek között VAE-k által generált képek minőségének értékelésére. Az FID azokat a különbségeket méri, amelyek az eredeti (valós) és a generált (mesterséges) képek jellemzői között fennállnak, egy előre betanított neurális hálózat segítségével. A virágokat tartalmazó halmaz esetében a kapott pont

318.1361 lett, míg az állatokat tartalmazó képek esetében 305.5305. Sajnos általánosságban igaz, hogy 100-as érték fölött a generáció nem számít a legtökéletesebb minőségűnek, ami a fenti képeken is látszik.

Az IS (Inception Score) szintén a generált képek minőségének és sokféleségének mérésére szolgáló mérési eljárás, így ezt is alkalmaztunk mindkét adathalmaz esetében és rendre 2.36 valamint 2.58-as értékeket kaptunk. Általánosságban igaz, hogy ezek a nem kimagasló, de elfogadható generációkra jellemző értékek.

Advanced modell

Következőnek választottunk egy jobb képességűnek tartott modellt, amit kicsit továbbfejlesztettünk. A választásunk a huggingface DDPM modellje (1) lett. Pontosabban szólva a google/ddpm-cifar10-32 jelölésű, előre tanított modell. Ennek a tovább tanításához felhasználtuk a huggingface oldalán megtalálható segédletet. A továbbfejlesztésünk, azaz a hiperparaméter optimalizálásunk gyakorlatilag a learning rate finomhangolását jelentette.

Az átalakítások után a modellt két alkalommal tanítottuk, egyszer a virágokat és egyszer az állatokat tartalmazó adathalmazon. Az implementálása és futtatása során előforduló hibák esetében, valamint az adathalmazok tanításba való megfelelő beillesztésében a ChatGPT 4o által nyújtott segítséget is felhasználtunk.

A futtatás alkalmával az alább látható minta képeket sikerült generáltatni a továbbfejlesztett DDPM modellel.

3. ábra A DDPM által generált néhány mintakép az OxfordIIITPets esetében

4. ábra Szintén a DDPM által generált képek, azonban a Flowers102 esetében

Advanced kiértékelése

Természetesen a kiértékelésbe beletartozott ebben az esetben is, hogy mindkét adathalmaz esetében megnéztük a korábbi baseline esetében is megvizsgált FID és IS értékeket és ezek a következő képpen alakultak. Bár nem a várt mértékű volt a javulás, de egyértelműen jobban teljesített az újabb DDPM alapú modellünk.

Az evaluation folyamatát ebben az esetben úgy végeztük, hogy generáltattunk mindkét halmaz esetében száz-száz képet. Ezután random kiválasztottunk az eredeti adathalamazokból véletlenszerűen száz-száz képet. Végül a halmazonkénti 200 képen végeztük a FID kiértékelést. Ennek az eredménye az lett, hogy a virágokat tartalmazó halmaz esetében 318-ról 309-re, valamint az állatokat tartalmazó esetében 305-ről 256-ra sikerült lejjebb vinni a FID értékét. A második esetben ez 16%-os javulást jelent. Az OxfordIIITPets esetében vélhetően azért fedezhető fel nagyobb mértékű javulás mert ebben az esetben jóval nagyobb adathalmazon folyt a tanítás.

Ezek után elvégeztük az IS értékek kiszámítását is, ami egy sokkal érdekesebb eredményre vezetett. Ebben az esetben a virágok adathalmazán kiértékelve 2.35-ről 2.26-ra esett vissza az interception score. Tekintve, hogy a nagyobb IS érték élesebb képeket jelent, ebben az esetben a DDPM, bár minimálisan, de rosszabb eredményt ért el, mint a VAE. Ugyanez nem igaz az állatokat tartalmazó dataset esetében, ahol 2.579-ről 2.605-re sikerült az IS értéket növelni. Noha ez is kis mértékű javulásnak számít, mégis figyelembe véve az összes érték alakulását a DDPM hozott jobb eredményeket.

Összegzés

Összességében úgy véljük, hogy a DDPM modell mindkét esetben jobban teljesített (még annak ellenére is, hogy a flowers esetében a FID értéke nem az elvárt mértékbe tért ki a VAE-hez képes). A DDPM sokkal színesebb, valósághoz közelibb képeket generált, és az objektumok láthatóbbak és felismerhetőbbek az általa létrehozott képeken, mint voltak azok a VAE esetében. Véleményünk természetesen fakadhat abból is, hogy kicsit elfogultabbak vagyunk a komplexebb modell (DDPM) iránt, és úgy véljük, hogy ezek a metrikák nem alkalmazhatók olyan jól ehhez a modellhez, mint azt előzetesen gondoltuk.

Végeredményként úgy érezzük, hogy a kis siker is siker. Valamint, hogy a nem kevés munkamennyiség miatt, amit a projektbe fektettünk, elégedettek lehetünk az elért eredményekkel, már csak azért is, mert a feladat megoldása során rengetek hasznos tudással lettünk gazdagabbak.

Hivatkozások

1. DDPM tanításának huggingface dokumentációja, https://colab.research.google.com/github/huggingface/notebooks/blob/main/diffusers/training_example.ipynb#scrollTo=1f740dfe-e610-4479-ac30-cce1f9e62553
2. Huggingface cikk a diffúziós modellekről, https://huggingface.co/blog/annotated-diffusion
3. Huggingface git repo diffúziós modellekhez, https://github.com/huggingface/diffusers
4. Keras leírás a DDIM modellről, https://keras.io/examples/generative/ddim/
5. Cornell University publikáció DDPM modellekről, https://arxiv.org/abs/2006.11239
6. Cornell University publikáció DDIM modellekről, https://arxiv.org/abs/2010.02502