SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1154

VARIJACIJSKI AUTOENKODERI S KVANTIZIRANOM LATENTNOM REPREZENTACIJOM

Dominik Babić

Zagreb, lipanj 2023.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1154

VARIJACIJSKI AUTOENKODERI S KVANTIZIRANOM LATENTNOM REPREZENTACIJOM

Dominik Babić

Zagreb, lipanj 2023.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 10. ožujka 2023.

ZAVRŠNI ZADATAK br. 1154

Pristupnik:	**Dominik Babić (0036530161)**
Studij:	Elektrotehnika i informacijska tehnologija i Računarstvo
Modul:	Računarstvo
Mentor:	prof. dr. sc. Siniša Šegvić

Zadatak: **Varijacijski autoenkoderi s kvantiziranom latentnom reprezentacijom**

Opis zadatka:

Razumijevanje slika važan je zadatak računalnog vida s mnogim zanimljivim primjenama. U posljednje vrijeme posebno su zanimljivi generativni modeli jer omogućuju uređivanje postojećih i stvaranje novih podataka. Ovaj rad razmatra generativne modele zasnovane na konvolucijskim varijacijskim autoenkoderima s kvantiziranom latentnom reprezentacijom. U okviru rada, potrebno je odabrati okvir za automatsku diferencijaciju te upoznati biblioteke za rukovanje matricama i slikama. Proučiti i ukratko opisati postojeće arhitekture utemeljene na konvolucijama i pažnji. Odabrati javno dostupnu prednaučenu parametrizaciju kvantiziranog konvolucijskog autoenkodera i ocijeniti njenu prikladnost za uređivanje slika. Procijeniti mogućnost učenja modela na malenim slikama. Radu priložiti izvorni i izvršni kod razvijenih postupaka, ispitne slijedove i rezultate, kao i potrebna objašnjenja te dokumentaciju. Citirati korištenu literaturu i navesti dobivenu pomoć.

Rok za predaju rada: 9. lipnja 2023.

Zahvaljujem se prof. dr. sc. Siniši Šegviću na pruženoj pomoći, podršci i savjetima pri izradi rada.

SADRŽAJ

1. Uvod

Razumijevanje slika važan je zadatak računalnog vida s mnogim zanimljivim i korisnim primjenama. U posljednje vrijeme sve popularniji su generativni modeli jer omogućuju uređivanje postojećih i generiranje novih podataka. Neki od danas najpoznatijih generatora slika poput DALL-E 2, Midjourney, DreamStudio itd. zasnovani su na generativnom modeliranju.

Modeli strojnog učenja koji spadaju u područje "generativnog modeliranja" su *generativni modeli*, a njihova primjena je široka; od generiranja novih podataka, preko uređivanja postojećih, do popunjavanja nedostajućih podataka itd. Treniranje ovog tipa modela dugo je bio problem u zajednici strojnog učenja, a klasično su se pristupi suočavali s jednim od tri ozbiljna nedostatka. Prvo, mogli bi zahtijevati jake pretpostavke o strukturi podataka. Drugo, mogli bi koristiti ozbiljne aproksimacije, što dovodi do suboptimalnih modela. Ili treće, mogli bi se oslanjati na računalno skupe postupke zaključivanja poput Markovljevih lanaca Monte Carlo. Nedavna istraživanja napravila su ogroman napredak u treniranju neuronskih mreža kao moćnih aproksimatora funkcija putem propagacije pogreške unatrag. Ti napretci doveli su do obećavajućih pristupa za izgradnju generativnih modela.

Jedan od najproširenijih generativnih modela s raznim primjenama i izvan stvaranja podataka je *autoenkoder*, čija je glavna ideja sažimanje ulaznih podataka u podatke manjih dimenzija. Generativni aspekt autoenkodera se zatim postiže rekonstruiranjem tih sažetih podataka natrag u originalne podatke, po mogućnosti sa što manjim gubitkom kvalitete. Danas postoji mnogo različitih izvedbi autoenkodera, a jedni od najpopularnijih su *varijacijski autoenkoder* (VAE) [5], te varijacijski autoenkoder s kvantiziranom latentnom reprezentacijom (VQ-VAE) [10].

Cilj ovog rada je predstaviti koncepte i iz njih izvesti navedene potrebne modele, izvesti matematičku pozadinu takvih modela, prikazati njihovu primjenu, a potom i ispitati njihova svojstva.

2. Arhitekture autoenkoderskih modela

VQ-VAE (*eng.* Vector-Quantized Variational Autoencoder) je, kao što i sam naziv predstavljanja, vrsta autoenkodera, čija je osnovna ideja dekonstrukcija ulaznih podataka u latentan prostor manje dimenzije te rekonstrukcija ulaznih podataka iz skrivenog predstavljanja.

Kako bismo bolje razumjeli arhitekturu VQ-VAE modela, potrebno je prvo razumjeti osnovne koncepte latentnog prostora, autoenkodera i varijacijskih autoenkodera, a pritom i koncepte koji se koriste u njihovoj implementaciji.

Kroz ovo poglavlje uvest ćemo osnovne pojmove i koncepte potrebne za razumijevanje arhitekture VQ-VAE modela, a koji su potrebni za razumijevanje i za postizanje programske implementacije samog modela.

2.1. Svojstva latentnog prostora

Latentni prostor autoenkodera predstavlja niždimenzionalni prostor reprezentacija koji se koristi za komprimiranje i reprezentaciju ulaznih podataka. To je ključni dio autoenkodera koji sadrži informacije o bitnim značajkama podataka.

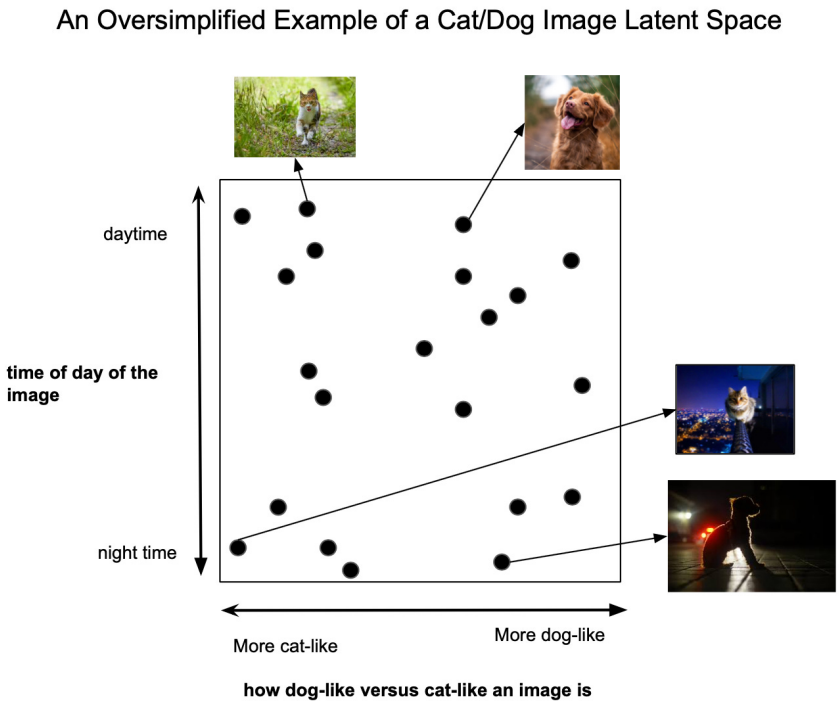
Latentni prostor je prostor u kojem se ulazni podaci transformiraju pomoću enkodera. Ova transformacija za cilj ima uhvatiti i komprimirati bitne informacije iz podataka, izostavljajući manje bitne ili beskorisne detalje. Također, jedna od ideja je preći iz koreliranih piksela u djelomično ili potpuno dekorelirane latentne aktivacije koje predstavljaju nezavisne faktore varijacije podataka[4]. Ovisno o arhitekturi autoenkodera, latentni prostor treba imati manji broj dimenzija u odnosu na izvorni prostor podataka.

Jedna od ključnih prednosti latentnog prostora autoenkodera je to što ona ima strukturu koja olakšava modeliranje i generiranje novih podataka. Latentni prostor omogućuje grupiranje sličnih podataka zajedno, što olakšava prepoznavanje i

generiranje novih podataka koji su slični ulaznim podacima. Na primjer, ako koristimo autoenkoder za modeliranje slika, latentni prostor može sadržavati reprezentacije različitih kategorija objekata, boja, tekstura i drugih karakteristika prisutnih u slikama.

Ovakav pristup sažimanja podataka i ekstrahiranje najbitnijih značajki je jednako onome koji je vidljiv i kod analize svojstvenih komponenata (PCA). No, za razliku od PCA, autoenkoderi uspijevaju dobro prikazati latentan prostor i kada postoji nelinearan odnos između podataka i njihovih latentnih reprezentacija ili kada su sami podaci suviše kompleksni. Jedan primjer takvog zadatka prikazan je slikom 2.1.

Prije nego li nastavimo dalje, važno je napomenuti razliku između *latentnog prostora* i *distribucije latentnog prostora* koji se koriste u generativnim modelima. Latentni prostor predstavlja sve moguće vektorske vrijednosti koje može poprimiti latentna varijabla, dok distribucija latentnog prostora predočuje koje vrijednosti će se češće pojavljivati, a koje rjeđe ukoliko uzorkujemo primjerke iz distribucije skupa za učenje.



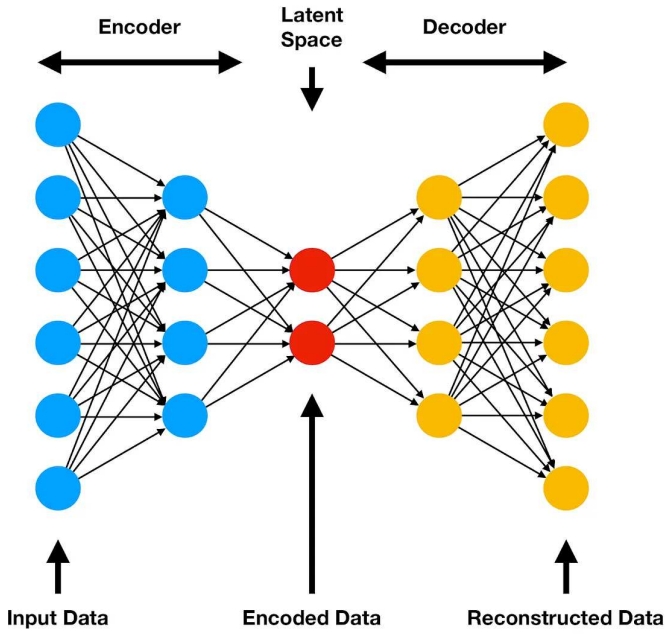
Slika 2.1: Pojednostavljeni prikaz latentnog prostora za slike mačaka i pasa. Nezavisne značajke slike u ovom slučaju su vrijeme u danu i sličnost psu odnosno sličnost mački. Preuzeto sa [9].

2.2. Autoenkoderi

Autoenkoderi su generativni modeli dubokog učenja koji nenadzirano uče efikasno reprezentirati podatake uz minimalnu potrebu za ljudskom intervencijom. Glavni elementi autoenkodera su:

- Enkoder $z = f(x)$
- Dekoder $x' = g(z)$

Pri čemu x predstavlja ulazne podatke, z latentnu vektorsku reprezentaciju ulaza, a x' je *rekonstrukcija* samog ulaznog podatka.



Slika 2.2: Arhitektura autoenkodera; plavi čvorovi predstavljaju neurone enkodera, crveni čvorovi predstavljaju neurone latentne reprezentacije, a narančasti čvorovi predstavljaju neurone dekodera. Preuzeto sa [9]

Enkoder je odgovoran za mapiranje ulaznih podataka u nižedimenzionalni latentni prostor. U procesu enkodiranja, ulazni podaci se smanjuju na svoju suštinu ili bitne značajke koje se zatim koriste za rekonstrukciju ulaznih podataka. Ove latentne reprezentacije su često manje dimenzionalne od ulaznih podataka i obično imaju strukturu koja olakšava modeliranje i generiranje novih podataka.

Dekoder, s druge strane, preuzima uzorke iz latentnog prostora i rekonstruira originalne podatke. Cilj je minimizirati razliku između rekonstruiranih podataka i originalnih podataka, čime se osigurava kvalitetna rekonstrukcija.

Autoenkoderi se često treniraju koristeći metodu rekonstrukcijske pogreške, koja mjeri koliko dobro se izvorni podaci mogu rekonstruirati iz latentnog prostora. Kroz iterativni postupak optimizacije, model se prilagođava kako bi minimizirao tu pogrešku i naučio reprezentacije koje su najbolje za rekonstrukciju podataka.

Kriterij rekonstrukcijske pogreške kod autoenkodera mogu biti ili srednja kvadratna pogreška ili negativna log-izglednost rekonstruiranog podatka koje su definirane kao:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2 \tag{2.1}$$

i

$$\text{NLL} = -\sum_{i=1}^{n}\log(p(x_i|z)) \tag{2.2}$$

Jedna zanimljivost oko logaritamske vjerojatnosti $\log p_\phi(x_i|z)$ je da se ona može prikazati i kao kvadratna razlika između ulaza x_i i izlaza dekodera x_i' pod uvjetom da je x_i aproksimiran iz normalne distribucije $\mathcal{N}(f(z), I)$:

$$\begin{aligned}\log p(x|z) &\approx \log\exp -(x - f(z))^2 \\ &\approx -(x - f(z))^2 \\ &= -(x - \hat{x})^2\end{aligned} \tag{2.3}$$

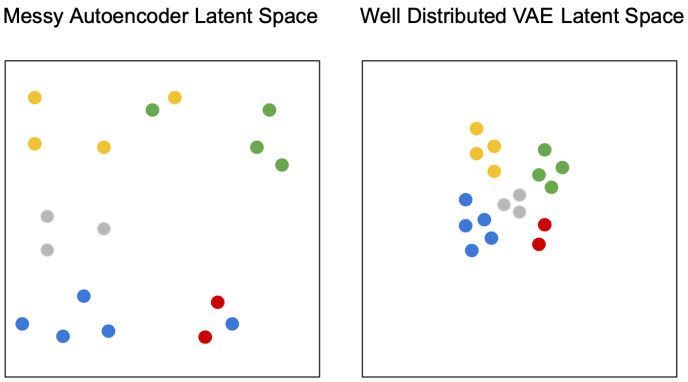
Dakle, možemo zaključiti da je MSE poseban slučaj poopćenog kriterija NLL. Inače, NLL se može primijeniti i na sigmoidalno aktivirane piksele koji se ravnaju po Bernoullijevoj distribuciji.

Jedna od ključnih ideja autoenkodera je učenje komprimirane reprezentacije podataka. Kroz ograničavanje kapaciteta latentnog prostora, model se potiče da nauči bitne značajke podataka i apstrahira beskorisne informacije. Ova komprimirana reprezentacija može se koristiti za razne zadatke kao što su klasifikacija, semantička segmentacija, uklanjanje šuma, rekonstrukcija ili generiranje novih podataka.

2.3. Varijacijski autoenkoderi

Idealno bismo željeli da naš latentni prostor grupira semantički slične podatke te da se semantički različiti podatci preslikavaju daleko jedan od drugih. Po mogućnosti, većina distribucije podataka trebala bi činiti relativno kompaktni volumen u latentnom prostoru, a ne protezati se u beskonačnost. Glavni problem s običnim autoenkoderima je što naučene latentne reprezentacije nemaju nužno navedene karakteristike. Model

može naučiti bilo koji latentni prostor koji želi, pa često završi s memoriranjem pojedinačnih podataka smještanjem u udaljene džepove latentnog prostora. Slika ispod vizualizira ove probleme s latentnim prostorima autoenkodera i uspoređuje ih s prostorima VAE modela.



Slika 2.3: Razlika u latentnim prostorima običnog autoenkodera i VAE modela. Na lijevoj slici prikazani su loše distribuirani latentni vektori običnog autoenkodera, dok su na desnoj slici prikazani kompaktno grupirani latentni vekotri VAE modela. Preuzeto sa [9].

Varijacijski autoenkoderi rješavaju taj problem uvodeći ograničenja na latentni prostor. Jedna od ključnih karakteristika VAE-a je učenje latentnog prostora putem varijacijskog zaključivanja. Umjesto da koristi točno zaključivanje, VAE koristi varijacijski pristup u kojem modelira distribuciju latentnih varijabli. To omogućuje generiranje novih podataka kroz uzorkovanje iz te distribucije. Ova probabilistička formulacija omogućuje fleksibilnost u generiranju različitih varijacija podataka.

Formalno, VAE modelira distribuciju latentnog prostora pomoću *apriorne latentne distribucije* $p(z)$ te *aposteriorne latentne distribucije* $p(z|x)$ distribucija. U većini VAE modela, to su normalne distribucije $N(0, I)$. Naš glavni cilj za postizanje generativnih svojstva je izraziti složenu gustoću $p(x) = \int p(x|z)p(z)$ pomoću jednostavnijih distribucija $p(x|z)$ i $p(z)$ kako bismo omogućili jednostavno uzorkovanje u dva koraka:

- uzorkovanje iz $p(z)$ kako bismo dobili latentni vektor z.
- uzorkovanje iz $p(x|z)$ kako bismo dobili rekonstrukciju x.

Osim generativnog svojstva VAE dekodera, potrebno izraziti i svojstvo enkodera da za dane podatke x_i daje latentne vektore z_i koji najbolje reprezentiraju latentni prostor. Dakle, enkoder treba modelirati *aposteriornu* distribuciju $p(z|x)$ iz koje ćemo potom uzorkovati latentne vektore koje ćemo koristiti kao ulaz u dekoder. Ovo svojstvo enkodera možemo izraziti pomoću Bayesove formule:

$$p(z|x) = \frac{p(x|z)p(z)}{p(x)} = \frac{p(x|z)p(z)}{\int p(x|z)p(z)dz} \tag{2.4}$$

Međutim, izračunavanje *aposteriorne* distribucije je teško, jer uključuje marginalizaciju po svim mogućim vrijednostima latentnih varijabli. Umjesto toga, koristimo varijacijsko zaključivanje čiji je cilj aproksimirati *aposteriornu* distribuciju pomoću jednostavnije distribucije $q(z|x)$. Ova distribucija se naziva *varijacijska distribucija* i obično je normalna distribucija $N(\mu, \sigma)$.

Učenje varijacijskih autoenkodera svodi se na optimiranje donje granice izglednosti $p(x)$. U praksi optimiramo dva člana. **Rekonstrukcijski gubitak** mjeri koliko dobro VAE rekonstruira ulazne podatke iz latentnog prostora. **Regularizacijski gubitak**, zasnovan na Kullback-Leibler (KL) divergenciji, mjeri razliku između pretpostavljene *apriorne* distribucije $p(z)$ i aproksimirane *aposteriorne* distribucije $p(z|x)$ te time pomaže u oblikovanju i kontroliranju latentnog prostora tako da se podaci grupiraju na smislene načine.

Prednosti VAE-a uključuju sposobnost generiranja novih podataka, interpolaciju između postojećih podataka i kontrolirano manipuliranje latentnim prostorom radi dobivanja željenih varijacija podataka. Također, VAE-i mogu služiti kao moćan alat za ekstrakciju značajki i smanjenje dimenzionalnosti podataka.

Ipak, VAE-i imaju svoje izazove. Interpretacija latentnog prostora može biti teška, a ponekad generirani podaci mogu biti manje kvalitetni u usporedbi s izvornim podacima. Veliki problem je u tome što $p(x|z)$ tipično pretpostavlja dekorelirane piksele uzorkovane iz Gaussove distribucije, što nikad nije slučaj. Također, odabir odgovarajuće arhitekture i parametara modela može biti izazovan zadatak.

Unatoč tim izazovima, VAE-i su postali popularni alati u području generativnog modeliranja i strojnog učenja. Njihova sposobnost generiranja novih podataka i fleksibilnost u oblikovanju latentnog prostora čini ih snažnim alatom za mnoge aplikacije, uključujući generativno umjetničko stvaralaštvo, rekonstrukciju i obnovu podataka, kao i stvaranje virtualnih svjetova u igrama i animaciji.

2.3.1. Varijacijsko zaključivanje

Varijacijsko zaključivanje je tehnika u strojnom učenju koja nam pomaže procijeniti skrivene varijable u modelima koji uključuju nesigurnosti. Kada radimo s probabilističkim modelima, često želimo saznati vrijednosti skrivenih varijabli na temelju dostupnih podataka. Varijacijsko zaključivanje nam omogućuje da procijenimo ove skrivene varijable tako što pretpostavljamo određenu jednostavniju

raspodjelu i pomoću optimizacije tražimo najbližu raspodjelu koja odgovara stvarnim podacima. Ova procjena nas vodi do boljeg razumijevanja i modeliranja podataka s nesigurnostima. [5]

2.4. Diskretne latentne reprezentacije

U realnom svijetu, dijelovi slika, zvukova i općenitih podataka često su povezani s diskretnim kategorijama. Na primjer, u slikama, neki dijelovi mogu odgovarati pojedinim kategoričkim objektima, a u zvuku, određeni dijelovi mogu odgovarati pojedinim fonemima. U takvim slučajevima, diskretna reprezentacija podataka može biti korisna za modeliranje podataka. Upravo takvi diskretni odnosi među dijelovima podataka su ideja i motivacija za VQ-VAE (*eng. Vector-Quantized VAE*) modele. [10].

Glavne smjernice za VQ-VAE modele su:

1. Diskretna reprezentacija - Umjesto kontinuiranog latentnog prostora koji koristi tradicionalni VAE, VQ-VAE je usredotočen na učenje diskretne reprezentacije latentnog prostora. Jednostavnije objašnjeno, umjesto da za odabir latentnog vektora imamo cijeli prostor realnih vektora, ograničit ćemo se na odabir vektora iz diskretne skupine vektora.

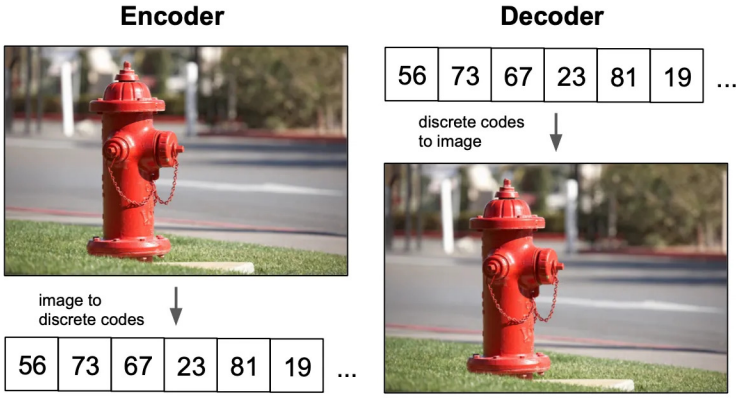
2. Vektorska kvantizacija - Ključni element VQ-VAE modela je tehnika nazvana vektor kvantizacija. Umjesto da svaki latentni vektor bude predstavljen kontinuiranom distribucijom, koristi se kvantizacija kako bi se svaki vektor mapirao na najbliži vektor iz diskretne skupine vektora. Ova tehnika omogućuje efikasno grupiranje sličnih latentnih vektora i bolje zadržavanje informacija u latentnom prostoru.

3. Gubitak kvantizacije - Da bi se potaknulo učenje diskretne reprezentacije, koristi se gubitak kvantizacije koji kažnjava razlike između originalnog latentnog vektora i najbližeg vektora iz diskretne skupine vektora. Ovaj gubitak potiče model da pravilno kvantizira latentne vektore i poboljšava sposobnost generiranja i rekonstrukcije podataka. Više o tome u poglavlju 3.4.3.

Diskretna reprezentacija latentnog prostora omogućuje bolje modeliranje podataka i olakšava primjenu diskretnih algoritama na latentni prostor. Ideja diskretnog latentnog prostora potječe od činjenice da mnogi podaci s kojima se susrećemo u stvarnom svijetu imaju tendenciju da se bolje predstavljaju diskretnom reprezentacijom. Na primjer, ljudski govor se dobro reprezentira putem diskretnih

fonema i jezika. Također, slike sadrže diskretne objekte s određenim diskretnim skupom svojstava. Moguće je zamisliti diskretnu varijablu koja predstavlja vrstu objekta, zatim varijablu za boju, veličinu, orijentaciju, oblik, teksturu, boju pozadine, teksturu pozadine itd.

Također, diskretni latentni prostor omogućuje primjenu diskretnih algoritama na latentni prostor kao što su, na primjer, algoritmi za grupiranje i klasifikaciju podataka u latentnom prostoru, te je moguće primijeniti diskretne algoritme za generiranje novih podataka što je posebno korisno u području generativnog umjetničkog stvaralaštva.



Slika 2.4: Prikaz diskretizacije latentne reprezentacije slike. Preuzeto sa [9]

Dakle, dvije temeljne razlike između VAE i VQ-VAE modela su:

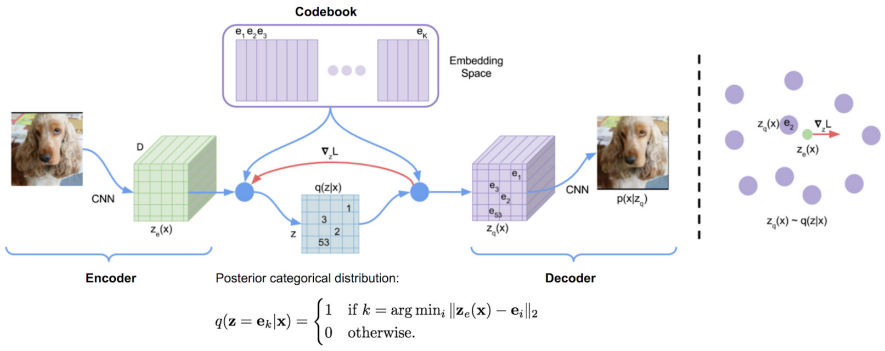
- Diskretni latentni prostor - dok VAE uči kontinuiranu latentnu reprezentaciju, VQ-VAE model diskretizira taj latentni prostor i uči nad njime.
- Učenje apriorne distribucije $p(z)$ - dok VAE učenjem pokušava postići zadanu aproksimiranu distribuciju, VQ-VAE dinamički uči diskretizirane vektore s jednolikom distribucijom, a tek nakon treniranja modela, distribucija priora diskretne skupine vektora se uči na ulaznim podacima.

2.4.1. Kvantizacija autoenkodera

VQ-VAE proširuje standardni autoenkoder dodavanjem diskretne komponente - kvantizator koji sadrži vizualni rječnik (*eng. codebook*). Vizualni rječnik povezuje vektore značajki vizualnih koncepata i odgovarajući indeks. Tijekom zaključivanja, izlaz iz mreže enkodera uspoređuje se sa svim vektorima iz rječnika, a vektor iz knjižice koji je najbliži u euklidskoj udaljenosti koristi se kao ulaz za dekoder:

$$z_q(x) = \arg\min_i ||z_e(x) - e_i||_2^2 \tag{2.5}$$

U prikazanoj jednadžbi $z_e(x)$ označava kontinuiranu reprezentaciju ulaza x koju je izračunao enkoder, e_i je i-ti vektor iz knjižice, a $z_q(x)$ je rezultirajući diskretna latentna varijabla koja se koristi kao ulaz za dekoder.



Slika 2.5: Prikaz izgleda i rada VQ-VAE modela. Enkoder prvo iz dobivenog ulaza generira latentnu reprezentaciju, čiji se latentni vektori zatim diskretiziraju u diskretnu latentnu reprezentaciju koja se sastoji od indeksa dobivenih iz rječnika. Dobivena diskretna latentna reprezentacija se zatim ponovno slaže koristeći vektore vizualnog rječnika kako bi se tako izgrađena reprezentacija iskoristila kao ulaz u dekoder. Preuzeto iz [10].

Zatim je zadatak dekodera rekonstruirati ulaz iz ovog kvantiziranog vektora kao u standardnoj formulaciji autoenkodera.

3. Generativno modeliranje, učenje reprezentacija i vrednovanje modela

Sada kada smo upoznati s osnovnim pojmovima autoenkoderskih modela, možemo se upoznati s matematičkim metodama koje se koriste za postizanje generativnog modeliranja, evaluiranja modela i učenja reprezentacija.

3.1. Nenadzirano učenje (eng. *unsupervised learning*)

Područje strojnog učenja sastoji se od tri glavna područja s obzirom na proces: nadzirano učenje, nenadzirano učenje i učenje pojačavanjem. Kod nenadziranog učenja dani podaci ne sadrže ciljane vrijednosti, već je cilj pronaći pravilnosti i strukturu u podacima.

Najčešće procesi koji zahtjevaju nenadzirano učenje su grupiranje (eng. *clustering*), smanjenje dimenzionalnosti (eng. *dimensionality reduction*), generativno modeliranje (eng. *generative modeling*) i otkrivanje anomalija (eng. *anomaly detection*).

Kod generativnog modeliranja, cilj je iskoristiti podatke za učenje kako bi se izgradila vjerojatnosna distribucija koja opisuje raspodjelu podataka. Najćešće mjere koje se koriste za evaluaciju generativnih modela su maksimalna izglednost (eng. *maximum likelihood*), log-izglednost (eng. *log-likelihood*) i rekonstrukcijski gubitak (eng. *reconstruction loss*). U sljedećim poglavljima prikazat ćemo kako se ove mjere koriste za evaluaciju modela autoenkodera (VAE i VQ-VAE).

3.2. Bayesovo zaključivanje

Bayesov zaključak jedan je od načina zaključivanja u statistici. Uz pomoć Bayesovog zaključivanja izmjenjujemo postojeće vjerovanje kako nam novi dokazi

dolaze. Vjerojatnost hipoteze z u ovisnosti o podatku x smo već izrazili jednadžbom 2.4.

Pri sljedećim izračunima koristit će se sljedeće oznake:

Tablica 3.1: Oznake

Oznaka	Opis
z	Latentna varijabla
x	Dokaz
$p(z)$	Apriorna gustoća vjerojatnosti
$p(x\|z)$	Gustoća vjerojatnost podatka x uzimajući u obzir hipotezu z
$p(z\|x)$	Gustoća vjerojatnost hipoteze z uzimajući u obzir podatak x
$q(z\|x)$	Aproksimacija $p(z\|x)$
$p(x)$	Vjerojatnost dokaza x

Važnost Bayesovog zaključka kod modela autoenkodera je u tome što nam omogućava izračun vjerojatnosti latentne varijable z uzimajući u obzir dokaze x i obrnuto. Upravo iz tog razloga, Bayesov zaključak je ključan za generativno modeliranje.

U praksi je često nemoguće ili je računski zahtjevno izračunati $p(x)$. Upravo zbog toga, u praksi se pri računanju koristi varijacijska inferencija koja pomoću različitih aproksimacija omogućuje izračun $p(z|x)$ i $p(x)$.

3.3. Kullback–Leiblerova divergencija

Kullback-Leiblerova (KL) divergencija, često nazivana i relativnom entropijom ili informacijskim gubitkom, je mjerilo razlike između dvije vjerojatnosne raspodjele.

Prikažimo prvo definiciju informacije. **Informacija** je mjera iznenađenja. Ako je događaj vjerojatan, onda je informacija koju dobivamo od tog događaja mala. Ako je događaj malo vjerojatan, onda je informacija koju dobivamo od tog događaja velika. To ima intuitivnog smisla jer ako nam netko kaže nešto "očito", tj. vrlo vjerojatno, tj. nešto što mi i gotovo svi već znamo, tada nam taj informator nije povećao količinu informacija koje posjedujemo. S druge strane, ako nam netko kaže nešto što je vrlo malo vjerojatno, tada nam je taj informator povećao količinu informacija koje posjedujemo. Najčešće se informacija podatka x s obzirom na distribuciju p mjeri u kao negativna log-izglednost, a definirana je kao:

$$I_p(x) = -\log p(x) \tag{3.1}$$

Razlika informacija između dvije distribucije s obzirom na podatak x određujemo oduzimanjem odgovarajućih informacija pojedinačnih distribucija:

$$\Delta I_{p,q}(x) = I_p(x) - I_q(x) = -\log p(x) + \log q(x) = \log \frac{q(x)}{p(x)} \tag{3.2}$$

Tada je očekivanje razlike informacija između dvije distribucije $q(x)$ i $p(x)$ definirano kao:

$$\mathrm{KL}(p(x)||q(x)) = E_p[\Delta I_{p,q}x] = \int p(x)\Delta I_{p,q}(x)dx = \int p(x)\log\frac{q(x)}{p(x)}dx \tag{3.3}$$

i

$$\mathrm{KL}(q(x)||p(x)) = E_q[\Delta I_{p,q}x] = \int q(x)\Delta I_{p,q}(x)dx = \int q(x)\log\frac{p(x)}{q(x)}dx \tag{3.4}$$

Treba napomenuti kako KL distribucija nije simetrična, tj. ne vrijedi :

$$\mathrm{KL}(p(x)||q(x)) \neq \mathrm{KL}(q(x)||p(x))$$

$\mathrm{KL}(p(x)||q(x))$ nam govori koliko je očekivanje razlike distribucije $q(x)$ s obzirom na distribuciju $p(x)$, dok nam $\mathrm{KL}(q(x)||p(x))$ govori koliko je očekivanje razlike distribucije $p(x)$ s obzirom na distribuciju $q(x)$. Od tuda dolazi i naziv divergencija, a ne udaljenost ili razlika jer udaljenosti i razlike moraju biti simetrične.

Jedno važno svojstvo KL divergencije je da je ona uvijek nenegativna, tj.

$$\mathrm{KL}(p(x)||q(x)) = -\int q(x)\log\frac{p(x)}{q(x)} \geq 0$$

.

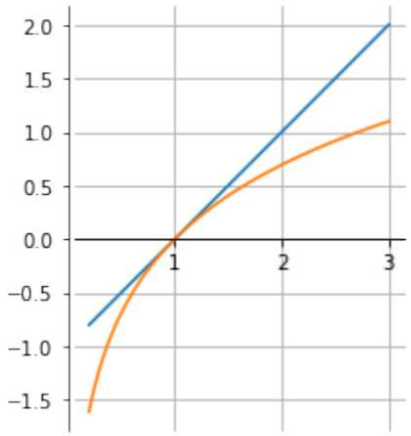
To važno svojstvo dokazat ćemo uz pomoć sljedeće nejednakosti:

$$\log x \leq x - 1 \tag{3.5}$$

Tada je:

$$\begin{aligned} -\mathrm{KL}(q(x)||p(x)) &= \int q(x)\log\frac{p(x)}{q(x)}dx \\ &\leq \int q(x)\left(\frac{p(x)}{q(x)} - 1\right)dx \\ &= \int p(x)dx - \int q(x)dx = 1 - 1 = 0 \end{aligned} \tag{3.6}$$

S obzirom na to da je $-\mathrm{KL}(q(x)||p(x)) \leq 0$, tada je $\mathrm{KL}(q(x)||p(x)) \geq 0$.



Slika 3.1: Grafovi funkcija $f(t) = \log t$ te $g(t) = t - 1$. Vidimo da uvijek vrijedi $\log t \leq t - 1$. Preuzeto sa [6]

3.4. Kriterijske funkcije za učenje modela

3.4.1. Obični autoenkoder

Kao što je već spomenuto, modeli autoenkodera se sastoje od dva dijela: enkodera i dekodera. Enkoder je funkcija $f(x)$ koja preslikava ulazni vektor x u latentni vektor z, tj. $z = f(x)$. Dekoder je funkcija $g(z)$ koja preslikava latentni vektor z u rekonstruirani vektor $\hat{x}$, tj. $\hat{x} = g(z)$.

Glavni cilj autoenkodera je prvotno dobro enkodirati sliku u vektor latentnog prostora s predodređenom distribucijom, a potom iz dobivenog latentnog vektora rekonstruirati ulazni podatak uz što manji gubitak. S tim ciljem definirat ćemo općeniti oblik funkcije gubitka autoenkodera kao:

$$\mathcal{L}_{AE} = \mathcal{L}_{rec} + \mathcal{L}_{reg} \tag{3.7}$$

gdje je $\mathcal{L}_{rec}$ funkcija gubitka rekonstrukcije, a $\mathcal{L}_{reg}$ funkcija gubitka regularizacije.

Rekonstrukcijski gubitak običnih autoenkodera prikazuju jednadžbe 2.2 i 2.1. S obzirom da obični autoenkoder nema informacija niti postavljenih pretpostavki o distribuciji latentnog prostora, odgovarajući gubitakneće imati regularizacijski član.

3.4.2. Varijacijski autoenkoder

Kako bismo mogli trenirati varijacijske autoenkodere, potrebno je definirati funkciju gubitka koja će u obzir uzimati i postavljene pretpostavke i definicije distribucije latentnog prostora. Enkoder vraća latentni vektor z koji je distribuiran

prema nekoj distribuciji $q(z|x)$ parametriziranoj pomoću dubokog modela ili neuralne mreže s parametrima θ, tako da ju možemo od sada pisati kao $q_\theta(z|x)$. Dekoder vraća rekonstruirani vektor $\hat{x}$ koji je distribuiran prema nekoj distribuciji $p(x|z)$ s parametrima ϕ, pa nju od sada zapisujemo kao $p_\phi(x|z)$.

Kako smo uveli aproksimacijsku distribuciju $q_\theta(z|x)$, potrebno je definirati mjeru koja će kvantificirati razliku između stvarne distribucije $p(z|x)$ i aproksimacijske distribucije $q_\theta(z|x)$. U tu svrhu koristimo KL divergenciju između $p(z|x)$ i $q_\theta(z|x)$, tj. $\mathrm{KL}(q_\theta(z|x)||p(z|x))$ koju želimo minimizirati i iz koje ćemo izvesti funkciju gubitka.

Prije nego što krenemo, potrebno je napomenuti da je $p(z|x_i)$ distribucija nepoznata, tj. ne znamo njenu funkciju gustoće vjerojatnosti. Zbog toga, potrebno je uvesti neku distribuciju $p(z)$ koja će biti naša apriorna pretpostavka o distribuciji $p(z|x_i)$. Najčešće je $p(z)$ definirana kao $p(z) = \mathcal{N}(0, I)$, tj. multivarijatna normalna distribucija s nul-vektorom očekivanja i jediničnom kovarijacijskom matricom. Ova distribucija je često korištena u praksi, a razlog tome je što jelako izračunati KL divergenciju s obzirom na nju. Očekivanje je u ovom slučaju nul-vektor jer želimo da se svi latentni vektori nalaze oko nule, a kovarijacijska matrica je jedinična jer želimo da svi latentni vektori budu nekorelirani.

KL divergenciju i njenu nejednakost možemo zapisati kao:

$$\mathrm{KL}(q_\theta(z|x_i)||p(z|x_i)) = -\int q_\theta(z|x_i) \log \frac{p(z|x_i)}{q_\theta(z|x_i)} dz \geq 0 \tag{3.8}$$

Primjenom Bayesove formule 2.4 na $p(z|x_i)$ dobivamo:

$$\begin{aligned} \mathrm{KL}(q_\theta(z|x_i)||p(z|x_i)) &= -\int q_\theta(z|x_i) \log \frac{p_\phi(x_i|z)p(z)}{q_\theta(z|x_i)p(x_i)} dz \geq 0 \\ \mathrm{KL}(q_\theta(z|x_i)||p(z|x_i)) &= -\int q_\theta(z|x_i) \left[\log \frac{p_\phi(x_i|z)p(z)}{q_\theta(z|x_i)} - \log p(x_i)\right] dz \geq 0 \end{aligned} \tag{3.9}$$

Daljnjim raspisivanjem desne strane:

$$\begin{aligned} &-\int q_\theta(z|x_i) \log \frac{p_\phi(x_i|z)p(z)}{q_\theta(z|x_i)} dz + \int q_\theta(z|x_i) \log p(x_i) dz \geq 0 \\ &-\int q_\theta(z|x_i) \log \frac{p_\phi(x_i|z)p(z)}{q_\theta(z|x_i)} dz + \log p(x_i) \int q_\theta(z|x_i) dz \geq 0 \end{aligned} \tag{3.10}$$

$$
\begin{aligned}
\log p(x_i) \int q_\theta(z|x_i)dz &\geq \int q_\theta(z|x_i) \log \frac{p_\phi(x_i|z)p(z)}{q_\theta(z|x_i)}dz \\
\log p(x_i) &\geq \int q_\theta(z|x_i) \log \frac{p_\phi(x_i|z)p(z)}{q_\theta(z|x_i)}dz \\
\log p(x_i) &\geq \int q_\theta(z|x_i) \left[\log p_\phi(x_i|z) + \log p(z) - \log q_\theta(z|x_i)\right] dz \\
\log p(x_i) &\geq \int q_\theta(z|x_i) \log \frac{p(z)}{q_\theta(z|x_i)}dz + \int q_\theta(z|x_i) \log p_\phi(x_i|z)dz \\
\log p(x_i) &\geq -\mathrm{KL}(q_\theta(z|x_i)||p(z)) + \mathbb{E}_{q_\theta(z|x_i)}\left[\log p_\phi(x_i|z)\right]
\end{aligned}
\tag{3.11}
$$

Desna strana gornje jednadžbe je Donja granica gustoće vjerojatnosti podatka (engl. *Evidence Lower Bound*, ELBO), poznata i kao varijacijska donja granica. Naziva se tako jer ograničava vjerojatnost podataka, koju želimo maksimizirati. Stoga, maksimizacija ELBO-a neizravno maksimizira logaritam vjerojatnosti naših podataka, a to je upravo osnovna ideja varijacijskog zaključivanja, budući da je izravna maksimizacija logaritma vjerojatnosti $p(x_i)$ nije traktabilna jer implicira integriranje preko svih z.

$$
\begin{aligned}
\log p(x_i) &= \log \int_z p(x_i, z) \\
&= \log \int_z p(x_i|z)p(z)
\end{aligned}
\tag{3.12}
$$

Optimizacija desne strane nejednadžbe [**?**] može napredovati sve dok se desna strana ne izjednači s lijevom stranom. Ako se to dogodi, jednadžba [**?**] će iz nejednakosti preći u jednakost, a to je upravo ono što želimo postići. Sada kada smo dobili ELBO, možemo ga koristiti za izračun funkcije gubitka.

Dakle, funkcija gubitka je:

$$
\begin{aligned}
\mathcal{L}(\theta, \phi; x_i) &= -\mathrm{ELBO}(\theta, \phi; x_i) \\
\mathcal{L}(\theta, \phi; x_i) &= -\left(-\mathrm{KL}(q_\theta(z|x_i)||p(z)) + \mathbb{E}_{q_\theta(z|x_i)}\left[\log p_\phi(x_i|z)\right]\right) \\
\mathcal{L}(\theta, \phi; x_i) &= \mathrm{KL}(q_\theta(z|x_i)||p(z)) - \mathbb{E}_{q_\theta(z|x_i)}\left[\log p_\phi(x_i|z)\right]
\end{aligned}
\tag{3.13}
$$

Možemo uočiti da je funkcija gubitka varijacijskog autoenkodera poprimila oblik funkcije gubitka koji smo definirali formulom 3.7, pri čemu lijevi član funkcije gubitka predstavlja regularizacijski gubitak između aproksimirane distribucije $q_\theta(z|x_i)$ i stvarne distribucije latentnog prostora $p(z)$, a desni član predstavlja rekonstrukcijski gubitak.

No, sada je potrebno pojednostaviti očekivanje $\mathbb{E}_{q_\theta(z|x_i)}\left[\log p_\phi(x_i|z)\right]$ i KL divergenciju $\mathrm{KL}(q_\theta(z|x_i)||p(z))$ kako bismo mogli programski izvesti ovako definiranu funkciju gubitka.

Računanje očekivanja

Jedna od prvih metoda računanja očekivanja je tehnika Monte Carlo uzorkovanja (engl. *Monte Carlo sampling*). Ova tehnika se koristi za aproksimaciju očekivanja kontinuirane slučajne varijable, tako da se aproksimacija računa kao srednja vrijednost funkcije koja se uzorkuje iz distribucije slučajne varijable. U ovom slučaju, funkcija koja se uzorkuje je $\log p_\phi(x_i|z)$, a distribucija iz koje se uzorkuje je $q_\theta(z|x_i)$. Dakle, aproksimacija očekivanja je:

$$\mathbb{E}_{q_\theta(z|x_i)}\left[\log p_\phi(x_i|z)\right] \approx \frac{1}{L}\sum_{l=1}^{L} \log p_\phi(x_i|z^{(l)}) \tag{3.14}$$

Ovakva metoda računanja aproksimacije je računalno zahtjevna i neefikasna, jer je potrebno uzorkovati veliki broj uzoraka kako bi aproksimacija bila dovoljno dobra.

Kao standardna metoda računanja očekivanja u stohastičkom računanju gradijenta (engl. *stochastic gradient descent*, SGD), uzima se samo jedan dobiveni uzorak, tj. $L = 1$. S obzirom da je traženo očekivanje stohastički aproksimirano, funkcija gubitka varijacijskog autoenkodera se naziva *stohastička funkcija gubitka* (engl. *stochastic loss function*). Dakle, funkcija gubitka varijacijskog autoenkodera je sada:

$$\mathcal{L}(\theta, \phi; x_i) = -\frac{1}{L}\sum_{l=1}^{L} \log p_\phi(x_i|z^{(l)}) + \mathbb{KL}\left[q_\theta(z|x_i)||p(z)\right], L = 1 \tag{3.15}$$

Osim toga, logaritamska vjerojatnost $\log p_\phi(x_i|z)$ kod podataka s binarnim vrijednostima može se prikazati i kao binarna križna entropija (engl. *binary cross-entropy*) između ulaza x_i i izlaza dekodera x_i' pod uvjetom da je x_i aproksimiran iz Bernoullijeve distribucije $\mathrm{Bernoulli}(f(z))$.

Računanje KL divergencije

Dobivena KL divergencija se također može analitički izračunati. Naime, opet uzimamo u obzir da su distribucije $q_\theta(z|x_i)$ i $p(z)$ normalne distribucije. Tada ih možemo prikazati kao:

$$\begin{aligned} q_\theta(z|x_i) &= \mathcal{N}(\mu_\theta(x_i), \sigma_\theta(x_i)) \\ p(z) &= \mathcal{N}(\mu_p, \sigma_p) \end{aligned} \tag{3.16}$$

tj. matematički:

$$
\begin{aligned}
q_\theta(z|x_i) &= \frac{1}{\sqrt{2\pi\sigma_\theta(x_i)^2}} \exp\left(-\frac{(z-\mu_\theta(x_i))^2}{2\sigma_\theta(x_i)^2}\right) \\
p(z) &= \frac{1}{\sqrt{2\pi\sigma_p}} \exp\left(-\frac{(z-\mu_p)^2}{2\sigma_p^2}\right)
\end{aligned}
\tag{3.17}
$$

Tada je KL divergencija između dvije normalne distribucije:

$$
\begin{aligned}
&-\mathrm{KL}\left[q_\theta(z|x_i)\|p(z)\right] = \int q_\theta(z|x_i)\log\frac{q_\theta(z|x_i)}{p(z)}dz \\
&= \int \frac{1}{\sqrt{2\pi\sigma_\theta(x_i)^2}} \exp\left(-\frac{(z-\mu_\theta(x_i))^2}{2\sigma_\theta(x_i)^2}\right) \log\left(\frac{\frac{1}{\sqrt{2\pi\sigma_p^2}}\exp\left(-\frac{(z-\mu_p)^2}{2\sigma_p^2}\right)}{\frac{1}{\sqrt{2\pi\sigma_\theta(x_i)^2}}\exp\left(-\frac{(z-\mu_\theta(x_i))^2}{2\sigma_\theta(x_i)^2}\right)}\right) dz \\
&= \frac{1}{\sqrt{2\pi\sigma_\theta(x_i)^2}} \int \exp\left(-\frac{(z-\mu_\theta(x_i))^2}{2\sigma_\theta(x_i)^2}\right)\left\{-\log\sigma_p - \frac{(z-\mu_p)^2}{2\sigma_p^2} + \log\sigma_\theta(x_i) + \frac{(z-\mu_\theta(x_i))^2}{2\sigma_\theta(x_i)^2}\right\} dz \\
&= \frac{1}{\sqrt{2\pi\sigma_\theta(x_i)^2}} \int \exp\left(-\frac{(z-\mu_\theta(x_i))^2}{2\sigma_\theta(x_i)^2}\right)\left\{\log\frac{\sigma_\theta(x_i)}{\sigma_p} - \frac{(z-\mu_p)^2}{2\sigma_p^2} + \frac{(z-\mu_\theta(x_i))^2}{2\sigma_\theta(x_i)^2}\right\} dz
\end{aligned}
\tag{3.18}
$$

Prikažimo sada izraz iz jednadžbe 3.17 kao očekivanje:

$$
\begin{aligned}
&-\mathrm{KL}\left[q_\theta(z|x_i)\|p(z)\right] = \mathbb{E}_q\left\{\log\frac{\sigma_\theta(x_i)}{\sigma_p} - \frac{(z-\mu_p)^2}{2\sigma_p^2} + \frac{(z-\mu_\theta(x_i))^2}{2\sigma_\theta(x_i)^2}\right\} \\
&= \log\frac{\sigma_\theta(x_i)}{\sigma_p} - \frac{1}{2\sigma_p^2}\mathbb{E}_q\left[(z-\mu_p)^2\right] + \frac{1}{2\sigma_\theta(x_i)^2}\mathbb{E}_q\left[(z-\mu_\theta(x_i))^2\right]
\end{aligned}
\tag{3.19}
$$

Sada možemo iskoristit svojstvo normalne distribucije da je $\mathbb{E}_q\left[(z-\mu_\theta(x_i))^2\right] = \sigma_\theta(x_i)^2$ pa je:

$$
\begin{aligned}
&-\mathrm{KL}\left[q_\theta(z|x_i)\|p(z)\right] = \log\frac{\sigma_\theta(x_i)}{\sigma_p} - \frac{1}{2\sigma_p^2}\mathbb{E}_q\left[(z-\mu_p)^2\right] + \frac{1}{2\sigma_\theta(x_i)^2}\mathbb{E}_q\left[(z-\mu_\theta(x_i))^2\right] \\
&= \log\frac{\sigma_\theta(x_i)}{\sigma_p} - \frac{1}{2\sigma_p^2}\mathbb{E}_q\left[(z-\mu_\theta(x_i)+\mu_\theta(x_i)-\mu_p)^2\right] + \frac{1}{2\sigma_\theta(x_i)^2}\sigma_\theta(x_i)^2
\end{aligned}
\tag{3.20}
$$

Sada ako odvojimo $z-\mu_\theta(x_i)$ i $\mu_\theta(x_i)-\mu_p$ možemo rastaviti kvadrat u očekivanju na sljedeći način:

$$
\begin{aligned}
&- \mathrm{KL}\left[q_\theta(z|x_i)\|p(z)\right] = \log\frac{\sigma_\theta(x_i)}{\sigma_p} - \frac{1}{2\sigma_p^2}\mathbb{E}_q\left[(z-\mu_\theta(x_i)+\mu_\theta(x_i)-\mu_p)^2\right] + \frac{1}{2} \\
&= \log\frac{\sigma_\theta(x_i)}{\sigma_p} - \frac{1}{2\sigma_p^2}\mathbb{E}_q\left[(z-\mu_\theta(x_i))^2 + 2(z-\mu_\theta(x_i))(\mu_\theta(x_i)-\mu_p) + (\mu_\theta(x_i)-\mu_p)^2\right] + \frac{1}{2} \\
&= \log\frac{\sigma_\theta(x_i)}{\sigma_p} - \frac{1}{2\sigma_p^2}\Bigg\{\mathbb{E}_q\left[(z-\mu_\theta(x_i))^2\right] + 2\mathbb{E}_q\left[(z-\mu_\theta(x_i))(\mu_\theta(x_i)-\mu_p)\right] \\
&+ \mathbb{E}_q\left[(\mu_\theta(x_i)-\mu_p)^2\right]\Bigg\} + \frac{1}{2}
\end{aligned}
\tag{3.21}
$$

$$
\begin{aligned}
&= \log\frac{\sigma_\theta(x_i)}{\sigma_p} - \frac{1}{2\sigma_p^2}\Big\{\sigma_\theta(x_i) + 2*0*(z-\mu_p) + (\mu_\theta(x_i)-\mu_p)^2\Big\} + \frac{1}{2} \\
&= \log\frac{\sigma_\theta(x_i)}{\sigma_p} - \frac{\sigma_\theta(x_i) + (\mu_\theta(x_i)-\mu_p)^2}{2\sigma_p^2} + \frac{1}{2}
\end{aligned}
\tag{3.22}
$$

Sjetimo se sada da je $\sigma_p = 1$ i da je $\mu_p = 0$ pa je:

$$
- \mathrm{KL}\left[q_\theta(z|x_i)\|p(z)\right] = \frac{1}{2}\left[1 + \log\sigma_\theta(x_i)^2 - \sigma_\theta(x_i)^2 - \mu_\theta(x_i)^2\right] \tag{3.23}
$$

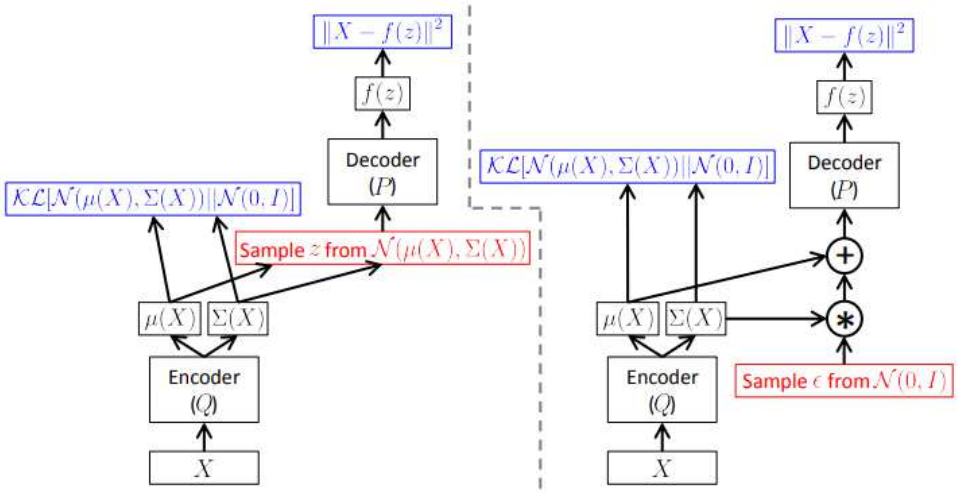
Sada možemo izraziti ukupni gubitak kao:

$$
\mathcal{L}(\theta,\phi;x_i) = -\frac{1}{2}\left[1 + \log\sigma_\theta(x_i)^2 - \sigma_\theta(x_i)^2 - \mu_\theta(x_i)^2\right] - \frac{1}{L}\sum_{l=1}^{L}\log p_\phi(x_i|z^{(l)}) \tag{3.24}
$$

Kako bismo jednostavno izlučili informacije o parametrima distribucije $q_\theta(z|x_i)$, bez potrebe za višestrukim uzorkovanjem i računanjem $\mu_\theta(x_i)$ i $\sigma_\theta(x_i)$ pomoću klasičnih statističkih funkcija, koristi se tehnika *reparametrizacije* (engl. *reparametrization*). Ova tehnika se koristi za uzorkovanje iz kontinuirane vjerojatnosne distribucije, tako da se uzorci uzimaju iz distribucije slučajne varijable koja je nezavisna o parametrima distribucije. U ovom slučaju, distribucija slučajne varijable je $\mathcal{N}(0,1)$, tj. normalna distribucija s očekivanjem 0 i varijancom 1. Dakle, uzorkovanje se vrši tako da se uzorci uzimaju iz distribucije $\mathcal{N}(0,1)$, a zatim se ti uzorci transformiraju u uzorke iz distribucije $q_\theta(z|x_i)$, tj. iz distribucije kodirane reprezentacije z. Transformacija se vrši tako da se uzorci iz distribucije $\mathcal{N}(0,1)$ pomnože sa standardnom devijacijom distribucije $q_\theta(z|x_i)$, tj. s $\sigma_\theta(x_i)$ koja je pritom dobivena iz našeg enkodera te im se nadoda očekivanje distribucije $q_\theta(z|x_i)$, tj. $\mu_\theta(x_i)$ koja je također dobivena iz našeg enkodera. Dakle, uzorkovanje se vrši na sljedeći

način:

$$z^{(l)} = \mu_\theta(x_i) + \sigma_\theta(x_i) \cdot \epsilon^{(l)}$$
$$\epsilon^{(l)} \sim \mathcal{N}(0, 1) \tag{3.25}$$



Slika 3.2: Desna slika predstavlja rad autoenkodera pri čemu se latentna varijabla uzorkuje iz distribucije $\mathcal{N}(\mu(X), \sigma(X))$, dok lijeva slika predstavlja rad autoenkodera pri čemu se latentna varijabla, koristeći reparametrizaciju, uzorkuje iz distribucije $\mathcal{N}(0, 1)$, a zatim transformira u distribuciju $\mathcal{N}(\mu(X), \sigma(X))$. Preuzeto sa [2]

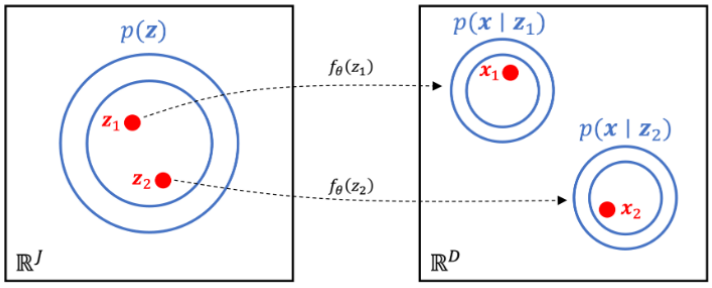
Ovakvim postupkom, moguće je izračunati gradijente funkcije gubitka $\mathcal{L}(\theta, \phi; x_i)$ po parametrima θ i ϕ i koristiti ih za optimizaciju modela.

Generiranje novih primjera

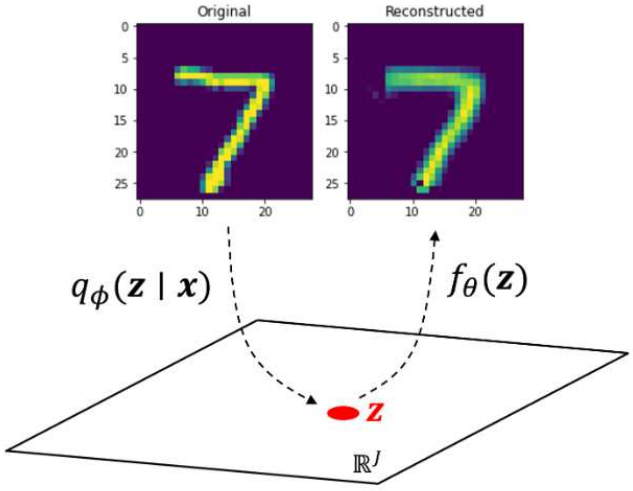
Kada je model naučen, možemo ga iskoristiti za generiranje novih primjera. Kako je model naučen da aproksimira distribuciju $p(z)$, možemo uzorkovati latentnu reprezentaciju te primjenom dekodera doći do parametara distribucije $p(x|z)$ iz koje možemo uzorkovati podatke.

Jedna od ključnih prednosti VAE modela je mogućnost variranja latentnih varijabli. Umjesto da se uzima samo jedan uzorak iz latentnog prostora, možemo uzeti više uzoraka i eksperimentirati s različitim vrijednostima. Ovo nam omogućuje da generiramo različite varijacije primjera unutar iste distribucije.

Generirani primjeri pomoću VAE modela često pokazuju sposobnost modela da nauči karakteristične značajke ulaznih podataka. Oni mogu biti vrlo slični stvarnim primjerima, ali također mogu sadržavati i neke varijacije ili kreativne interpretacije.



Slika 3.3: Vizualni prikaz procesa generiranja novih primjera. Preuzeto sa [1]



Slika 3.4: Vizualni prikaz procesa rekonstruiranja slika. Preuzeto sa [1]

VAE modeli omogućuju generiranje novih primjera sličnih ulaznim podacima, a istovremeno nude i mogućnost eksploracije latentnog prostora i stvaranja potpuno novih primjera koji nisu bili prisutni u skupu podataka za trening.

Slabosti VAE modela

Iako VAE modeli imaju mnoge prednosti nabrojane u prošlom odjeljku 3.4.2, također imaju i neke negativne strane. Neki od problema su:

– **Kolaps posteriora** - Pojava u kojoj se model ne može naučiti razlikovati različite vrijednosti, a to je posljedica prejakog utjecaja regularizacije. To znači da se tokom učenja modela, zbog mogućeg jakog utjecaja regularizacije, distribucija posteriora $p(z|x)$ može potpuno preklopiti s priorom $p(z)$ i time prestati razlikovati različite vrijednosti ulaza. Primjer kolapsa posteriora

prikazan je u sljedećem poglavlju.

- **Statičnost priora** - Pretpostavka distribucije priora $p(z)$ nije uvijek zadovoljena, pogotovo kod kompleksnijih podataka. Zbog nefleksibilnosti priora model ne može naučiti rekonstruirati kompleksnije podatke.
- **Pretpostavke modela** - Rekonstrukcijski gubitak modela pretpostavlja dekoreliranost piksela, što u stvarnosti nije uvijek zadovoljeno.

3.4.3. Funkcija gubitka VQVAE modela

Dobivanje funkcije gubitka za VQVAE model je složenije nego za VAE model. Kao što je već spomenuto, VQVAE model se sastoji od enkodera, dekodera i kvantizatora. Enkoder i dekoder su isti kao i u VAE modelu, dok je kvantizator nova komponenta. Kvantizator kvantizira ulazne podatke u diskretne vrijednosti s obzirom na već spomenutu knjižicu koja se sastoji od K vektora $e_k \in \mathbb{R}^D$ gdje k predstavlja diskretne vrijednosti indeksa vektora. Ulazni podatak x se kvantizira tako da se pronađe najbliži vektor e_k i kao rezultat se dobije indeks tog vektora. Kvantizator je definiran funkcijom [**?**].

Prednost učenja VQVAE modela je što $prior$ distribuciju $p(z)$ možemo fiksirati da bude uniformna i jednaka među svim elementima knjižice. Stoga nam se sada proces učenja dijeli na dvije faze:

- Prva faza - Učenje modela s jednog kraja (dekoder) na drugi kraj (enkoder) (engl. *end-to-end*) kojem je glavni fokus naučiti parametre enkodera za postizanje $q_\theta(z|x)$, dekodera za postizanje $p_\phi(x|z)$ i elemenata knjižice e_i za strukturiranje latentnog prostora.
- Druga faza - Učenje priora $p(z)$ pomoću autoregresijskih modela (eng. *autoregressive models*) s ciljem poboljšanja generativnog modeliranja.

U ovom poglavlju ćemo izvesti funkciju gubitka za prvu fazu učenja modela.

Proces kvantizacije [**?**] uzrokuje probleme za propagaciju gradijenta. Kako je kvantizacija diskretna, gradijent se ne može propagirati kroz nju. Kako bi se riješio ovaj problem, apoksimira se gradijent koristeći tzv. *prolazni* estimator (eng. *straight-through* estimator) tako da se ulazi iz dekodera $z_q(x)$ kopiraju i šalju na izlaz enkodera $z_e(x)$. Prolaz takvog gradijenta $\nabla_z \mathbf{L}$ je prikazan na slici 2.5.

Kako zbog prolaznog estimatora gradijenta elementi knjižice e_i ne dobivaju nikakav gradijent iz rekonstrukcijskog gubitka, a kao važan element modela, potrebno

je potaknuti kvantizator na učenje tih elemenata. Iz tog razloga, funkcija gubitka će se sastoji od tri dijela. Prvi dio je funkcija gubitka rekonstrukcije $\mathcal{L}_r$ koja je ista kao i u VAE modelu. Drugi dio je funkcija gubitka kvantizacije $\mathcal{L}_z$ koja mjeri koliko je dobro kvantizator kvantizirao ulazne podatke te joj je cilj pomaknuti elemente rječnika e_i prema izlazima enkodera $z_e(x)$. Treći dio je opet funkcija gubitka kvantizatora $\mathcal{L}_c$, no ona je izmijenjena tako da se izlazi enkodera prilagode elementima knjižice e_i.

Za razliku od VAE modela, VQVAE model sadrži konačan broj elemenata knjižice e_i što znači da je moguće izračunati log-izglednost kao sumu po svim elementima knjižice. Sada izračun log-izgleđnosti izgleda ovako:

$$\log p(x) = \log \sum^{k} p(x|z_k)p(z_k) \tag{3.26}$$

Ovaj račun vrijedi jer je dekoder treniran na $z = z_q(x)$, tako da kada model u potpunosti konvergira, dekoder ne bi trebao imati vjerojatnosnu "masu" $p(x|z)$ za $z \neq z_q(x)$. Sada empirijski možemo aproksimirati log-izglednost koristeći Jensenovu nejednakost koja je izražena sljedećom jednadžbom:

$$\mathbb{E}_{f(X)} \geq f(\mathbb{E}_X) \tag{3.27}$$

Gdje je X slučajna varijabla s vjerojatnosnom raspodjelom p, a f konveksna funkcija nad određenim područjem. Ako sada nad ovom nejednakošću primjenimo logaritam, dobivamo:

$$\begin{aligned} &\mathbb{E}_{\log p(x)} \geq \log \mathbb{E}_{p(x)} \\ &\log p(x) \geq \log p(x|z_q(x))p(z_q(x)) \end{aligned} \tag{3.28}$$

Funkcija gubitka rekonstrukcije

Funkcija gubitka rekonstrukcije je ista kao i u VAE modelu i definirana je funkcijom 3.24. Njen izračun izvodi se jednako kao i kod VAE modela.

$$\mathcal{L}_x = -\mathbb{E}_{x\sim p_{data}(x)} \left[\log p_\theta(x|z_q(x))\right] \tag{3.29}$$

Funkcija gubitka kvantizacije

Funkcija gubitka kvantizacije mjeri koliko je dobro kvantizator kvantizirao ulazne podatke, a prikazuje se kao L_2 gubitak između elementa knjižice e_i i izlaza iz enkodera $z_e(x)$. No, kako je već spomenuto, ista funkcija gubitka se koristi i za pomak elementa

knjižice prema izlazima enkodera i za pomak izlaza enkodera prema elementima knjižice. Zbog toga je potrebno uvesti neke oznake kako bi se razlikovalo koji dio funkcije gubitka se računa.

Funkcija gubitka kvantizacije za pomak elemenata knjižice prema izlazima enkodera je definirana funkcijom:

$$\mathcal{L}_z = ||sg[z_e(x)] - e_i||^2 \tag{3.30}$$

Pri ovom računu uvedena je nova oznaka sg koja označava *stop-gradijent* (eng. *stop-gradient*) operator. Ovaj operator zaustavlja propagaciju gradijenta kroz njega. Ovaj operator je potreban kako bi se spriječila propagacija gradijenta kroz enkoder.

Funkcija gubitka kvantizacije za pomak izlaza enkodera prema elementima knjižice je definirana funkcijom:

$$\mathcal{L}_c = ||z_e(x) - sg[e]||^2 \tag{3.31}$$

Ovdje se također koristi *stop-gradijent* operator kako bi se spriječilo da se gradijent propagira kroz kvantizator i knjižicu.

Ukupna funkcija gubitka sada je definirana kao:

$$\mathcal{L} = \mathcal{L}_x + \mathcal{L}_z + \beta\mathcal{L}_c \tag{3.32}$$

Pri čemu je β hiperparametar koji određuje kolika će se prilagodba elementima knjižice propuštati na izlaz enkodera. Zbog hiperparametra β je bilo potrebno predstaviti dva različita oblika funkcije gubitka kvantizacije. Prema [10] model bi trebao biti robusan na hiperparametar β za vrijednosti između 0.1 i 2.0.

3.5. Generativno modeliranje s VQVAE

U prošlom odjeljku 3.4.3 je opisana funkcija gubitka VQ-VAE modela koja se koristi za prvu fazu učenja modela. Dakle, u prvoj fazi učenja, uz to što je naučio rekonstruirati podatke, model je naučio kako kvantizirati ulazne podatke, što znači da naučene diskretne latentne varijable iz knjižice e_i kvalitetno prikazuju strukturu latentnog prostora za ulazne podatke. No, u toj fazi učenja, distribucija diskretnog latentnog prostora je bila konstantna i uniformna.

S takvom definiranom strukturom latentnog prostora, želimo odbaciti uniformnost apriorne distribucije $p(z)$ i želimo naučiti novu koja će našim diskretnim latentnim varijablama pridružiti odgovarajuće vjerojatnosti. A upravo je to cilj druge faze

učenja. U drugoj fazi želimo za naše određene ulazne podatke naučiti distribuciju diskretnih latentnih varijabli $p(z)$. Valja napomenuti da je distribucija diskretnih latentnih varijabli $p(z)$ kategorička.

Intuitivno možemo zamisliti da će različiti skupovi ulaznih podataka imati različite distribucije diskretnih latentnih varijabli. Na primjer, skup slika pasa će imati drugačiju distribuciju diskretnih latentnih varijabli od skupa slika mačaka. Zbog toga je potrebno naučiti različite distribucije diskretnih latentnih varijabli za različite skupove ulaznih podataka. U ovom primjeru mačaka i pasa, to bi značilo da ako želimo generirati sliku psa, onda bi trebali koristiti distribuciju diskretnih latentnih varijabli koja je naučena na skupu slika pasa.

Učenje distribucije diskretnih latentnih varijabli $p(z)$ postiže se tako da se uz pomoć enkodera istreniranog VQ-VAE modela enkodira određeni skup ulaznih podataka. S tako enkodiranim ulaznim podacima, koristeći autoregresivni model, nauči se distribucija diskretnih latentnih varijabli. U radu [10] za učenje distribucije diskretnih latentnih varijabli koriste autoregresivni model PixelCNN za slike, te WaveNet za audio podatke.

Za kraj je potrebno spomenuti kako su obje faze učenja nezavisne jedna o drugoj, tj. učenje distribucije diskretnih latentnih varijabli ne utječe na učenje kvantizatora i knjižice, te obrnuto. Kao sljedeći koraci istraživanja mogućnosti treniranja VQ-VAE modela, u radu [10] se predlaže paralelno provođenje navedenih faza.

4. Metode istraživanja

U sklopu rada provedena su dva istraživanja. Prvo istraživanje je provedeno nad VAE modelima, pri čemu je cilj bio utvrditi utjecaj različitih arhitektura i određenih parametara modela na kvalitetu generiranih slika.

Drugo istraživanje je provedeno nad VQ-VAE modelima, pri čemu je cilj bio ispitati svojstva takvih modela, te utvrditi utjecaj različitih parametara i različitih manipulacija latentnih vektora na kvalitetu generiranih slika.

Prikazat ćemo jednostavan primjer raspodjele latentnog prostora za VAE model te ćemo prikazati jednostavne oblike kvantiziranih vektora za VQVAE modele.

4.1. Programska podrška

Za izvedbu eksperimenata korišteni su sljedeći alati:

- Python 3.8 - programski jezik u kojem su napisani svi programi i procedure potrebne za izvedbu eksperimenata.
- TensorFlow 2.12.0 [1] - biblioteka za strojno učenje koja je korištena za treniranje i evaluiranje modela, te za manipulaciju podacima.
- Keras 2.12.0 [2] - biblioteka za strojno učenje koja je korištena za definiranje modela i njegovih slojeva.
- NumPy 1.20.2 [3] - biblioteka za znanstveno računanje koja je korištena za manipulaciju podacima i matematičke operacije.
- Matplotlib 3.4.1 - biblioteka korištena za vizualizaciju slika i dobivenih rezultata.
- Jupyter Notebook 6.3.0 - alat korišten za provedbu određenih eksperimenata.

[1] https://www.tensorflow.org/
[2] https://keras.io/
[3] https://numpy.org/

4.2. Skup podataka

Za potrebe istraživanja korišten je skup podataka MNIST koji se sastoji od 70000 slika dimenzija 28x28 koje prikazuju rukom napisane znamenke od 0 do 9. Skup podataka je podijeljen na 60000 slika za učenje i 10000 slika za testiranje, no s obzirom da se modeli izvedeni iz autoekodera nenadgledano uče, ta dva skupa mogu biti spojena. Slike su normalizirane tako da vrijednosti piksela pripadaju intervalu [0, 1].

4.3. Modeli

Za potrebe istraživanja korišteni su sljedeći modeli:

- VAE model s dva konvolucijska sloja i dva potpuno povezana u enkoderu i dekoderu:
  - Funkcije aktivacije: ReLU, sigmoida
  - Rekonstrukcijski gubitak: MSE
- VAE model s dva konvolucijska sloja i dva potpuno povezana u enkoderu i dekoderu:
  - Funkcije aktivacije: ReLU, sigmoida
  - Rekonstrukcijski gubitak: Binarna križna entropija
- VAE model s dva konvolucijska sloja u enkoderu i dekoderu:
  - Funkcije aktivacije: ReLU, sigmoida
  - Rekonstrukcijski gubitak: MSE
- VAE model s dva konvolucijska sloja u enkoderu i dekoderu:
  - Funkcije aktivacije: ReLU, sigmoida
  - Rekonstrukcijski gubitak: Binarna križna entropija
- VQ-VAE model s dva konvolucijska sloja u enkoderu i dekoderu:
  - Funkcije aktivacije: ReLU
  - Rekonstrukcijski gubitak: MSE
- Pretrenirani dVAE (eng. *Discrete VAE*) (podvrsta VQ-VAE model) definiran u [3] i [7]:

Dimenzije latentnih prostora VAE modela su svima jednake i iznosi 2, dok je dimenzija latentnog VQVAE modela 16.

4.4. Optimizacija

Optimizacija modela uključuje prilagodbu parametara modela kako bi se minimizirala definirana funkcija gubitka. Ovaj postupak može uključivati primjenu različitih optimizacijskih algoritama kao što su stohastički gradijentni spust (SGD) ili Adam. Proces optimizacije uključuje iterativno ažuriranje parametara modela koristeći gradijente funkcije gubitka. Ovaj postupak tipično zahtijeva više prolaza kroz skup podataka kako bi se postigla konvergencija i poboljšala performansa modela. Važno je odabrati prikladnu stopu učenja koja omogućuje stabilno ažuriranje parametara modela.

Prilikom izvođenja eksperimenata u sljedećem poglavlju 5 korišten je Adam kao optimizacijski algoritam za sve modele.

4.5. Evaluacija

Evaluacija modela uključuje procjenu performansi modela na temelju definiranih metrika. S obzirom da se radi o nenadgledanom učenju, metrike evaluacije koje se mogu koristiti su ograničene. U ovom radu korištene su sljedeće metrike evaluacije:

- Rekonstrukcijska pogreška - mjeri koliko dobro model rekonstruira ulazne podatke. Funkcije korištene za računanje rekonstrukcijske pogreške su MSE i binarna križna entropija.
- Log-izglednost - s obzirom da je računanje log-izglednosti $\log p(x)$ netraktabilno, indirektno računamo doljnju granicu log-izglednosti (ELBO).
- Vizualna inspekcija - ponekad nije dovoljno koristiti samo kvantitativne metrike evaluacije, već je potrebno i vizualno provjeriti kvalitetu rekonstrukcije.

4.6. Izvedba

Treniranje je provedeno za svaki navedeni model kroz 15 epoha s veličinom gomile (eng. *batch-size*) od 128 slika, izuzevši predtrenirani gotovi model dVAE. Osim evaluacije modela, provedena je i usporedba performansi modela s različitim parametrima. Također, ispitana su određena svojstva dobivenih modela.

Tako smo kod VAE modela ispitali različite parametre kao što su: različite rekonstrukcijske funkcije (MSE i binarna križna entropija), različite arhitekture

modela (Konvolucijski slojevi i potpuno povezani slojevi), te smo ispitali i njihove utjecaje na performanse modela.

Kod VQ-VAE i dVAE modela ispitali smo različita svojstva modela kao što su: otpornost na manipuliranje latentnih vektora, bilo da se radi o transformacijama kao što su rotacije, translacije ili kombiniranje latentnih vektora jedne slike s latentnim vektorima druge slike.

Također smo vizualno usporedili rekonstrukcije slika za različite modele i različite parametre.

5. Rezultati

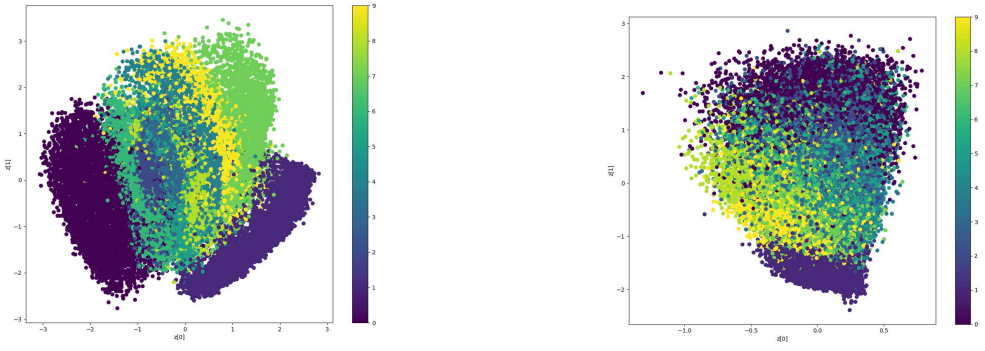
U ovom poglavlju prikazani su rezultati različitih metoda istraživanja provedenih na modelima opisanim u poglavlju 4.

5.1. Istraživanja nad VAE modelima

5.1.1. Usporedba različitih arhitektura

Usporedbu provodimo između arhitekture VAE modela s konvolucijskim slojevima na izlazu enkodera i ulazu dekodera, te VAE modela s potpuno povezanim slojevima na izlazu enkodera i ulazu dekodera. Obje arhitekture su u ovom slučaju istrenirane s binarnom križnom entropijom kao rekonstrukcijskom pogreškom te s latentnim prostorom dimenzije 2.

Prvo ćemo vizualno usporediti latentne prostore i njihove distribucije dobivene iz enkodera svakog od modela.

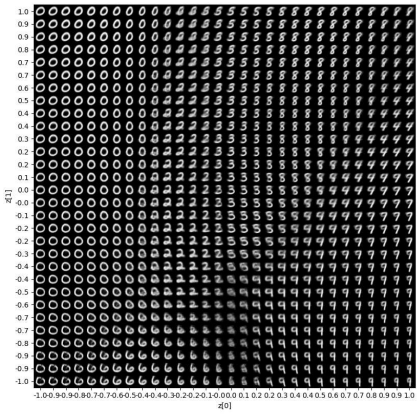


(a) VAE model s potpuno povezanim slojevima.

(b) VAE model s konvolucijskim slojevima.

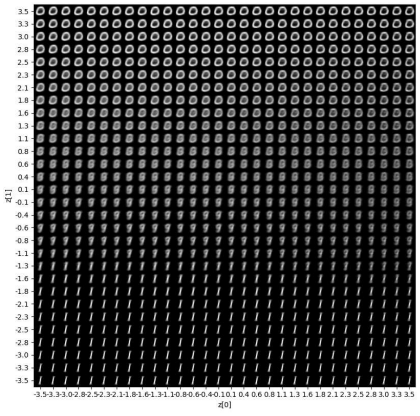
Slika 5.1: Usporedba distribucije latentnih vektora dobivenih iz enkodera različitih arhitektura. Skup podataka nad kojim su trenirani modeli je MNIST, a dimenzionalnost latentnog prostora je 2.

Već na sami prvi pogled možemo primijetiti razlike između distribucija latentnih vektora unutar latentnog prostora. Iako oba modela kao posterior distribuciju latentnih vektora uspijevaju prikazati kao normalnu distribuciju, bolje rezultate pri razdvajanju različitih kategorija slika pokazuje VAE model s potpuno povezanim slojevima, no možemo uočiti da se i kod modela s konvolucijskim slojem uspijevaju odvojiti latentni vektori za određene kategorije znamenki. Isti zaključak možemo donijeti i na temelju slika koje su rekonstruirane za različite kombinacije latentnih vektora iz latentnog prostora ako uzimamo da je distribucija $p(z)$ uniformna i dimenzionalnost latentnog prostora jednaka 2.



(a) VAE model s potpuno povezanim slojevima.

.



(b) VAE model s konvolucijskim slojevima.

Slika 5.2: Usporedba generiranih slika iz latentnog prostora dobivenih iz dekodera različitih arhitektura.

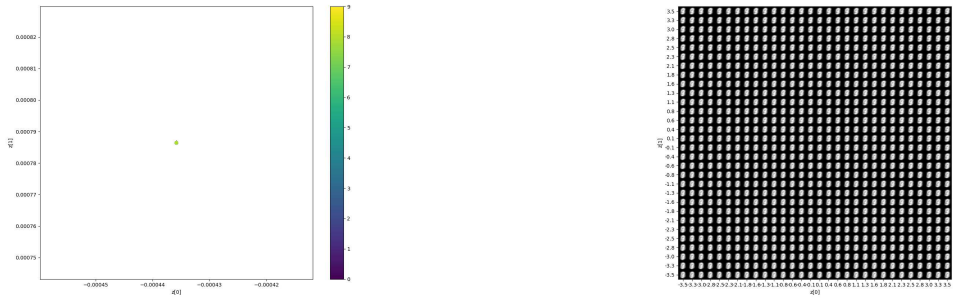
Razlog ovakvih rezultata može biti što su podaci nad kojima su se trenirali modeli jednostavni i ne zahtijevaju složenije arhitekture modela za dobre rezultate, tj. modeli s konvolucijskim slojevima su prekompleksni za ovakav tip podataka i mogu težiti prenaučenosti.

5.1.2. Usporedba različitih rekonstrukcijskih funkcija pogreške

U ovom istraživanju uspoređujemo različite rekonstrukcijske funkcije pogreške koje se koriste u VAE modelima. Funkcije pogreške koje smo usporedili su binarna križna entropija (eng. *binary cross entropy*) i srednja kvadratna pogreška (eng. *mean squared error* - MSE). VAE modeli koje smo trenirali su imali jednake arhitekture navedene u poglavlju 4.3 u oba slučaja.

Teorijski gledano, sličnost između ove dvije vrste rekonstrukcijskih pogrešaka postoji kao što je pokazano u 2.3. No, s obzirom da se pretpostavke takvog izvoda često ne mogu ispuniti u praksi, odlučili smo ispitati razlike u stvarnoj primjeni.

Prilikom treniranja, modeli trenirani s rekonstrukcijskom pogreškom MSE su često imali problema s kolapsom posteriora, tj. utjecaj regularizacije je bio uvelike jači od utjecaja rekonstrukcijske pogreške, tako da je distribucija latentnih vektora bila u potpunosti jednaka normalnoj distribuciji, pri čemu smo izgubili mogućnost razlikovanja različitih kategorija slika.

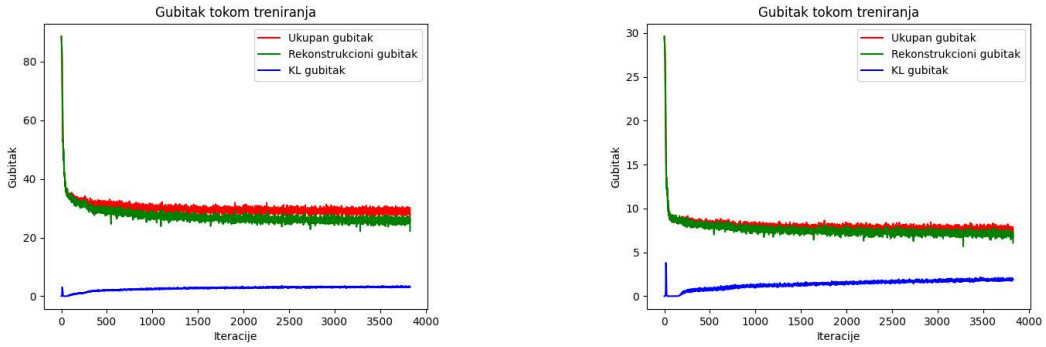


(a) Distribucija latentnih vektora.

(b) Rekonstrukcija slika.

Slika 5.3: Dobiveni prikazi nakon kolapsa posteriora tokom učenja modela.

Možemo uočiti kako naučeni model svaku sliku enkodira u središte prostora, a time dekoder ne može naučiti razlikovati različite kategorije slika. Kako bi se to izbjeglo, u modelu s rekonstrukcijskom pogreškom MSE smo regulirali utjecaj rekonstrukcijske i regularizacijske pogreške kako bi njihov utjecaj bio jednak na učenje distribucije $p(x|z)$ s obzirom na apriornu distribuciju.



(a) Binarna križna entropija.

(b) Regularizirani MSE.

.

Slika 5.4: Prikaz gubitaka modela kroz iteracije učenja.

Iz prikaza 5.4 možemo uočiti kako je gubitak modela s regulariziranom rekonstrukcijskom pogreškom MSE znatno manji od gubitka modela s binarnom križnom entropijom kao funkcija rekonstrukcijskog gubitka. Također možemo uočiti kako u oba slučaja regularizacijski gubitak znatno manji od rekonstrukcijskog, te se nakon određenog broja iteracija povećava. Intuitivno to možemo zamisliti kao da se distribucija latentnih vektora tokom učenja modela širi kako bi se mogli smjestiti svi latentni vektori svake slike i njene kategorije. Izuzevši razlike u mjeri gubitka, razlike u rekonstrukciji samih slika su zanemarive, što je i očekivano s obzirom na teorijsku sličnost ovih dviju rekonstrukcijskih funkcija pogreške. U oba slučaja rekonstrukcija slika te distribucija latentnog prostora izgledaju kao određena varijacija onih prikazanih u 5.1a i 5.2a.

5.2. Istraživanja nad VQ-VAE modelima

U ovom poglavlju ćemo ispitati svojstva otpornosti VQ-VAE modela na manipulacije latentnih vektora. Ispitivanje će biti provedeno samo na područje rekonstruiranja slika, jer ispitivanje svojstava generiranja slika zahtjeva učenje nekog od autoregresivnih modela za definiranje apriorne distribucije željenih generiranih slika.

Također, kao što je spomenuto u poglavlju 4.3, ispitivanje će biti provedeno na jednom jednostavnijem modelu definiranom u sklopu rada te na modelu koji se koristi kao komponenta u popularnom modelu za prevođenje teksta u slike - DALL-E.

5.2.1. VQ-VAE modeli trenirani na MNIST skupu podataka

U sklopu rada napravljen je jednostavniji VQ-VAE model koji je treniran na MNIST skupu podataka, a arhitektura modela opisana je u poglavlju 4.3 pri čemu je dimenzionalnost latentnog prostora 16, a broj vektora unutar vizualnog rječnika 64.

Htjeli bismo ispitati utjecaj manipulacije latentnih vektora na rekonstrukciju slike. Isto tako bi htjeli predočiti kako izgleda kodirani latentni vektor dobiven iz enkodera i kvantizatora. Nad latentnim vektorima primjenit ćemo sljedeće manipulacije:

- Pomicanje vektora po osima.
- Rotacija za 90 stupnjeva.
- Inverzija dimenzija.
- Kombinacija latentnih vektora dvaju slika.
- Inverzija vrijednosti.

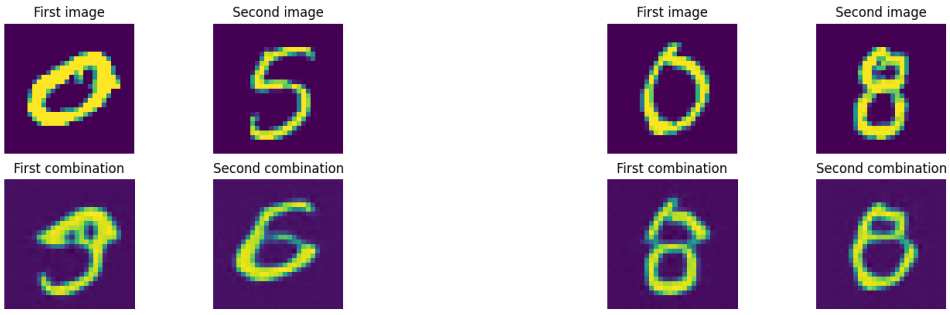
Slika 5.5: Prikaz originalne slike, njenog koda dobivenog iz latentnog vektora, rekonstrukcije te koda nastalog pomakom vektora po y osi.

Slika 5.6: Prikaz originalne slike, njenog koda dobivenog iz latentnog vektora, rekonstrukcije te koda nastalog pomakom vektora po x osi.

Slika 5.7: Prikaz originalne slike, njenog koda dobivenog iz latentnog vektora te rekonstrukcije i koda nastalog rotacijom latentnog vektora za 90 stupnjeva.

Slika 5.8: Prikaz originalne slike, njenog koda dobivenog iz latentnog vektora te rekonstrukcije i koda nastalog inverzijom latentnog vektora po osi x.

Slika 5.9: Prikaz originalne slike, njenog koda dobivenog iz latentnog vektora te rekonstrukcije i koda nastalog inverzijom latentnog vektora po osi y.



Slika 5.10: Prikaz originalnih slika te rekonstrukcija nastalih spajanjem prepolovljenih latentnog vektora po osi x.

Slika 5.11: Prikaz originalnih slika te rekonstrukcija nastalih spajanjem prepolovljenih latentnog vektora po osi y.

Slika 5.12: Prikaz originalne slike, njenog koda dobivenog iz latentnog vektora te rekonstrukcije i koda nastalog inverzijom vrijednosti latentnog vektora.

Kao što je iz priloženog uočljivo, model je otporan na translacije po osima, dok se rotacijom i zrcaljenjem potpuno gubi informacija o slici. Može se primjetiti kako ni inverzija latentnog vektora ne daje dobre rezultate, no za razliku od rotacije, moguće

je uočiti neke od karakteristika originalne slike, što pokazuje da je model uspio naučiti neke od karakteristika slike. Također, moguće je uočiti kako spajanje latentnih vektora po osima daje dobre rezultate, u smislu da rekonstruirana slika izgleda potpuno identično kao da smo spojili dve polovice svake od ulazne slike, što je i očekivano, jer se radi o jednostavnoj operaciji koja ne mijenja strukturu slike. Naime, zanimljivo je da je inverzija vrijednosti latentnog vektora uzrokovala da su se vrijednosti rekonstruirane slike također invertirale, što može donijeti zanimljive zaključke o izgledu latentnog prostora, no to je izvan opsega ovog rada.

5.2.2. Predtrenirani dVAE model

Prvo ćemo ispitati svojstva otpornosti manipulacija nad latentnim vektorima za ponuđeni dVAE model pribavljen iz radova [3] i [7]. Ovaj pretrenirani model je učen na svim mogućim, javno dostupnim slikama na internetu, tako da ćemo nasumično odabrati dvije slike nad kojima ćemo provesti ispitivanja.

(a) Slika pingvina.

(b) Slika tornja iz Pise.

.

Slika 5.13: Prikaz slika korištene pri izvođenju ispitivanja.

(a) Rekonstruirana slika pingvina.

(b) Rekonstruirana slika tornja iz Pise.

.

Slika 5.14: Prikaz rekonstruiranih slika korištene pri izvođenju ispitivanja.

Možemo primjetiti prema slikama 5.14 kako je rekonstrukcija slike pingvina znatno bolja od rekonstrukcije slike tornja u Pisi, što je uzrok mnogih sitnih detalja same te slike koji se izgube tokom rekonstrukcije.

Nad latentnim vektofrima ovih slika primjenit ćemo sljedeće manipulacije:

- Rotacija za 90 stupnjeva.
- Inverzija dimenzija.
- Pomicanje slike po osima.

Iz dobivenih rezultata možemo primjetiti kako na rekonstrukciju slike najviše utječe inverzija latentnog vektora, nakon čega je rekonstrukcija u potpunosti neprepoznatljiva. Zatim dolazi rotacija, nakon koje se slika može prepoznati, ali je i dalje znatno izmjenjena i vidljiv je znatni gubitak kvalitete. Na kraju dolazi pomak latentnog vektora, nakon čega se slika može prepoznati, a promjene su gotovo neuočljive s obzirom na početne rekonstrukcije. Za razliku od prehodno definiranog jednostavnijeg modela, ovaj model je znatno otporniji na manipulaciju latentnog prostora (pogotovo u slučaju rotacije latentnog vektora), s obzirom da je latentni prostor predtreniranog modela bolje strukturiran i sadrži više informacija o samoj slici, a to je sve rezultat veće i kompliciranije arhitekture.

(a) Rekonstruirana slika pingvina.

(b) Rekonstruirana slika tornja iz Pise.

.

Slika 5.15: Prikaz rekonstruiranih slika nakon rotacije latentnog vektora.

(a) Rekonstruirana slika pingvina.

(b) Rekonstruirana slika tornja iz Pise.

.

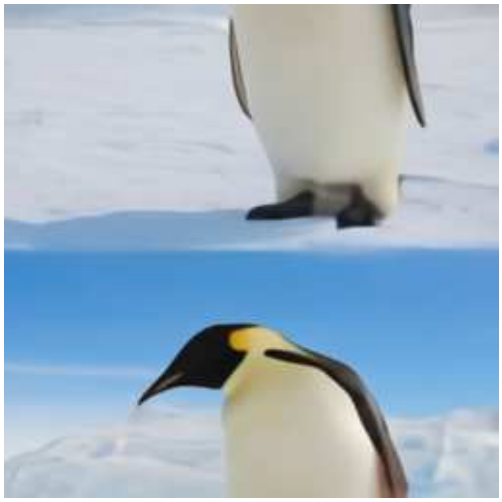
Slika 5.16: Prikaz rekonstruiranih slika nakon inverzije latentnog vektora.

(a) Rekonstruirana slika pingvina.

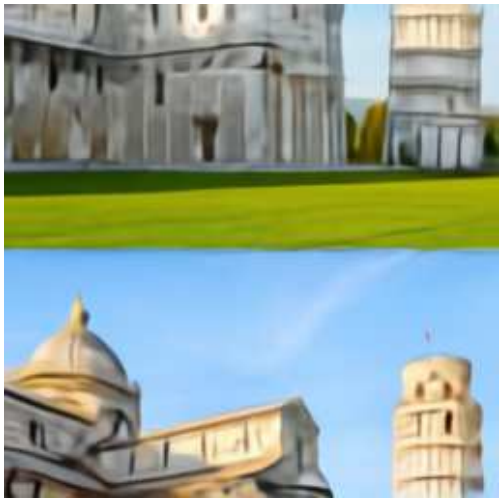
(b) Rekonstruirana slika tornja iz Pise.

.

Slika 5.17: Prikaz rekonstruiranih slika nakon pomaka latentnog vektora po osi x.



(a) Rekonstruirana slika pingvina.



(b) Rekonstruirana slika tornja iz Pise.

.

Slika 5.18: Prikaz rekonstruiranih slika nakon pomaka latentnog vektora po osi y.

Za kraj, prikazat ćemo i premještanje dijela latentnog vektora jedne slike u drugu, što je također jedna od mogućnosti manipulacije latentnog prostora. U sklopu ovog pokusa iskoristit ćemo sliku pingvina iz prijašnjih primjera i sliku polarnog medvjeda.

(a) Slika polarnog medvjeda.

(b) Rekonstruirana slika pingvina s glavnom medvjeda.

.

Slika 5.19: Prikaz rekonstruirane slike nakon premještanja dijela latentnog vektora jedne slike u drugu.

Iz prikaza 5.19 možemo ponovno uočiti da je model, unatoč svim manipulacijama latentnog vektora, rekonstruirao sliku jednaku onoj koju bismo dobili da smo slici pingvina zalijepili glavu polarnog medvjeda. Ovo je još jedan dokaz da je latentni prostor ovog modela dobro strukturiran i sadrži informacije o slici. Svi ovi rezultati ukazuju na to da ovakvi modeli prikladni i za uređivanje slika, a ne samo za rekonstrukciju.

6. Zaključak

U ovom radu smo istražili VQ-VAE modele, a pritom i uveli koncepte varijacijskih autoenkodera koji su generativni modeli temeljeni na neuronskim mrežama. Cilj nam je bio razumjeti koncepte iza ovih modela, njihovu matematičku pozadinu i primjene u generativnom modeliranju, a potom i ispitati njihove sposobnosti i svojstva.

VAE modeli su se pokazali kao moćan alat za rekonstruiranje starih i generiranje novih podataka. Njihova sposobnost naučavanja latentne reprezentacije podataka omogućuje generiranje novih primjera koji su slični onima u skupu podataka, ali nisu potpuno isti. Također, VAE modeli omogućuju efikasno pretraživanje latentnog prostora, što pruža mogućnosti za manipulaciju i kreiranje različitih svojstava generiranih podataka.

VQ-VAE modele rješava taj problem, uvodeći kvantiziranu latentnu reprezentaciju. Ova dodatna komponenta doprinosi generiranju podataka s jasnijim strukturama i sličnijim stvarnim primjerima. VQ-VAE modeli pružaju mogućnost diskretnog predstavljanja latentnog prostora, što omogućuje efikasno pretraživanje i manipulaciju latentnim prostorom. Također pružaju mogućnost učenja priorne distribucije latentnog prostora, što omogućuje generiranje podataka izvan skupa podataka.

Kroz ovaj rad smo također shvatili da su VAE i VQ-VAE modeli samo dio šireg polja generativnog modeliranja. Također smo zaključili da ovakvi modeli, osima svoje sposobnosti rekonstruiranja i generiranja podataka, pružaju i mogućnost uređivanja i manipuliranja podacima. Iako su ovi modeli pokazali svoju snagu, postoje i drugi napredni pristupi i tehnike koje se mogu istražiti i primijeniti.

Za daljnje istraživanje, moguće je proučiti preostala istraživanja izvedena nad VQ-VAE modelima [10] i [7], te novi rad na temu VQ-VAE-2 [8].

LITERATURA

[1] Matthew N. Bernstein. Variational autoencoders, 2023. URL `https://mbernste.github.io/posts/vae/`.

[2] Carl Doersch. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*, 2016.

[3] Patrick Esser, Robin Rombach, i Björn Ommer. Taming transformers for high-resolution image synthesis. 2020.

[4] Ian Goodfellow, Yoshua Bengio, i Aaron Courville. *Deep learning*. MIT press, 2016.

[5] Diederik P Kingma i Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[6] Stephen Odaibo. Tutorial: Deriving the standard variational autoencoder (vae) loss function. *arXiv preprint arXiv:1907.08956*, 2019.

[7] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, i Ilya Sutskever. Zero-shot text-to-image generation. stranice 8821–8831, 2021.

[8] Ali Razavi, Aaron Van den Oord, i Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019.

[9] Charlie Snell. Understanding vq-vae (dall-e explained pt. 1), 2021. URL `https://mlberkeley.substack.com/p/vq-vae`.

[10] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

Konvolucijski varijacijski autoenkoderi s kvantiziranom latentnom reprezentacijom

Sažetak

Zadatak ovog rada je bio uvod u osnovne pojmove i koncepte VQ-VAE modela te uspostaviti vezu s već definiranim varijacijskim autoenkoderima. Uz to, cilj je bio predstaviti osnovne matematičke izvode za definiranje VQ-VAE modela, kao što su varijacijska inferencija, rekonstrukcijski gubitak, regularizacijski gubitak i gubitak kvantizacije.

Zatim je provedeno izvođenje varijacijskih autoenkodera i VQ-VAE modele te njihovo treniranje na jednostavnijem skupu slika (konkretno, MNIST slika znamenki) kako bi se istražila njihova sposobnost rekonstrukcije slika. Također je bilo potrebno odabrati određenu javno dostupnu prednaučenu arhitekturu i provesti ispitivanje nad njom.

Potom su prikazani dobiveni rezultati provedenih ispitivanja te su izvedeni zaključci o performansama ispitanih arhitektura.

Ključne riječi: VQ-VAE, varijacijski autoenkoderi, generativni modeli, varijacijska inferencija, rekonstrukcijski gubitak, kvantizacija.

Vector Quantized Convolutional Variational Autoencoders

Abstract

This study aimed to provide an introduction to the fundamental concepts and principles of the VQ-VAE model and establish a connection with the already defined variational autoencoders. Additionally, the goal was to present the basic mathematical derivations for defining the VQ-VAE model, such as variational inference, reconstruction loss, regularization loss, and quantization loss.

Furthermore, variational autoencoders and VQ-VAE models were derived and trained on a more specific dataset of images (specifically, MNIST digit images) to explore their ability to reconstruct images. It was also necessary to select a specific publicly available pre-trained architecture and evaluate it.

Subsequently, the obtained results of the conducted evaluations were presented, and conclusions were drawn regarding the performance of the examined architectures.

Keywords: VQ-VAE, variational autoencoders, generative models, variational inference, reconstruction loss, quantization.