Treniranje "tekst u sliku" modela strojnog učenja

Strakoš, Leon Ivan

Sveučilište J. J. Strossmayera u Osijeku

Filozofski fakultet Osijek

Prijediplomski studij informatologije

Leon Ivan Strakoš

Treniranje „tekst u sliku“ modela strojnog učenja

Završni rad

Mentor: Izv. prof. dr. sc. Tomislav Jakopec

Osijek, 2024

Sveučilište Josipa Jurja Strossmayera u Osijeku

Filozofski fakultet Osijek

Odsjek za informacijske znanosti

Prijediplomski studij informatologije

Leon Ivan Strakoš

Treniranje „tekst u sliku“ modela strojnog učenja

Završni rad

Društvene znanosti, informacijske i komunikacijske znanosti,

informacijski sustavi i informatologija

Mentor: Izv. prof. dr. sc. Tomislav Jakopec

Osijek, 2024

IZJAVA

Izjavljujem s punom materijalnom i moralnom odgovornošću da sam ovaj rad samostalno napisao/napisala te da u njemu nema kopiranih ili prepisanih dijelova teksta tuđih radova, a da nisu označeni kao citati s navođenjem izvora odakle su preneseni.

Svojim vlastoručnim potpisom potvrđujem da sam suglasan/suglasna da Filozofski fakultet u Osijeku trajno pohrani i javno objavi ovaj moj rad u internetskoj bazi završnih i diplomskih radova knjižnice Filozofskog fakulteta u Osijeku, knjižnice Sveučilišta Josipa Jurja Strossmayera u Osijeku i Nacionalne i sveučilišne knjižnice u Zagrebu.

U Osijeku 9. rujna 2024.

Leon Ivan Strakoš, 0303097804

Ime i prezime studenta, JMBAG

Sažetak: Rad daje pregled aspekata generativnih modela umjetne inteligencije s fokusom na modele pretvorbe teksta u sliku. Objašnjeni su procesi potrebni za generiranje poput enkodiranja teksta, generiranja slika, mehanizma pažnje, preciznog podešavanja i dr. Također, obrađene su arhitekture generativnih modela poput autoenkodera, varijacijskih autoenkodera, transformera, generativnih suparničkih mreža i difuzijskih modela. Objašnjeni su i pojedini procesi i tehnike uključene u treniranje modela i generiranje sadržaja poput mehanizma pažnje, gradijentnog spusta, propagacije unatrag i slično. U radu se posebna pozornost pridaje preciznom podešavanju modela treniranog na slikama Mjeseca i prilagodbu niskog ranga iz konteksta praktične primjene.

Ključne riječi: text to image, fine tunning, generative AI, deep learning, diffusion models

Sadržaj

1. Uvod

Strojno učenje („*Machine Learning“)* je područje proučavanja unutar računalnih znanosti i umjetne inteligencije koje se bavi razvojem i proučavanjem statističkih algoritama koji mogu učiti iz skupova podataka, te obavljati zadatke bez eksplicitnih uputa.[1]

Izraz je 1959. godine skovao Arthur Samuel. Koji je također izumio i najraniji model strojnog učenja za predviđanje rezultata u igri dame. A prije njega koristio se izraz "računala koja se sama uče". Jedna od najčešćih definicija je:

"Računalni program se smatra učenim iz iskustva E *(„experience“)* u odnosu na neku vrstu zadatka T *(„task“)* i mjeru uspješnosti P *(„performance“)* ako se njegova uspješnost u zadacima u T, mjerena P, poboljšava s E." - Tom M. Mitchell [2]

Alternativni način definiranja strojnog učenja dao je Turing s prijedlogom misaonog eksperimenta *"Imitation Game"* u radu *"Computing Machinery and Intelligence"*.[3] Već 1949. postavljeni su temelji neuronskih mreža u knjizi *"The Organization of Behavior"* Donalda Hebba. Osim njega, tu su i logičari poput Waltera Pittsa i Warrena McCullocha, koji su predložili rane matematičke modele neuronskih mreža u svrhu izrade algoritama koji sliče biološkim misaonim procesima. Prvo eksperimentalno računalo Cyberton koje je koristilo oblik strojnog učenja izumila je tadašnja *"Raytheon Company"* na zahtjev američke vlade u svrhu analize sonara, elektrokardiograma i uzoraka govora koristeći podržano učenje. Glavni cilj strojnog učenja je generaliziranje na temelju iskustva. To jest, sposobnost stroja da obavlja nove različite zadatke na kojima je učen određenim skupom podataka. Njegova uloga je klasifikacija podataka na temelju već učenih modela i predviđanje budućih ishoda. [4]

---

[1] IBM. *Strojno učenje*. Preuzeto 4. rujna 2024. s https://www.ibm.com/topics/machine-learning
[2] Turing, A. M. (1950). Computing machinery and intelligence. *Mind, 59*(236), 433–460. https://doi.org/10.1093/mind/LIX.236.433
[3] Awad, M., & Khanna, R. (2015). Machine learning. In *Efficient learning machines*. Apress. https://doi.org/10.1007/978-1-4302-5990-9_1
[4] TechTarget. *A timeline of machine learning history*. Preuzeto 4. rujna 2024. s https://www.techtarget.com/whatis/A-Timeline-of-Machine-Learning-History

Model teksta u sliku je model strojnog učenja koji uzima opis prirodnog jezika, a za rezultat daje sliku koja odgovara tom opisu. Doživjeli su nagli razvoj sredinom prošlog desetljeća za vrijeme "proljeća umjetne inteligencije", koje je uzrokovalo napredak dubokih neuronskih mreža. Modeli teksta u sliku općenito koriste kombinaciju modela jezika *(„language model“)* i generativnog modela *(„generative model“).*

2015. godine u Torontu predstavljen je prvi moderni model teksta u sliku *“alignDRAW”.* Model je nadogradio tadašnje korištenu arhitekturu DRAW (oblik varijacijskog autoenkodera - VAE) tako da ona bude uvjetovana tekstualnim nizom. Verzije tih početnih modela bile su mutne te loše kvalitete i oštrine. Slike su također bile jednostavne i trenirane na skupu podataka kojeg su većina činile tadašnje *"clip art"* slike. Važno je da je model mogao generirati stvari izvan skupa podataka na kojem je model treniran. I raspoznavao nove upite *(„prompts“)* što pokazuje "razumijevanje" podataka iz skupa, umjesto jednostavnog pamćenja.[5]

Pristupi strojnom učenju podijeljeni su u tri paradigme učenja koje se odnose na povratne informacije unutar procesa učenja: [6]

(i) Nadzirano učenje

(ii) Nenadzirano učenje

(iii) Podržano učenje

1.1 Skupovi podataka i etika

Za treniranje modela teksta u sliku potreban je skup podataka (slika) za koje bi bilo poželjno da su tekstualno opisane. Jedan od prvih skupova podataka koji se koristio za modele teksta u sliku je COCO *(„Common Objects in Context“)*, kojeg je objavio Microsoft 2014. godine. COCO se sastoji od više od sto dvadeset tisuća slika, s pet opisa po slici koji su ručno anotirani.[7]

---

[5] Wikipedia. (n.d.). Text-to-image model. Preuzeto 4. rujna 2024. s https://en.wikipedia.org/wiki/Text-to-image_model

[6] Bishop, C. M. (2006). Pattern recognition and machine learning. Microsoft Research. Preuzeto 4. rujna 2024. s https://www.microsoft.com/en-us/research/uploads/prod/2006/01/Bishop-Pattern-Recognition-and-Machine-Learning-2006.pdf

[7] COCO. (n.d.). COCO dataset. Preuzeto 4. rujna 2024. s https://cocodataset.org/#home

Danas korišten skup podataka u vodećim tvrtkama poput Midjourneya i Stable Diffusiona je LAUON-5B, [8] zbirka opisa i poveznica na 2.3 milijarde slika. Podatci skupa prikupljeni su s interneta bez ljudskog nadzora. Uz nekolicinu radnika koji podatke filtriraju. Poznat je problem moderiranja takvih podataka, radi izloženosti traumatičnom sadržaju, slikama nasilja te seksualnog zlostavljanja. 2022. godine Nacionalni centar za nestalu i zlostavljanu djecu identificirao je više od 32 milijuna slika dječje seksualne eksploatacije. Nadalje, prisutan je već dobro istražen problem pristranosti u skupovima podataka poput mizoginije i rasne diskriminacije. Postoje i pitanja autorskih prava ne samo kod prikupljanja podataka već i generiranja sadržaja s upitima i korištenjem imena umjetnika. 2023. godine u SAD-u je predložen zakon za autorska prava koji regulira skupove podataka i generiranje modela teksta u sliku korištenjem imena autora. [9]

Modeli teksta u sliku potencijalno mogu otkriti osjetljive i osobne informacije iz skupova podataka. Osim toga već navedene pristranosti u skupovima podataka koje dovode do mnogih oblika diskriminacija i marginalizacija društvenih skupina. Modeli teksta u sliku mogu se upotrijebiti u svrhu manipulacije i krađe identiteta, širenja lažnih informacija i dr. Modeli bi trebali biti transparentni u pružanju informacija i svojih izvora. Modeli i skupovi podataka moraju biti zakonski regulirani i na njih se trebaju odnositi autorska prava. Međutim, nije moguće kontrolirati učenje modela s vlastitim skupovima podataka poput ovog rada. Na kraju, tu je sam utjecaj na makro-sociologiju, konzumaciju sadržaja i utjecaj na društvene i kulturne norme. Korisnici trebaju imati kontrolu nad generiranjem i diseminacijom kreiranog sadržaja te bi sadržaj korišten u skupovima podataka trebao sam biti kontroliran i odobren od autora i autor bi trebao biti obaviješten o svrsi korištenja. U skupovima podataka postoji recipročan omjer između njegove kontrole i veličine. Međutim, nije nužno da su veći skupovi podataka kvalitetniji (generiraju objektivno slike bolje kvalitete) od pomno odabranih manjih, ni da su oni manje pristrani radi puno duplog sadržaja unutar njih. No, ovaj problem je trenutna posljedica naglog procvata umjetne inteligencije i nespremnosti, te već u skorijoj budućnosti ne bih trebao predstavljati problem . To se odnosi na veće "opće" modele, s obzirom da potencijalni specijalizirani ili vlastiti modeli mogu biti trenirani na mnogo manjim skupovima podataka za određenu namjenu.

[8] LAION. (2022). LAION-5B. Preuzeto 4. rujna 2024. s https://laion.ai/blog/laion-5b/
[9] Kirkpatrick, J. (2023). The state of artificial intelligence in 2023. Congressional Research Service. Preuzeto 4. rujna 2024. s https://crsreports.congress.gov/product/pdf/LSB/LSB10922

Skupovi podataka bi trebali biti reprezentativni, imati načine otkrivanja predrasuda i često provjeravanje. Zbog izazova manipulacije i zlouporabe imati mogućnosti autentifikacije i verifikacije, te je najbitnije razviti svijest o umjetnim medijima. Autori čiji su radovi korišteni u skupovima podataka trebaju biti suglasni i svjesni načina na koji se oni koriste.[10]

2. Generativni modeli

Generativni model jedan je od pristupa u statističkoj klasifikaciji. Oni se razlikuju po načinu računanja klasifikatora što uzrokuje različite stupnjeve statističkog modeliranja. Generativni model statistički je model distribucije zajedničke razdiobe. Osim njega postoji i diskriminativni ili često nazivan uvjetni model, iako se ta dva modela razlikuju. Također postoje klasifikatori koji računaju bez korištenja statističkog modela. [11]

Generativni modeli bave se zajedničkom distribucijom vjerojatnosti. Često prikazano kao $P(x, y)$ ili vjerojatnost dvije nasumične varijable x, y da poprime određene vrijednosti istodobno.

Na primjer varijabla x predstavlja ishod bacanja novčića dok y predstavlja ishod bacanja kocke vrijednostima od 1 do 6. U tom slučaju $P(x, y)$ bi predstavljao vjerojatnost specifičnog ishoda za oba slučaja. [12]

Iz toga slijedi da zajednička distribucija vjerojatnosti mora zadovoljiti određene uvjete, to jest sadrži svojstva:

1. bez negativnosti gdje $P(x, y) \geq$ za sve varijable,
2. normalizacije $\Sigma\Sigma\, P(x, y) = 1$, gdje su sume uzete od svih mogućih vrijednosti varijabli.

Generativni modeli imaju za cilj uhvatiti stvarnu distribuciju klasa u skupu podataka i koriste za zadatke strojnog učenja bez nadzora. Diskriminativni modeli modeliraju granicu odluke

[10] Bostrom, N. (2011). The ethics of artificial intelligence. Preuzeto 4. rujna 2024. s https://www.techpolicy.press/laion5b-stable-diffusion-and-the-original-sin-of-generative-ai/
[11] Jebara, T. (2004). Machine learning: Discriminative and generative. In *The Springer International Series in Engineering and Computer Science*. Kluwer Academic (Springer). https://doi.org/10.1007/978-1-4020-7647-3
[12] Unite.AI. (n.d.). Što je Bayesov teorem? Preuzeto 4. rujna 2024. s https://www.unite.ai/hr/%C5%A1to-je-Bayesov-teorem/

za klase skupova podataka i češće se koriste za nadzirane zadatke strojnog učenja, te su otporniji na ekstreme u vrijednostima.[13]

Dakle, diskriminativni algoritmi pokušavaju izravno naučiti $P(y|x)$ iz podataka, te nakon toga klasificirati podatke. S druge strane, generativni algoritmi pokušavaju naučiti varijable zajedničke distribucije vjerojatnosti i kasnije se mogu pretvoriti u $P(y|x)$ kako bi se klasificirali podatci.[14]

Jedna od prednosti generativnih algoritama je što se mogu koristiti za generiranje novih podataka na temelju postojećih. S druge strane, diskriminativni algoritmi su bolji u klasifikacijskim zadatcima.[15]

Kombinacijom generativnih modela i dubokih neuronskih mreža nastaju Duboki generativni modeli (DGM). Oni koriste arhitekture varijacijskih autoenkodera (VAE), generativnih suparničkih mreža (GAN) i dr. Spomenute arhitekture su najčešće arhitekture korištene u modelima "teksta u sliku" uz difuzijske modele. Dok se za modele jezika koriste auto-regresivni modeli.[16] Shannon je iznio prijedlog generiranja teksta u matematičkoj teoriji komunikacije još 1948.[17]

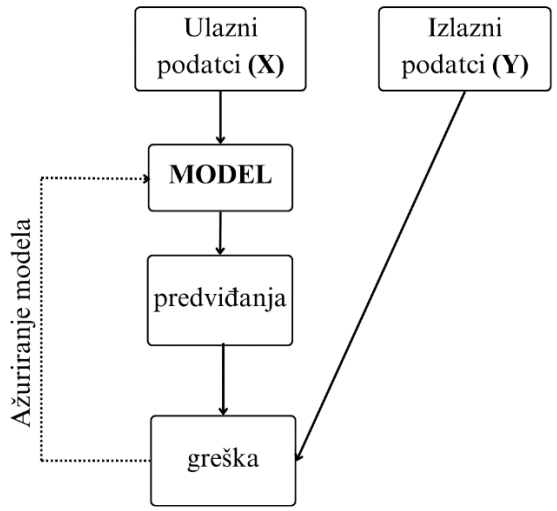
U nadziranom učenju (prikazano na **Slika 1**.) potreban je model koji previđa „klasu" za dani primjer, to jest klasificira ga. Klasifikacija je zapravo ono što čini diskriminativni model nadziranog učenja (primjer **Slika 3**.) Postoje modeli bez izlaznih podataka (kao na

Slika 4.), koji su napravljeni od uzorkovanja ulaznih podataka, te u njima nema korekcije modela, jer model zapravo ništa ne predviđa - što je primjer nenadziranog učenja (prikazano na **Slika 2**.).

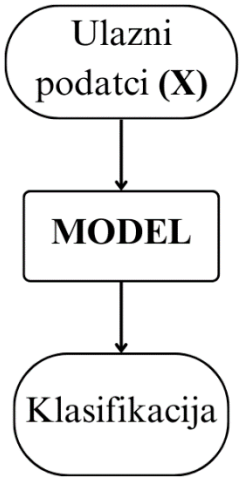
---

[13] Mitchell, T. M. (2015). Generative and discriminative classifiers: Naive Bayes and logistic regression. U *Machine learning*. MIT Press.
[14] Ng, A. Y., & Jordan, M. I. (2002). On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes. *Neural Information Processing Systems (NIPS), 14.*
[15] Unite.AI. (n.d.). Generativni vs. diskriminativni modeli strojnog učenja. Preuzeto 4. rujna 2024. s https://www.unite.ai/hr/generativni-vs-diskriminativni-modeli-strojnog-u%C4%8Denja/
[16] Microsoft Research. (2023). A deep generative model trifecta: Three advances that work towards harnessing large-scale power. Preuzeto 4. rujna 2024. s https://www.microsoft.com/en-us/research/blog/a-deep-generative-model-trifecta-three-advances-that-work-towards-harnessing-large-scale-power/
[17] Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, 27 (July, October), 379–423, 623–656.

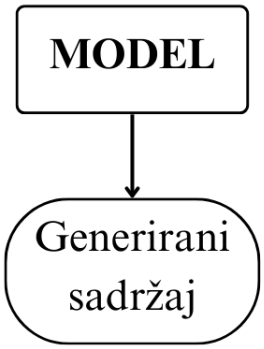
Kod podržanog učenja računalni program unutar dinamičnog okruženja mora izvršiti određeni cilj. Pritom dobiva povratnu informaciju koja je oblik nagrade *("reward based selection"),* koju pokušava maksimizirati.[18]



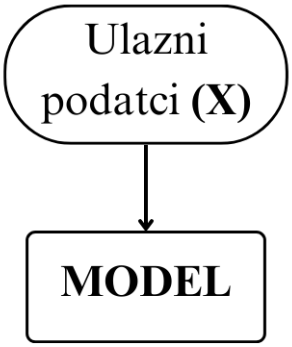
Slika 1. *Prikaz nadziranog učenja*



Slika 2*. Prikaz nenadziranog učenja.*



Slika 3. *Prikaz diskriminativnog modela.*



Slika 4*. Prikaz generativnog modela.*

[18] Machine Learning Mastery. (n.d.). What are generative adversarial networks (GANs)? Preuzeto 4. rujna 2024. s https://machinelearningmastery.com/what-are-generative-adversarial-networks-gans/

3. Autoenkoderi

Autoenkoder je vrsta arhitekture umjetne neuronske mreže koja se koristi za učenje učinkovitog kodiranja neoznačenih podataka, to jest nenadziranog učenja. Cilj je sažimanje ulaznih podataka u latentni prostor nižih dimenzija i njihov pokušaj rekonstrukcije u originalni oblik.[19]

Autoenkoder se sastoji od funkcije enkodiranja koja transformira ulazne podatke i funkciju dekodiranja koja rekreira ulazne podatke iz kodiranog prikaza radi smanjenja dimenzija. Autoenkoder se sastoji od dva skupa koja su većinom euklidski prostori:

1. Z enkodirani prostor
2. X prostor dekodirani prostor

Predstavljeni kao $\mathbb{R}m, \mathbb{R}n$, gdje su m i n dimenzije X i Z.

I dvjema vrstama funkcija:

1. Enkoder $E\phi$ koja preslikava ulazne podatke iz X u Z.
2. Dekoder $D\theta$ koja rekonstruira kodirane podatke iz Z u X.

Definirane su s grčkim phi (ϕ) i theta (θ) koji predstavljaju parametre neuronske mreže enkodera i dekodera koji služe u procesu učenja, redi smanjenja pogrešaka u rekonstrukciji i učinkovito pohranili ulazne podatke u latentni prostor.

Za svaki ulaz (x) koji pripada X, Eϕ preslikava x na odgovarajući izlaz $(z) \in Z$ koji se naziva skrivena (latentna) varijabla, reprenzetacija ili vektor. Dok se $x' = D\theta\ (Z)$ naziva dekodirana poruka.

Važno je da su enkoder i dekoder višeslojni perceptroni (umjetni neuroni). Tako da je jedan njihov sloj $E\phi(x) = \sigma(Wx + b)$. Gdje je sigma σ funkcija aktivacije sigmoidne funkcije, a W matrica zvana težina, te b vektor pristranosti.

[19] Kramer, M. A. (1992). Autoassociative neural networks. *Computers & Chemical Engineering*, 16(4), 313–328. https://doi.org/10.1016/0098-1354(92)80051-A

Zadatak autoenkodera definiran je referentnom distribucijom vjerojatnosti μ_{ref} nad skupom X i funkcijom rekonstrukcije koja uzima kartezijski produkt skupa X, za rezultat interval između nule i beskonačnosti $d: X \times X \rightarrow [0, \infty]$. Tako da se mjeri razlika rekonstukcije x' od ulaza x.

Proces pronalaženja optimalnih parametara autoenkoder modela za zadani zadatak je definiran referentnom distribucijom vjerojatnosti μ_{ref} i funkcijom kvalitete rekonstrukcije d . Cilj je pronaći parametre θ, ϕ koji će smanjiti funkciju gubitka $L(\theta, \phi)$, koja mjeri razliku između ulaznih i rekonstruiranih podataka. Izraz $arg\ min\ \theta, \phi\ L(\theta, \phi)$ označava vrijednosti parametara θ, ϕ koje dovode do najmanjeg gubitka.

Ovaj proces optimizacije, to jest pronalaženja optimalnih parametara, može se provoditi koristeći različite matematičke metode od kojih je najčešća gradijentni spust.[20]

Tako da funkciju gubitka enkodera možemo definirati kao prosjek kvalitete rekonstrukcije svih uzoraka iz μ_{ref} s slijedećim izrazom:

$$L(\theta, \phi) := \ Ex \sim \mu_{ref}\ [d(x, D\theta(E\phi(x)))]$$

Gdje $Ex \sim \mu_{ref}$ predstavlja aritmetičku sredinu funkcije gubitka svih uzoraka x iz distribucije vjerojatnosti. I d funkcije kojoj je rezultat razlika između ulaznih podataka x.

Skup podataka se može prikazati kao $\{x1, \dots, xN\}$ (gdje svaki podatak pripada setu x) i uz pomoć Dirakove delta funkcije dobijemo da svaki podatak unutar skupa ima jednaku težinu u distribuciji.

$$\mu_{ref} = \frac{1}{N} \sum_{i=1}^{N} \delta_{xi}$$

[20] Allen-Zhu, Z., & Orecchia, L. (2014). Linear coupling: An ultimate unification of gradient and mirror descent. *arXiv*. https://doi.org/10.48550/arXiv.1407.1537

Konačno kad imamo Dirakovu mjeru δ_{xi} koja dodjeljuje težinu u prostoru, te d koji mjeri razliku između x i x', to jest kvadriranu euklidsku udaljenost između njih, dobijemo da su optimalni parametri optimizacija najmanjih kvadrata. [21][22]

$$\min_{\phi,\theta} L(\phi,\theta), gdje\ L(\phi,\theta) = \frac{1}{N}\sum_{i=1}^{N} ||\ x_i - D_\theta(E_\phi(x_i))||_2^2$$

3.1 Varijacijski Autoenkoderi

U modelima teksta u sliku češće se koriste varijacijski autoenkoderi (VAE). Oni su vrsta autoenkodera, to jest arhitektura neuronske mreže koju su prvi opisali Kingma i Welling.[23] Pripadaju varijacijskim Bayesovim metodama i probabilističkim grafičkim modelima. [24] Razlikuju se od autoenkodera po tome što neuronske mreže čine samo dio njihove strukture. Varijacijski autoenkoder je generativni model prethodne distribucije („*with a prior probability"*) ili „pretpostavkama" o strukturi ili parametrima prije samih podataka. Ujedno je i model distribucije šuma koji pridodaje nasumičnosti kod generiranja koja bi bila prisutna u stvarnim podatcima. Obično se za njih koriste algoritmi maksimizacije očekivanja (EM). Oni su meta-algoritmi za procjenu parametara u modelima vjerojatnosti, gdje su neke varijable skrivene.

Bayesove metode služe za procjenu nerješivih integrala koje proizlaze unutar Bayesovih inferencija. Koriste se za aproksimaciju aposteriornih vjerojatnosti nepromatranih varijabli u svrhu inferencijalne statistike kao alternativa Monte Carlovim metodama. Te za određivanje donje granice marginalne vjerojatnosti ili dokaza promatranih podataka, najčešće u svrhu odabira modela. Varijacijske Bayesove metode vežu se na EM algoritam, budući da također pronalaze

[21] Wikipedia. (n.d.). Autoencoder. Preuzeto 4. rujna 2024. s https://en.wikipedia.org/wiki/Autoencoder
[22] Analytics Vidhya. (n.d.). Mathematical prerequisites for understanding autoencoders and variational autoencoders (VAEs). Preuzeto 4. rujna 2024. s https://medium.com/analytics-vidhya/mathematical-prerequisites-for-understanding-autoencoders-and-variational-autoencoders-vaes-8f854025390e
[23] Kingma, D. P., & Welling, M. (2013). Auto-Encoding Variational Bayes. *arXiv*. https://doi.org/10.48550/arXiv.1312.6114
[24] Pinheiro Cinelli, L., et al. (2021). Variational autoencoder. U *Variational methods for machine learning with applications to deep networks* (str. 111–149). Springer. https://doi.org/10.1007/978-3-030-70679-1_5

optimalne vrijednosti parametara i imaju istu izmjeničnu strukturu i set jednadžbi koje se ne mogu riješiti analitički.[25]

Cilj je maksimizacija vjerojatnosti podataka „x“ s parametrima distribucije vjerojatnosti $p\Theta(x) = p(\mathrm{x} \mid \theta)$.

U običnom varijacijskom autoenkoderu z je dimenzionalni vektor cijelih brojeva, a $p\theta(x|z)$ normalna distribucija, pa iz toga slijedi: [26]

(i) $p\Theta(z)$ – apriorna vjerojatnost *(prior)*

(ii) $p\Theta(\mathrm{x} \mid z)$ – izglednost (likelihood)

(iii) $p\Theta(\mathrm{z} \mid x)$ – aposteriorna vjerojatnost *(posterior)*[27]

3.1.1 Donja granica dokaza (ELBO)

U varijacijskim autoenkoderima donja granica dokaza („*Evidence Lower Bound“)* je vrijednost koja predstavlja donju granicu log-vjerojatnosti podataka u probabilističkom modelu i u VAE se koristi za optimizaciju theta i phi parametara. Potrebno je odrediti funkciju gubitka da bi se mogli prilagođavati težine mreže kroz propagaciju unatrag. Cilj je optimizirati parametre modela da bi se smanjila greška u rekonstrukciji između izlaznih i ulaznih podataka. Zato se koristi Kullback–Leibler divergencija $D_{KL}(q_{phi}(z \mid \mathrm{x}) \parallel p_{\Theta}(z \mid \mathrm{x})$.[28]

[25] https://en.wikipedia.org/wiki/Variational_Bayesian_methods [pristupljeno 18.8.2024.]
[26] Zhao, S., Song, J., & Ermon, S. (2019). Infovae: Balancing learning and inference in variational autoencoders. U *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 33, str. 5885–5892).
[27] Fer.unizg.hr. (n.d.). *Bayesov klasifikator*. Preuzeto 4. rujna 2024. s https://www.fer.unizg.hr/_download/repository/SU-2016-15-BayesovKlasifikator.pdf
[28]Wikipedia. (n.d.). Evidence lower bound. Preuzeto 4. rujna 2024. s https://en.wikipedia.org/wiki/Evidence_lower_bound

3.1.2 Propagacija unatrag

U VAE prilagodbe težina neuronske mreže radi smanjivanja funkcije gubitka obavlja se pomoću metode propagacije unatrag ili povratnog širenja („*backpropagation"*). [29] To je algoritam nadziranog učenja neuronskih mreža koje koriste gradijentni spust („*gradient descent"*).

Računanje gradijenata odvija se unatrag unutar mreže (zadnji sloj težina se računa prvi). Djelomični izračuni gradijenata koriste se u računanju prethodnih slojeva, što je efikasnije od računanja svakog sloja posebno. Propagacija unatrag koristi se za prepoznavanje slika i govora, te u grafičkim karticama.[30]

Za propagaciju unatrag potrebno je:

(i) skup podataka
(ii) mreža bez povratnih veza *("feed-forward network")*
(iii) funkcija greške

Pojednostavljeni koraci propagacije unatrag: [31]

(i) Računanje „*forward*" faze za svaki ulaz-izlaz par,
(ii) Računanje „backward" faze
(iii) Zbrajanje individualnih gradijenta
(iv) Ažuriranje vrijednosti težina

3.1.3. Reparametrizacija

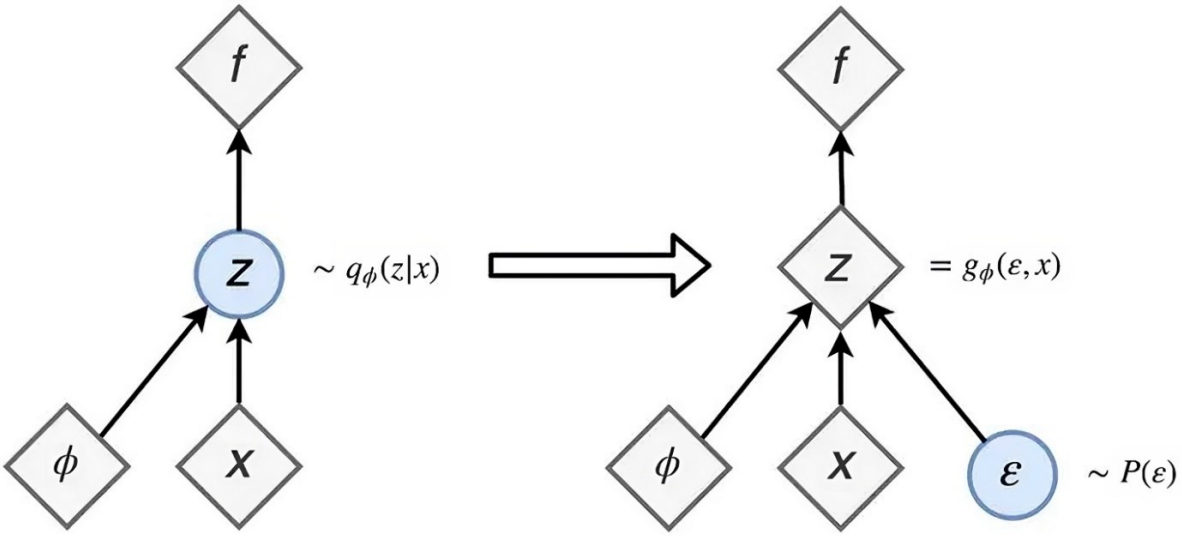
Reparametrizacija ili stohastička propagacija unatrag omogućuje propagaciju unatrag probabilističkim modelima kroz stohastičke procese[32] u neuronskim mrežama, u svrhu optimizacije parametara i računanja gradijenata. Umjesto izravnog uzorkovanja latentne varijable „z" iz

[29] Kabić, S. (2023). *Povratne neuronske mreže i primjene* (Diplomski rad). Zagreb: Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet. Preuzeto s https://urn.nsk.hr/urn:nbn:hr:217:857092
[30] Brilliant.org. (n.d.). Backpropagation. Preuzeto 4. rujna 2024. s https://brilliant.org/wiki/backpropagation
[31] Unite.AI. (n.d.). Što je povratno širenje? Preuzeto 4. rujna 2024. s https://www.unite.ai/hr/%C5%A1to-je-povratno-%C5%A1irenje/
[32] Sveučilište u Zagrebu, Građevinski fakultet. (n.d.). *Stohastički procesi - vježbe*. Preuzeto 4. rujna 2024. s https://www.grad.unizg.hr/_download/repository/Stohasticki_procesi_-_vjezbe.pdf

distribucije, uzima se uzorak iz normalne distribucije. Nasumična varijabla „ε" ubacuje se u latentni prostor z (prikazano i na **Slika 5**.).

$$z = \mu + \sigma \cdot \epsilon$$

Reparametrizacija omogućuje da jednadžba bude diferencijabilna. Također omogućuje korištenje metoda poput stohastičkog gradijentnog spusta (SGD) ili Adam optimizatora[33].[34]



Slika 5. *Prikaz reparametrizacije* [35]

4. Transformer

Transformeri su arhitektura dubokog učenja koju su 2017. predložili Googleovi znanstvenici s radom *„Attention Is All You Need"*. Oni su vrsta neuronske mreže s povratnim vezama gdje izlaz može putovati iz čvorova i utjecati na ulaz istih. Široko su adoptirani u učenju velikih modela jezika (LLM's). Kod transformera tekst je pretvoren u numeričke reprezentacije "*tokene*" ili žetone koji

[33] Cornell University. (n.d.). Adam. Preuzeto 4. rujna 2024. s https://optimization.cbe.cornell.edu/index.php?title=Adam

[34] Fer.unizg.hr. (n.d.). *Logistička regresija*. Preuzeto 4. rujna 2024. s https://www.fer.unizg.hr/_download/repository/SU-2019-06-LogistickaRegresija.pdf

[35] DiningPhil. (n.d.). The reparameterization trick. Preuzeto 4. rujna 2024. s https://diningphil.github.io/project/reparamtrick/

su većinom binarni par, i svaki od njih je onda pretvoren u vektor prema vektorskoj reprezentaciji riječi u tablici *("word embedding table").* [36] Kod svakog sloja, svaki token je kontekstualiziran unutar opsega prozora *("context window")* s drugim nemaskiranim *("unmasked")* tokenima (koji nisu prikriveni ili zamijenjeni tijekom treniranja) pomoću paralelnog mehanizma pažnje s više glava[37] s kojim je signal za određene tokene pojačan ili umanjen za manje vrijedne tokene. Arhitektura je važna radi svoje primjene u raznim područjima procesiranja prirodnog jezika, zvuka i multimedije, te je omogućila razvoj GPT-a [38] i BERT-a.

4.1 Treniranje

U originalnoj arhitekturi postojao je izazov kod konvergencije**,** te je za rješenje predložen stupanj zagrijavanja učenja *„learning rate warmup"* (2% od ukupnih koraka). Kasnije je otkriveno da koristeći sloj normalizacije prije, umjesto poslije mehanizma pažnje s više glava i bez povratnih veza, stabilizira učenje bez korištenja zagrijavanja.[39]

Transformeri su prvo trenirani na samo-nadziranom učenju na velikim skupovima općih podataka. Nakon toga se nadziru i precizno podešavaju na manjim određenim skupovima podataka za odgovarajuće zadatke. [40] Postoje tri razreda *("classes")* zadataka kod modela jezika:

(i) maskirano[41]

Jedan ili više tokena se maskira, a model stvara distribuciju vjerojatnosti za predviđanje maskiranih tokena na temelju konteksta. Funkcija gubitka zbroj je log-nesigurnosti *(„perplexities")* za maskirane tokene.

---

[36] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. U *Advances in Neural Information Processing Systems* (Vol. 30). Curran Associates, Inc. Preuzeto s https://arxiv.org/pdf/1706.03762

[37] Salha, R. (2023). *Transformer arhitektura* (Završni rad). Osijek: Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet primijenjene matematike i informatike. Preuzeto s https://urn.nsk.hr/urn:nbn:hr:126:272269

[38] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., et al. (2020). Transformers: State-of-the-art natural language processing. U *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations* (str. 38–45).

[39] Xiong, R., Yang, Y., He, D., Zheng, K., Zheng, S., Xing, C., Zhang, H., Lan, Y., Wang, L., & Liu, T.-Y. (2020). On layer normalization in the transformer architecture. *arXiv*. https://doi.org/10.48550/arXiv.2002.04745

[40] Wikipedia. (n.d.). Transformer (deep learning architecture). Preuzeto 4. rujna 2024. s https://en.wikipedia.org/wiki/Transformer_(deep_learning_architecture)

[41] Hugging Face. (n.d.). Masked language modeling. Preuzeto 4. runa 2024. s https://huggingface.co/docs/transformers/tasks/masked_language_modeling

(ii) autoregresivno[42]

Cijeli niz je maskiran i model stvara distribuciju vjerojatnosti prema prvom tokenu. Zatim se on otkriva i predviđa sljedeći u nizu, i tako dalje. (GPT modeli). Funkcija gubitka je obično log-vjerojatnost stvarnih tokena u odnosu na predviđene tokene.

(iii) prefixLM[43]

Niz je podijeljen na dva dijela, prvi dio se koristi kao kontekst gdje model predviđa prvi token za drugi dio. Funkcija gubitka ista je kao i za (ii).

4.2 Arhitektura

Arhitektura transformera sastoji se od sljedećih dijelova:

(i) *"Tokenizera"* koji pretvaraju tekst u tokene.

(ii) *"embedding"* sloja koji pretvara tokene u vektorske reprezentacije.

(iii) sloja transformera, koji sadrže pretvorene vektorske reprezentacije i iz njih izvlače lingvističke informacije. Oni se sastoje od slojeva izmjenične pažnje i slojeva bez povratnih veza *"feed-forward"*[44], te enkoder/dekoder slojeva.

(iv) *"un-embedding"* sloj, koji pretvara posljednju vektorsku reprezentaciju nazad u vrijednosnu distribuciju.[45]

Token je cijeli broj koji predstavlja znamenku ili binarni par. Ukupan set tokena je rječnik *(„vocabulary") tokenizera*, kada on naiđe na token izvan rječnika on se tipično označuje kao

[42] Hugging Face. (n.d.). Language modeling. Preuzeto 15. kolovoza 2024. s https://huggingface.co/docs/transformers/tasks/language_modeling

[43] Tay, Y., Dehghani, M., Tran, V. Q., Garcia, X., Wei, J., Wang, X., Chung, H. W., Shakeri, S., & Bahri, D. (2023). UL2: Unifying language learning paradigms. *arXiv*. https://doi.org/10.48550/arXiv.2205.05131

[44] Fer.unizg.hr. (n.d.). *Uvod*. Preuzeto 4. rujna 2024. s https://www.fer.unizg.hr/_download/repository/01-Uvod-1s.pdf

[45] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. U *Advances in Neural Information Processing Systems* (Vol. 30). Curran Associates, Inc. Preuzeto s https://arxiv.org/pdf/1706.03762

„UNK“ za nepoznato. Svaki token je ugrađen *„embedded“* vektor pomoću *„lookup table“*. On ujedno koristi jedan-vruće kodiranje[46] za reprezentaciju tokena u matrici.

Pozicijsko ugrađivanje je vektorska reprezentacija određene veličine relativnog položaja tokena unutar niza. Pruža informacije o mjestu riječi unutar niza. Bez toga model ne bih mogao učinkovito procesuirati riječi u nizu (rečenicama).

Originalno model koristi enkoder i dekoder slojeve koji su neuronske mreže bez povratnih veza za dodatno procesiranje „outputa“ i sadrže normalizacijske korake i većinu parametara modela transformera.

Jednadžba mreže bez povratnih veza u transformeru:

$$FFN(x) = \phi\left(xW^{(1)} + b^{(1)}\right)W^{(2)} + b^{(2)}$$

U kojoj x predstavlja ulaz. W su težinske matrice prvog i drugog sloja koje sadrže parametre, b su vektori pristranosti, a phi ϕ je aktivacijska funkcija poput ReLU, GELU i drugih nelinearnih funkcija.

4.3 Mehanizam pažnje

Pažnja je način na koji model dinamično dodjeljuje težinu (važnost) različitim dijelovima ulaznih podataka. Mehanizam pažnje u transformerima su skalarni umnožak *(„scaled dot-product“)* jedinice pažnje. Za svaku jedinicu model prima tri težinske matrice: težinske vrijednosti upita W^q *(“query weights)*, ključa W^k *(“key weights”)* i vrijednosti W^v *(“value weights”).*

Upiti su stvari koje se pretražuju. U kontekstu transformera oni su izvedeni od dekodera. Ključevi predstavljaju stvari koje se uspoređuju, te dolaze od enkodera. Vrijednosti su informacije koje se koriste ako ključevi odgovaraju određenim ulaznim podatcima.

[46] EITCA. (n.d.). What is one-hot encoding? Preuzeto 4. rujna 2024. s https://hr.eitca.org/artificial-intelligence/eitc-ai-gcml-google-cloud-machine-learning/first-steps-in-machine-learning/plain-and-simple-estimators/what-is-one-hot-encoding/

Svaki upit se uspoređuje sa svim ključevima i rezultati se koriste da bi se dodijelile težinske vrijednosti. Rezultat *("score")* računa se koristeći skalarni umnožak vektora. [47]

$$rezultat(Q, K) = Q * K^{\mathrm{T}}$$

Dobivani rezultati skraćuju se korjenovanjem da bi se izbjegle velike vrijednosti.

$$\text{korjenovan\_rezultat } \textit{(„scaled value“)} = \frac{Q*K^{T}}{\sqrt{d_k}}$$

Normalizirana eksponencijalna funkcija *"softmax"* primjenjuje se na „skalirane“ rezultate. Ona pretvara vektor *"K"* realnih brojeva u distribuciju vrijednosti, koja određuje koliku pažnju treba dobiti pojedinačni ključ. Konačna suma težinskih vrijednosti dobiva se sumom množenja svake vrijednosti s odgovarajućom težinom pažnje.

Transformeri koriste mehanizam pažnje s više glava u kojemu se koriste paralelne glave koja svaka uči na različitih aspektima ulaznih podataka. Podatci su podijeljeni u više glava od kojih svaka ima svoj set upita, ključeva i vrijednosti.[48]

[47] Enciklopedija.hr. (n.d.). Skalarni umnožak. Preuzeto 4. rujna 2024. s https://www.enciklopedija.hr/clanak/skalarni-umnozak

[48] Alammar, J. (n.d.). The illustrated transformer. Preuzeto 4. rujna 2024. s https://jalammar.github.io/illustrated-transformer/

5. Generativne suparničke mreže (GAN)

GAN-ovi su arhitektura strojnog učenja i trenutno najrasprostranjeniji okvir za generativnu umjetnu inteligenciju. Unutar njih dvije neuronske mreže natječu se igri s nultom sumom *(„zero-sum“).* [49]

Originalno model je predložio Ian Goodfellow [50] i suradnici (2014.) kao oblik generativnog nenadziranog učenja. Međutim, pokazali su se korisnim i u polu-nadziranom[51] i podržanom [52] učenju.

GAN-ovi zapravo predstavljaju nenadzirano učenje kao nadzirano i istovremeno koriste generativni i diskriminativni model.

Alec Radford i suradnici su 2015. objavili standardizirani pristup GAN-ovima nazvan DCGAN (*„deep convolutional GAN“)* na kojemu se djelomično temelje današnje arhitekture.[53]

GAN radi na temeljnom principu dva pod-modela u kojemu jedna neuronska mreža ima ulogu „generatora“ koji stvara sadržaj, te drugoga „diskriminatora“ koji procjenjuje autentičnost generiranog sadržaja.[54]

To jest diskriminator klasificira sadržaj pod „stvarno“ ili „lažno“ sve dok ne bude prevaren više od pola puta, u tom slučaju generator je dovoljno treniran da proizvodi vjerodostojne primjere. Stvarni primjeri dolaze iz skupa podataka za treniranje i nakon treniranja diskriminator se u većini slučajeva odbacuje jer više nema ulogu. [55]

---

[49] IEEE Computer Society. (2022). *The rise of generative AI*. Preuzeto 4. rujna 2024. s https://www.computer.org/csdl/magazine/co/2022/10/09903869/1H0G6xvtREk

[50] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. U *Proceedings of the International Conference on Neural Information Processing Systems (NIPS 2014)* (str. 2672–2680).

[51] Isola, P., Zhu, J.-Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial nets. U *Computer Vision and Pattern Recognition*.

[52] Ho, J., & Ermon, S. (2016). Generative adversarial imitation learning. U *Advances in Neural Information Processing Systems* (Vol. 29, str. 4565–4573). https://doi.org/10.48550/arXiv.1606.03476

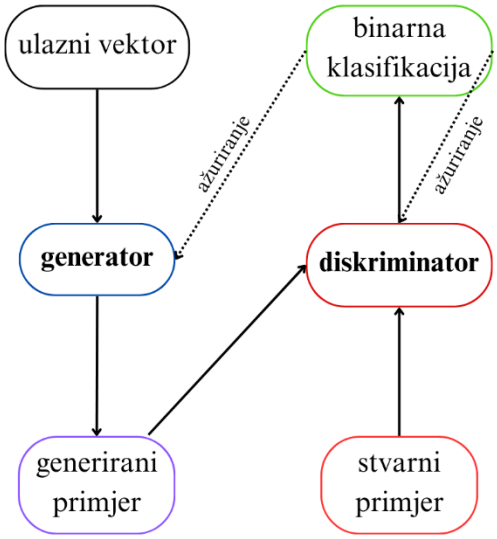
[53] Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks.

[54] IEEE Computer Society. (2022). *The rise of generative AI*. Preuzeto 4. rujna 2024. s https://www.computer.org/csdl/magazine/co/2022/10/09903869/1H0G6xvtREk

[55] Machine Learning Mastery. (n.d.). What are generative adversarial networks (GANs)? Preuzeto 4. rujna 2024. s https://machinelearningmastery.com/what-are-generative-adversarial-networks-gans/

Samostalne metode propagacije unatrag primjenjuju se na obje mreže. U slučaju generiranja slika generator je tipično dekonvolucijalna neuronska mreža, a diskriminator konvolucijalna neuroska mreža.

Diskriminator je nagrađen kada uspješno identificira primjere, dok je generator penaliziran s ažuriranjem parametara modela. I obrnuto, u slučaju kada generator prevari diskriminatora (prikazano na **Slika 6**.).



Slika 6. *Prikaz GAN arhitekture.*

Limit modela je kada generator proizvodi savršene replike ulaznih podataka, te diskriminator ne može predvidjeti uspješno ni jedan primjer (50% za stvarne i lažne). Međutim, u stvarnom primjeru model je dovoljno dobar i prije toga, te za neke upotrebe idealan model koji stvara savršene replike ulaznih podataka nije poželjan.

Kod treniranja s ograničenim skupom strategija standardne metode korištena gradijentnog spusta često nisu primjenjive. Radi nestabilne konvergencije dolazi do urušavanja modusa *(„mode collapse")* gdje pojedini skupovi iz ulaza uopće nisu prisutni u izlaznim podatcima. Također dolazi i do problema nestajućih gradijenata gdje je diskriminator „predobar", u kombinaciji s malim koracima radi gradijentnog spusta dolazi do situacije gdje generator ne može učiti. Da bi

konvergencija bila stabilnija koristi se metoda ažuriranja s dvama vremenskim skalama (TTUR) u kojima je treniranje generatora sporije od diskriminatora .[56]

GAN-ovi se temelje na teoremima Jensen-Shannonove divergencije i jedinstvene ravnotežne točke. Za razliku od *„flow-based“* generativnih modela. GAN-ovi su implicitni generativni modeli koji ne govore eksplicitno vjerojatnost funkcije ni ti je moguće pronaći latentnu varijablu koja odgovara danom uzorku.[57]

GAN-ovi također mogu generirati cijeli uzorak u jednom prolazu kroz mrežu i usporedbi s Boltzmannovim strojevima i linearnom ICA metodom nemaju ograničenje na tip funkcije korištene u mreži.[58]

5.1 Konvolucijske neuronske mreže (CNN)

Radi česte uporabe GAN-ova sa slikovnim podatcima, koriste se konvolucijske neuronske mreže (CNN). One se koriste za najsuvremenije zadatke u području računalnog vida poput prepoznavanja objekata i raspoznavanja lica.

Budući da modeli koristi slikovne podatke i prave njihovu sažetu reprezentaciju, izlazni podatci su lako vidljivi programerima. Mogućnost lake vizualne procjene kvalitete stvorenog sadržaja s CNN-ovima dovela je do ogromnih napredaka u sposobnostima GAN-ova i drugih generativnih modela.

CNN rješava problem težinskih vrijednosti, gdje bi bila potrebna prevelika veličina podataka za postizanje preciznosti. Konvolucijski slojevi imaju sposobnost prepoznavanja obilježja slika na kojima se temelji prepoznavanje apstraktnijih obilježja. Inače bi takva obilježja bilo nemoguće prepoznati na temelju samih piksela.[59]

---

[56]Wikipedia. (n.d.). Generative adversarial network Preuzeto 4. rujna 2024. s https://en.wikipedia.org/wiki/Generative_adversarial_network

[57] Šegvić, S. (2018). *Modeli i algoritmi za strojno učenje*. Preuzeto 4. rujna 2024. s https://www.zemris.fer.hr/~ssegvic/modeli/hrkac18modeli.pdf

[58] Mohamed, S., & Lakshminarayanan, B. (2016). Learning in implicit generative models. *arXiv*. https://doi.org/10.48550/arXiv.1610.03483

[59] Elements of AI. (n.d.). Generative adversarial networks. Preuzeto 4. rujna 2024. s https://course.elementsofai.com/hr/5/3

5.2 Uvjetovani GAN-ovi

Generativni model može biti treniran gdje je nasumični vektor uvjetovan („*conditioned*“) s dodatnim „inputom“. On može biti klasna vrijednost, poput (muško/žensko) za kreiranje slika osoba i dr.

Diskriminator je također uvjetovan da osim stvarno/lažno klasificira i dodatni ulazni podatak. Na taj način uvjetovani GAN može generirati primjere iz domene bilo kojeg danog tipa.

Na tom principu GAN modeli mogu biti uvjetovani da daju primjer iz domene, poput slike. Što omogućuje generiranje teksta u sliku, ili slike u sliku. Te i druge mogućnosti poput kopiranja stila, kolorizacije, i raznih manipulacija slika.[60]

6. Difuzijski modeli (DDPM)

Difuzijski modeli postepeno pretvaraju vizualnu buku u smislene slike. Dosegli su svoju popularnost jer daju više kontrole nad konačnim rezultatom u usporedbi s GAN-ovima. Model se sastoji od tri dijela: procesa unaprijed, procesa unatrag i uzorkovanja. [61] Većinom se treniraju koristeći varijacijsku inferenciju (varijacijske Bayesove metode).[62]

Model koji obavlja uklanjanje šuma *(„denoising“)* standardno se naziva *„backbone“*, i on je većinom U-net ili transformer. Difuzijski modeli počivaju na iznimno kompleksnim raspodjelama vjerojatnosti i koriste tehnike koje potječu iz neravnotežne termodinamike. [63]

Modeli rade na principu dodavanja Gaussovog šuma na ulazne podatke, te koriste to kao polazište od kojeg pokušavaju rekreirati/restaurirati podatke postepenim uklanjanjem šuma. (vidi

---

[60] Machine Learning Mastery. (n.d.). What are generative adversarial networks (GANs)? Preuzeto 4. rujna 2024. s https://machinelearningmastery.com/what-are-generative-adversarial-networks-gans/

[61] Chang, Z., Koulieris, G. A., & Shum, H. P. H. (2023). On the design fundamentals of diffusion models: A survey. *arXiv*. Preuzeto 4. rujna 2024. s https://arxiv.org/abs/2303.06150

[62] Babić, D. (2023). *Varijacijski autoenkoderi s kvantiziranom latentnom reprezentacijom* (Završni rad). Zagreb: Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva. Preuzeto s https://urn.nsk.hr/urn:nbn:hr:168:456628

[63] Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., & Ganguli, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics. U *Proceedings of the 32nd International Conference on Machine Learning* (Vol. 37, str. 2256–2265). PMLR. https://proceedings.mlr.press/v37/sohl-dickstein15.pdf

Izvorni kod 1. Prikaz više razine difuzijskog modela pomoću „denoising_diffusion_pytorch“ paketa.[64]

Izvorni kod 1. *Prikaz više razine difuzijskog modela pomoću „denoising_diffusion_pytorch“ paketa.*

```
$ pip install denoising_diffusion_pytorch
from denoising_diffusion_pytorch import Unet, GaussianDiffusion, Trainer
# Procces unatrag - Unet
model = Unet(
    dim = 64,                          # dimenzionalnost
    dim_mults = (1, 2, 4, 8),          # „scaling factor“
    flash_attn = True                  # „flash attention“
)
# Procces unaprijed - Noise Scheduler
diffusion = GaussianDiffusion(
    model,
    image_size = 128,             # rezolucija slike
    timesteps = 1000,              # broj koraka
    sampling_timesteps = 250     # broj koraka uzorkovanja
)
# Treniranje
trainer = Trainer(
    diffusion,
    'path/to/your/images',            # putanja do podataka
    train_batch_size = 32,            # veličina grupe
    train_lr = 1e-5,                  # faktor učenja
    train_num_steps = 100000,          # ukupni koraci učenja
    gradient_accumulate_every = 2,     # koraci zbrajanja gradijenta
    ema_decay = 0.995,                 # „exponential moving average decay“
    amp = True,                        # „mixed precision“
    calculate_fid = True               # „Fréchet Inception Distance“
)
trainer.train()
# Uzorkovanje „Sampling“
sampled_seq = diffusion.sample(batch_size = 4)
sampled_seq.shape # (4, 32, 128)
```

[64]Lucidrains. (n.d.). Denoising diffusion pytorch. Preuzeto 4. rujna 2024. s https://github.com/lucidrains/denoising-diffusion-pytorch?tab=readme-ov-file

6.1 CLIP (Contrastive Language-Image Pre-training)

CLIP je model napravljen da razumije slike i tekst zajedno u svrhu izvođenja zadataka poput tekstualnih opisa, opisivanja slika *(„image captioning“)* i slično. Glavna prednost CLIP modela je da ne zahtijeva trening za pojedinačan zadatak, već može donositi zaključke za više zadataka, radi treninga na velikom skupu podataka slika i teksta. Model je treniran na 400 milijuna slika i tekstualnih parova ugrađenih u isti prostor.[65]

Sastoji se od enkodera teksta koji koristi arhitekturu transformera i enkodera slike koji koristi arhitekturu CNN-a. Treniran je na kontrastivnom reprezentacijskom učenju kojemu je cilj slične parove slike i teksta ugraditi bliže u prostor, i obrnuto.

Ima velik utjecaj na kreiranje upita modela teksta u sliku i načina na koji se upiti stvaraju (*„prompt engineering“)*, iz tog razloga je bitan za praktični dio ovoga rada. Također je razlog zašto duži upiti poput *„photo of an {object}„* daju bolje rezultate od navođenja samo objekta.[66]

6.2 Precizno podešavanje „fine tuning“

Precizno podešavanje je korištenje već treniranog modela. On se prilagođava na novom skupu podataka za posebnu ulogu, stil ili novi koncept. Ideja je korištenje prethodno treniranog modela koji je često treniran na puno većem skupu podataka te na taj način štedi vrijeme i resurse potrebne na treniranje potpuno novog modela. Najčešće je oblik nadziranog učenja. [67]

Postoje razni načini ubacivanja *„fine tuned“* modela u prethodno trenirane. Precizno podešavanje može se raditi na cijeloj neuronskoj mreži ili samo na određenim slojevima, koji su zamrznuti *(„frozen“)* tijekom procesa, točnije, nisu podređeni propagaciji unatrag.[68] Precizno

---

[65] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., & Sutskever, I. (2021). Learning transferable visual models from natural language supervision. U *International Conference on Machine Learning*. Preuzeto s https://arxiv.org/abs/2103.00020.
[66] Paluchasz, A. (n.d.). Understanding OpenAI's CLIP model. Preuzeto 15. kolovoza 2024. s https://medium.com/@paluchasz/understanding-openais-clip-model-6b52bade3fa3
[67] Amanatulla, M. (n.d.). Fine-tuning the model: What, why, and how. Preuzeto 15. kolovoza 2024. s https://medium.com/@amanatulla1606/fine-tuning-the-model-what-why-and-how-e7fa52bc8ddf
[68] CS231n. (n.d.). Transfer learning. Preuzeto 4. rujna 2024. s https://cs231n.github.io/transfer-learning/

podešavanje potpunog modela većinom daje bolje rezultate, ali zahtjeva puno više vremena i računanja podataka, također, sama veličina modela je veća.

6.2.1. Prilagodba niskog ranga

„Low-rank adaptation“ (LoRA) je metoda preciznog podešavanja koja se temelji na „adapterima“. LoRA stvara matricu niskog ranga koja se kasnije nadodaje originalnoj matrici. Adapter je skup nisko rangiranih matrica koje uz originalne, čine precizno podešen model.[69] LoRA omogućava kvalitetu sličnu potpuno treniranih precizno prilagođenih modela, a zahtjeva puno manje prostora i za razliku od adaptera nemaju dodatno vrijeme čekanja.

LoRA koristi koncept dekompozicije ranga *(„rank decomposition“)* kojim se više-dimenzionalna matrica prikazuje uz pomoć prije manje-dimenzionalne matrice.

Rang matrice najveći je broj linearno neovisnih redova ili stupaca. Linearno neovisni vektori ne mogu biti prikazani kao kombinacija drugih vektora unutar skupa. Struktura niskog ranga podrazumijeva da su matrice poslagane na način da mogu biti pretpostavljeni malim brojem linearno neovisnih stupaca.[70]

Jednadžba dekompozicije može biti prikazana na sljedeći način: (gdje je „r“ rang, težine „W“, „d“ ulazna, a „k“ izlazna dimenzija, te su „B“ i „A“ manje dimenzionalne matrice).[71]

$W_n, W_0, \Delta W \in \mathbb{R}^{dxk}$

$B \in \mathbb{R}^{dxr}$

$A \in \mathbb{R}^{dxr}$

[69] Unite.AI. (n.d.). A full guide to fine-tuning large language models. Preuzeto 4. rujna 2024. s https://www.unite.ai/hr/a-full-guide-to-fine-tuning-large-language-models/
[70] Khadka, P. (n.d.). Low-rank adaptation (LoRA). Preuzeto 4. rujna 2024. s https://medium.com/@pranjalkhadka/low-rank-adaptation-lora-fedf37b92026
[71] GitConnected. (n.d.). LoRA: Low-rank adaptation explained with examples. Preuzeto 4. rujna 2024. s https://levelup.gitconnected.com/lora-low-rank-adaptation-explained-with-examples-fc3e31cfa243

$r \ll \min(d, k)$

$$W_n = W_0 + \Delta W = W_0 + BA$$

Primjer.

d=k=1000, r=2

$W_n = W_0 + \Delta W_{[1000x1000]} =$ 1000000 parametara

$W_n = W_0 + B_{[1000x2]} A_{[2x1000]} =$ 4000 parametara

$\frac{4000}{1000000} = 0.004$ Broj parametara smanjen za 99.6%

7. Istraživanje

U ovom radu korištena je LoRA tehnika za precizno podešavanje modela. Prethodno trenirani model kojeg se precizno podešavalo je Stable diffusion XL (SDXL). [72] Stable diffusion model koristi varijacijske enkodere, CLIP za razumijevanje upita, te je arhitektura koja počiva na transformerima, točnije U-Net s mehanizmima pažnje.

Radi hardverskih ograničenja korištene su grafičke kartice preko platforme u oblaku RunPod.io. Korišten je gpu (A40) od 48 gb virtualne memorije, 50gb RAM-a i 9 virtualnih procesora. Za treniranje bi najmanje 24 gb VRAM-a bilo potrebno. Konfiguracija se radila unutar Jupyter bilježnice. Programske bilježnice *(„notebook“)* su virtualna okruženja koja sadržavaju programski kod, tekst ili multimediju. Sadržaj unutar njih je podijeljen u ćelije, koje se mogu izvoditi neovisno o drugima.[73]

Pomoću Jupyter bilježnice postavljena su virtualna okruženja i instalirane zavisnosti (*“dependencies”*) poput PyTorch, diffusers, LoRA libraries i dr.. Korištena okruženja su Stable

[72] Stability AI. (n.d.). *Stable diffusion XL base 1.0*. Preuzeto 4. rujna 2024. s https://huggingface.co/stabilityai/stable-diffusion-xl-base-1.0/resolve/main/sd_xl_base_1.0.safetensors?download=true

[73] Alković, G. (2018). *Proširenja Jupyter notebook razvojne okoline* (Diplomski rad). Zagreb: Sveučilište u Zagrebu, Fakultet organizacije i informatike. Preuzeto s https://urn.nsk.hr/urn:nbn:hr:211:987871

Diffusion i automatic1111 WebUI sučelje (za učitavanje modela i generiranje), te KohyaSS GUI web sučelje (za treniranje *"fine tuned"* modela). Model je treniran na slikama Mjeseca. Skup podataka koji se koristio za treniranje preuzet je s platforme Kaggle i sadrži oko 1300 slika.[74] Parametri treniranja prikazani ispod na **Izvorni kod 2**. Prikaz trening parametara iz config.toml datoteke. i **Izvorni kod 3**. Isječak iz „kohya_ss.log" datoteke.

7.1 Parametri treniranja

***Izvorni kod 2**. Prikaz trening parametara iz config.toml datoteke.*

```
bucket_no_upscale = true
bucket_reso_steps = 64
cache_latents = true
caption_extension = ".txt"
clip_skip = 1
dynamo_backend = "no"
enable_bucket = true
epoch = 20
gradient_accumulation_steps = 1
huber_c = 0.1
huber_schedule = "snr"
learning_rate = 0.0001
loss_type = "l2"
lr_scheduler = "constant"
lr_scheduler_args = []
lr_scheduler_num_cycles = 1
lr_scheduler_power = 1
max_bucket_reso = 2048
max_data_loader_n_workers = 0
max_grad_norm = 1
max_timestep = 1000
max_token_length = 75
max_train_steps = 1600
min_bucket_reso = 256
mixed_precision = "fp16"
multires_noise_discount = 0.3
no_half_vae = true
noise_offset_type = "Original"
optimizer_args = []
optimizer_type = "Adafactor"
pretrained_model_name_or_path =
"stabilityai/stable-diffusion-xl-
base-1.0"
output_dir = "/workspace/model"
output_name = "mjesec"
train_data_dir =
"/workspace/img/moon/"
sample_prompts =
"/workspace/model/prompt.txt"
prior_loss_weight = 1
resolution = "1024,1024"
sample_sampler = "euler_a"
save_every_n_epochs = 1
save_model_as = "safetensors"
save_precision = "fp16"
text_encoder_lr = 0.0001
train_batch_size = 1
unet_lr = 0.0001
xformers = true
network_dim = 32
network_module = "networks.lora"
network_args = []
network_alpha = 8
```

[74] Kaggle. (n.d.). *Sun and moon images*. Preuzeto 4. rujna 2024. s https://www.kaggle.com/datasets/khushipitroda/sun-and-moon-images?resource=download

***Izvorni kod 3**. Isječak iz „kohya_ss.log“ datoteke.*

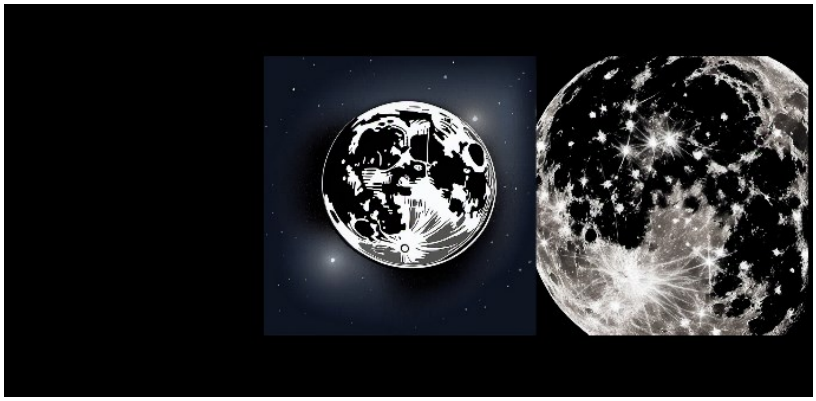
```
INFO     Kohya_ss GUI version: v24.1.6
INFO     Submodule initialized and updated.
INFO     nVidia toolkit detected
INFO     Torch 2.1.2+cu121
INFO     Torch backend: nVidia CUDA 12.1 cuDNN 8902
INFO     Python version is 3.10.12

INFO     create LoRA network. base dim (rank):   lora.py:928
                 32, alpha: 8
INFO     neuron dropout: p=None, rank dropout:   lora.py:929
                           p=None, module dropout: p=None
INFO     create LoRA for Text Encoder 1:         lora.py:1020
INFO     create LoRA for Text Encoder 2:         lora.py:1020
INFO     create LoRA for Text Encoder: 264       lora.py:1028
                           modules.
INFO     create LoRA for U-Net: 722 modules.     lora.py:1036
INFO     enable LoRA for text encoder: 264       lora.py:1077
                           modules
INFO     enable LoRA for U-Net: 722 modules      lora.py:1082

                    prepare optimizer, data loader etc.
INFO     use Adafactor optimizer |               train_util.py:4485
                           {'relative_step': True}
INFO     relative_step is true /                 train_util.py:4488
                           relative_stepがtrueです
WARNING  learning rate is used as                train_util.py:4490
                           initial_lr / 指定したlearning
                           rateはinitial_lrとして使用されま
                           す
WARNING  unet_lr and text_encoder_lr are         train_util.py:4502
                           ignored /
                           unet_lrとtext_encoder_lrは無視さ
                           れます
INFO     use adafactor_scheduler /               train_util.py:4507
                           スケジューラにadafactor_schedule
                           rを使用します
```

```
running training / 学習開始
num train images * repeats : 1344
num reg images: 0
num batches per epoch : 1344
num epochs : 2
batch size per device : 1
gradient accumulation steps  = 1
total optimization steps : 1600
```

7.2 Rezultati

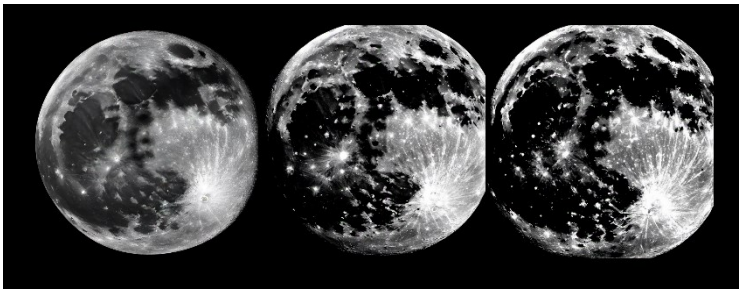
Za treniranje je korištena samo jedna epoha radi jednostavnog skupa podataka te da bi se izbjegao *„overfitting“* modela. Ukupno vrijeme treniranja za jednu epohu sa 1600 koraka bilo je 33:03 minute (1.23s/it), s prosječnim gubitkom od 0.0931.

***Slika** 7. Usporedba modela s kratkim upitom.*



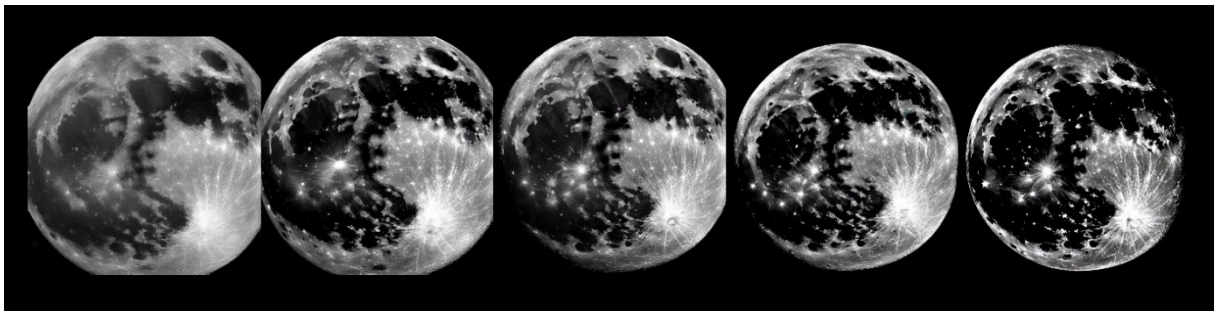
***Slika 8.** Usporedba modela na dužem upitu.*

Koristeći iste upite *„moon“* i *„moon {Lora},* precizno podešen model daje vidljivo jasne rezultate na nasumično korišten *„seed“* (**Slika 7**.). Također, koristeći preciznije duže upite i isti *„seed“*, obje generirane slike daju bolje rezultate (**Slika 8**). Međutim, novi model ponovno proizvodi detaljnije i stvarnije slike Mjeseca.



Slika 8. *Usporedba koraka uzorkovanja*

Testirajući i uspoređujući model na različitim parametrima, vidljivo je, da je jedna epoha u ovom slučaju bila sasvim dovoljna. Parametar koraka uzorkovanja (*„sampling steps“)* daje detaljnije slike proporcionalno s koracima (prikazano na ***Slika 8.***).

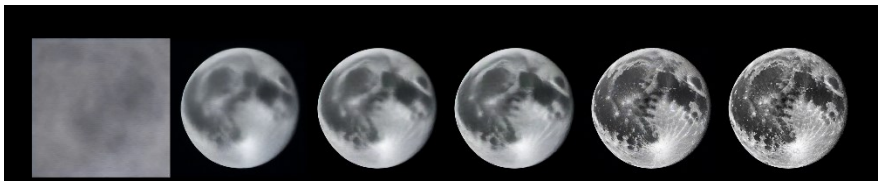


Slika 9. *Usporedba parametra „CFG scale“.*

Prikaz utjecaja CFG *(„classifier-free guidance“*) skale na generirane rezultate (**Slika 9**.). CFG skala utječe na težine modela i praćenje upita. Može se koristiti kao dobar pokazatelj prekomjernog opremanja („overfitting“)[75] modela. U tom slučaju algoritam je previše blizak ulaznim podatcima, te naučeni model previše sliči ili se uopće ne razlikuje od ulaznih podataka, radi loše generalizacije i prevelikog pamćenja ulaznih podataka. CGF skala govori modelu koliko da prati upit, što utječe na kreativnost generiranih podataka. Nakon neke razine generirani podatci

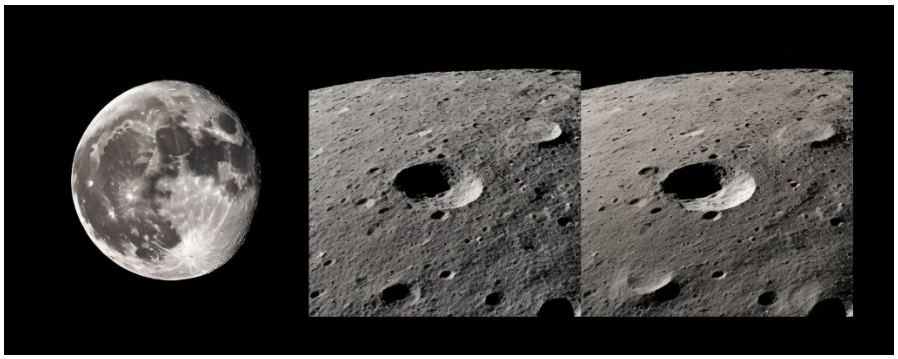
[75] EITCA. (n.d.). How can overfitting be visualized in terms of training and validation loss? Preuzeto 4. rujna 2024. s https://hr.eitca.org/artificial-intelligence/eitc-ai-tff-tensorflow-fundamentals/overfitting-and-underfitting-problems/solving-models-overfitting-and-underfitting-problems-part-1/examination-review-solving-models-overfitting-and-underfitting-problems-part-1/how-can-overfitting-be-visualized-in-terms-of-training-and-validation-loss/

počinju biti niže kvalitete. Kao što je iz primjera vidljivo CFG skala utječe i na zasićenost slike, pa su slike prevelike razine po tome prepoznatljive.



***Slika 10**. Primjer procesa uklanjanja šuma.*

Jedan pokazatelj prekomjernog opremanja je ukoliko model daje slične rezultate bez obzira na upit. Kako bi se to testiralo, u upitu se mogu podešavati težine LoRA modela (**Slika 11**).



***Slika 11**. Usporedba težina modela*

.

U ovom primjeru, između ostaloga, korišteni su i upiti *„moon surface*“, „moon's orbit“, *„ground“, „rock“* i slično. Koristeći potpune težine modela, u prvoj slici (lijevo) model je u potpunosti ignorirao ostale upite. Ali oni se već pojavljuju smanjujući težine na 0.8. Iz toga model je *„overfitted“* treniran i na samo jednoj epohi radi jednostavnosti koncepta. Unatoč tome, čini se da model daje detaljnije rezultate Mjesečeve površine (od SDXL modela), ali teško je izvoditi zaključke iz par primjera.

***Slika 12**. Prikaz 0.6 slike (desno) iz prethodnog primjera nakon „upscaling“ procesa.*

Osim teksta u sliku postoje mnoge druge primjere modela poput slike u slike *„img2img“* ili *„inpainting“* tehnike generiranja određenog dijela slika. Te se može koristiti u kombinaciji s drugim LoRA modelima ili generiranja na drugim ili također precizno podešavanim SDXL modelima (prikazano na **Slika 13**.[76]).

***Slika 13.** Primjer „inpainting“ generiranja.*

[76] Kyoshino. Full moon rising over the pine trees and lots of stars. Preuzeto 4. rujna 2024. s Let's Talk Science. (2023). https://letstalkscience.ca/sites/default/files/2023-01/moon%20through%20trees.jpg [Fotografija].

7.3 Rasprava

Za ovo istraživanje odabran je difuzijski model, iz razloga lakšeg treniranja od generativnih suparničkih mreža, kod kojih često dolazi do problema tijekom treniranja. Također, difuzijski model dobar je za demonstraciju koncepata poput latentnog prostora i reprezentacije, radi sličnosti s varijacijskim autoenkoderima. Osim toga, oni su trenutno najzastupljeniji u komercijalnoj upotrebi (modeli poput Dalle 3 i Midjourney).

Iz tog razloga konkretno je odabran Stable Diffusion model radi njegove popularnosti i sličnosti s spomenutim modelima. Nadalje, njegova arhitektura se sastoji od varijacijskog enkodera, posebnog UNet modela s elementima transformera i tekst enkodera (CLIP).

S obzirom da je Stable diffusion već prethodno trenirani model, bilo je lakše dobiti dobre rezultate, te je poslužio kao primjer demonstracije preciznog podešavanja. Nadalje, omogućio je uspoređivanje dvaju modela i kao mogućnost generiranja već njemu poznatih pojmova.

Korištenje LoRA tehnike preciznog podešavanja pokazalo se kao brz i učinkovit način za treniranje Stable Diffusion XL modela.

U rezultatima su prikazani parametri: broja koraka uzorkovanja, CFG skale i utjecaj težina podešenih slojeva modela. Budući da su tijekom ovog istraživanja korišteni samo raspoređivači *(„schedulers“) „euler a“* i *“ DPM++ 2M Karras“,* na drugom modelu bi se mogli usporediti drugi raspoređivači.[77]

Budući da je treniran jednostavan koncept s malo varijacije kod ulaznih podataka, teško je donositi zaključke o kvaliteti modela, te je radi sličnosti u rezultatima teško uspoređivati ostale parametre generiranja. Radi lakšeg donošenja zaključaka i usporedbe trebalo bi ponoviti treniranje na drugom skupu podataka.

Rezultat ističu važnost pravilnog odabira podataka za treniranje. Iako je odabran skup podataka pružio solidnu osnovu s ciljem treniranja jasno vidljivih rezultata, raznolikost skupa podataka ograničila je opseg testiranja.

---

[77] Hugging Face. (n.d.). *Schedulers* [Dokumentacija za Diffusers v0.16.0]. Preuzeto 4. rujna 2024. s https://huggingface.co/docs/diffusers/v0.16.0/en/using-diffusers/schedulers

8. Zaključak

Generativni modeli pretvorbe teksta u sliku spajaju obradu prirodnog jezika (NLP) i računalnu viziju, omogućavajući stvaranje vizualnog sadržaja putem upita prirodnog jezika.

Tehnologija ima potencijal široke primjene u kulturnim i kreativnim industrijama, poput marketinga, izrade raznih vrsta dizajna i grafika, filmske i zabavne industrije, te drugim područjima poput obrazovanja i šire.

Tehnike preciznog podešavanja korisni su alati za prilagodbu modela dubokog učenja, od modela teksta u sliku i velikih modela jezika, do posebnih područja i namjena stvaranja sadržaja. Metode poput LoRA-e omogućavaju brzu i učinkovitu prilagodbu i prenamjenu već postojećih modela. Možemo očekivati sve lakšu integraciju novih, poboljšanih i jednostavnijih sučelja u svakodnevni život.

Modeli dubokog učenja imaju potencijal biti vezani uz svaki aspekt društva i pojedinca. Iz tog razloga, važno je fokusirati se na etičke implikacije umjetne inteligencije. Ona je još uvijek nova, pa je teško s sigurnošću predvidjeti njen doseg i rast. Međutim, trenutno rapidno napreduje i nove tehnike i metode se neprestano pojavljuju.

Cilj ovog rada bio je pružiti opširan uvod u razne vrste i arhitekture modela teksta u sliku, te prikazati i objasniti neke od korištenih metoda. Nakon toga prikazati postupak i rezultate treniranja jednog generativnog modela, te moguće primjene. Budući da je tema novo područje brzog napretka, jedno poglavlje posvećeno je etici i implikacijama ove nove tehnologije. U idućim istraživanjima trebalo bi trenirati model na više epoha i na podatcima veće raznolikosti. Jer u ovom istraživanju nedostaje utjecaj koraka treninga na model.

Također, ova tema može se promatrati s filozofskog stajališta i kognitivnih znanosti, poput filozofije uma, estetike i etike. Budući da tehnologija postavlja nova pitanja o ljudskoj kreativnosti, načinima na koji stvaramo, obrađujemo informacije i slično.

9. Literatura

1. Analytics Vidhya. (n.d.). Mathematical prerequisites for understanding autoencoders and variational autoencoders (VAEs). Preuzeto 4. rujna 2024. s https://medium.com/analytics-vidhya/mathematical-prerequisites-for-understanding-autoencoders-and-variational-autoencoders-vaes-8f854025390e

2. Amanatulla, M. (n.d.). Fine-tuning the model: What, why, and how. Preuzeto 15. kolovoza 2024. s https://medium.com/@amanatulla1606/fine-tuning-the-model-what-why-and-how-e7fa52bc8ddf

3. Alammar, J. (n.d.). The illustrated transformer. Preuzeto 4. rujna 2024. s https://jalammar.github.io/illustrated-transformer/

4. Alković, G. (2018). *Proširenja Jupyter notebook razvojne okoline* (Diplomski rad). Zagreb: Sveučilište u Zagrebu, Fakultet organizacije i informatike. Preuzeto s https://urn.nsk.hr/urn:nbn:hr:211:987871

5. Allen-Zhu, Z., & Orecchia, L. (2014). Linear coupling: An ultimate unification of gradient and mirror descent. *arXiv*. https://doi.org/10.48550/arXiv.1407.1537

6. AssemblyAI. (n.d.). Diffusion models for machine learning: Introduction. Preuzeto 4. rujna 2024. s https://www.assemblyai.com/blog/diffusion-models-for-machine-learning-introduction/

7. Awad, M., & Khanna, R. (2015). Machine learning. In *Efficient learning machines*. Apress. https://doi.org/10.1007/978-1-4302-5990-9_1

8. Babić, D. (2023). *Varijacijski autoenkoderi s kvantiziranom latentnom reprezentacijom* (Završni rad). Zagreb: Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva. Preuzeto s https://urn.nsk.hr/urn:nbn:hr:168:456628

9. Bishop, C. M. (2006). Pattern recognition and machine learning. Microsoft Research. Preuzeto 4. rujna 2024. s https://www.microsoft.com/en-us/research/uploads/prod/2006/01/Bishop-Pattern-Recognition-and-Machine-Learning-2006.pdf

10. Bostrom, N. (2011). The ethics of artificial intelligence. Preuzeto 4. rujna 2024. s https://www.techpolicy.press/laion5b-stable-diffusion-and-the-original-sin-of-generative-ai/

11. Brilliant.org. (n.d.). Backpropagation. Preuzeto 4. rujna 2024. s https://brilliant.org/wiki/backpropagation

12. Chang, Z., Koulieris, G. A., & Shum, H. P. H. (2023). On the design fundamentals of diffusion models: A survey. *arXiv*. Preuzeto 4. rujna 2024. s https://arxiv.org/abs/2303.06150

13. COCO. (n.d.). COCO dataset. Preuzeto 4. rujna 2024. s https://cocodataset.org/#home

14. Cornell University. (n.d.). Adam. Preuzeto 4. rujna 2024. s https://optimization.cbe.cornell.edu/index.php?title=Adam

15. CS231n. (n.d.). Transfer learning. Preuzeto 4. rujna 2024. s https://cs231n.github.io/transfer-learning/

16. DiningPhil. (n.d.). The reparameterization trick. Preuzeto 4. rujna 2024. s https://diningphil.github.io/project/reparamtrick/

17. EITCA. (n.d.). How can overfitting be visualized in terms of training and validation loss? Preuzeto 4. rujna 2024. s https://hr.eitca.org/artificial-intelligence/eitc-ai-tff-tensorflow-fundamentals/overfitting-and-underfitting-problems/solving-models-overfitting-and-underfitting-problems-part-1/examination-review-solving-models-overfitting-and-underfitting-problems-part-1/how-can-overfitting-be-visualized-in-terms-of-training-and-validation-loss/

18. EITCA. (n.d.). What is one-hot encoding? Preuzeto 4. rujna 2024. s https://hr.eitca.org/artificial-intelligence/eitc-ai-gcml-google-cloud-machine-learning/first-steps-in-machine-learning/plain-and-simple-estimators/what-is-one-hot-encoding/

19. Elements of AI. (n.d.). Generative adversarial networks. Preuzeto 4. rujna 2024. s https://course.elementsofai.com/hr/5/3

20. Enciklopedija.hr. (n.d.). Skalarni umnožak. Preuzeto 4. rujna 2024. s https://www.enciklopedija.hr/clanak/skalarni-umnozak

21. Fer.unizg.hr. (n.d.). *Bayesov klasifikator*. Preuzeto 4. rujna 2024. s https://www.fer.unizg.hr/_download/repository/SU-2016-15-BayesovKlasifikator.pdf

22. Fer.unizg.hr. (n.d.). *Logistička regresija*. Preuzeto 4. rujna 2024. s https://www.fer.unizg.hr/_download/repository/SU-2019-06-LogistickaRegresija.pdf

23. Fer.unizg.hr. (n.d.). *Uvod*. Preuzeto 4. rujna 2024. s https://www.fer.unizg.hr/_download/repository/01-Uvod-1s.pdf

24. GitConnected. (n.d.). LoRA: Low-rank adaptation explained with examples. Preuzeto 4. rujna 2024. s https://levelup.gitconnected.com/lora-low-rank-adaptation-explained-with-examples-fc3e31cfa243

25. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. U *Proceedings of the International Conference on Neural Information Processing Systems (NIPS 2014)* (str. 2672–2680).

26. Ho, J., & Ermon, S. (2016). Generative adversarial imitation learning. U *Advances in Neural Information Processing Systems* (Vol. 29, str. 4565–4573). https://doi.org/10.48550/arXiv.1606.03476

27. Hugging Face. (n.d.). Masked language modeling. Preuzeto 4. runa 2024. s https://huggingface.co/docs/transformers/tasks/masked_language_modeling

28. Hugging Face. (n.d.). Language modeling. Preuzeto 15. kolovoza 2024. s https://huggingface.co/docs/transformers/tasks/language_modeling

29. Hugging Face. (n.d.). Schedulers [Dokumentacija za Diffusers v0.16.0]. Preuzeto 4. rujna 2024. s https://huggingface.co/docs/diffusers/v0.16.0/en/using-diffusers/schedulers

30. IBM. *Strojno učenje*. Preuzeto 4. rujna 2024. s https://www.ibm.com/topics/machine-learning

31. IEEE Computer Society. (2022). *The rise of generative AI*. Preuzeto 4. rujna 2024. s https://www.computer.org/csdl/magazine/co/2022/10/09903869/1H0G6xvtREk

32. Isola, P., Zhu, J.-Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial nets. U *Computer Vision and Pattern Recognition*.

33. Jebara, T. (2004). Machine learning: Discriminative and generative. In *The Springer International Series in Engineering and Computer Science*. Kluwer Academic (Springer). https://doi.org/10.1007/978-1-4020-7647-3

34. Kabić, S. (2023). *Povratne neuronske mreže i primjene* (Diplomski rad). Zagreb: Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet. Preuzeto s https://urn.nsk.hr/urn:nbn:hr:217:857092

35. Kaggle. (n.d.). Sun and moon images. Preuzeto 4. rujna 2024. s https://www.kaggle.com/datasets/khushipitroda/sun-and-moon-images?resource=download

36. Khadka, P. (n.d.). Low-rank adaptation (LoRA). Preuzeto 4. rujna 2024. s https://medium.com/@pranjalkhadka/low-rank-adaptation-lora-fedf37b92026

37. Kingma, D. P., & Welling, M. (2013). Auto-Encoding Variational Bayes. *arXiv*. https://doi.org/10.48550/arXiv.1312.6114

38. Kirkpatrick, J. (2023). The state of artificial intelligence in 2023. Congressional Research Service. Preuzeto 4. rujna 2024. s https://crsreports.congress.gov/product/pdf/LSB/LSB10922

39. Kramer, M. A. (1992). Autoassociative neural networks. *Computers & Chemical Engineering*, 16(4), 313–328. doi:10.1016/0098-1354(92)80051-A.

40. Kyoshino. Full moon rising over the pine trees and lots of stars. Preuzeto 4. rujna 2024. s Let's Talk Science. (2023). https://letstalkscience.ca/sites/default/files/2023-01/moon%20through%20trees.jpg [Fotografija].

41. LAION. (2022). LAION-5B. Preuzeto 4. rujna 2024. s https://laion.ai/blog/laion-5b/

42. Lucidrains. (n.d.). Denoising diffusion pytorch. Preuzeto 4. rujna 2024. s https://github.com/lucidrains/denoising-diffusion-pytorch?tab=readme-ov-file

43. Machine Learning Mastery. (n.d.). What are generative adversarial networks (GANs)? Preuzeto 4. rujna 2024. s https://machinelearningmastery.com/what-are-generative-adversarial-networks-gans/

44. Microsoft Research. (2023). A deep generative model trifecta: Three advances that work towards harnessing large-scale power. Preuzeto 4. rujna 2024. s https://www.microsoft.com/en-us/research/blog/a-deep-generative-model-trifecta-three-advances-that-work-towards-harnessing-large-scale-power/

45. Mitchell, T. M. (2015). Generative and discriminative classifiers: Naive Bayes and logistic regression. U Machine learning. MIT Press.

46. Mohamed, S., & Lakshminarayanan, B. (2016). Learning in implicit generative models. *arXiv*. https://doi.org/10.48550/arXiv.1610.03483

47. Ng, A. Y., & Jordan, M. I. (2002). On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes. *Neural Information Processing Systems (NIPS), 14.*

48. Paluchasz, A. (n.d.). Understanding OpenAI's CLIP model. Preuzeto 15. kolovoza 2024. s https://medium.com/@paluchasz/understanding-openais-clip-model-6b52bade3fa3

49. Pinheiro Cinelli, L., et al. (2021). Variational autoencoder. U *Variational methods for machine learning with applications to deep networks* (str. 111–149). Springer. https://doi.org/10.1007/978-3-030-70679-1_5

50. Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., & Sutskever, I. (2021). Learning transferable visual models from natural language supervision. U *International Conference on Machine Learning*. Preuzeto s https://arxiv.org/abs/2103.00020.

51. Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks.

52. Salha, R. (2023). *Transformer arhitektura* (Završni rad). Osijek: Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet primijenjene matematike i informatike. Preuzeto s https://urn.nsk.hr/urn:nbn:hr:126:272269

53. Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, 27 (July, October), 379–423, 623–656. Chemical Engineering, 16(4), 313–328. https://doi.org/10.1016/0098-1354(92)80051-A

54. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., & Ganguli, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics. U *Proceedings of the 32nd International Conference on Machine Learning* (Vol. 37, str. 2256–2265). PMLR. https://proceedings.mlr.press/v37/sohl-dickstein15.pdf

55. Stability AI. (n.d.). *Stable diffusion XL base 1.0*. Preuzeto 4. rujna 2024. s https://huggingface.co/stabilityai/stable-diffusion-xl-base-1.0/resolve/main/sd_xl_base_1.0.safetensors?download=true

56. Sveučilište u Zagrebu, Građevinski fakultet. (n.d.). *Stohastički procesi - vježbe*. Preuzeto 4. rujna 2024. s https://www.grad.unizg.hr/_download/repository/Stohasticki_procesi_-_vjezbe.pdf

57. Šegvić, S. (2018). *Modeli i algoritmi za strojno učenje*. Preuzeto 4. rujna 2024. s https://www.zemris.fer.hr/~ssegvic/modeli/hrkac18modeli.pdf

58. Tay, Y., Dehghani, M., Tran, V. Q., Garcia, X., Wei, J., Wang, X., Chung, H. W., Shakeri, S., & Bahri, D. (2023). UL2: Unifying language learning paradigms. *arXiv*. https://doi.org/10.48550/arXiv.2205.05131

59. TechTarget. *A timeline of machine learning history*. Preuzeto 4. rujna 2024. s https://www.techtarget.com/whatis/A-Timeline-of-Machine-Learning-History

60. Turing, A. M. (1950). Computing machinery and intelligence. *Mind, 59*(236), 433–460. https://doi.org/10.1093/mind/LIX.236.433

61. Unite.AI. (n.d.). A full guide to fine-tuning large language models. Preuzeto 4. rujna 2024. s https://www.unite.ai/hr/a-full-guide-to-fine-tuning-large-language-models/

62. Unite.AI. (n.d.). Generativni vs. diskriminativni modeli strojnog učenja. Preuzeto 4. rujna 2024. s https://www.unite.ai/hr/generativni-vs-diskriminativni-modeli-strojnog-u%C4%8Denja/

63. Unite.AI. (n.d.). Što je Bayesov teorem? Preuzeto 4. rujna 2024. s https://www.unite.ai/hr/%C5%A1to-je-Bayesov-teorem/

64. Unite.AI. (n.d.). Što je povratno širenje? Preuzeto 4. rujna 2024. s https://www.unite.ai/hr/%C5%A1to-je-povratno-%C5%A1irenje/

65. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. U *Advances in Neural Information Processing Systems* (Vol. 30). Curran Associates, Inc. Preuzeto s https://arxiv.org/pdf/1706.03762

66. Wikipedia. (n.d.). Autoencoder. Preuzeto 4. rujna 2024. s https://en.wikipedia.org/wiki/Autoencoder

67. Wikipedia. (n.d.). Evidence lower bound. Preuzeto 4. rujna 2024. s https://en.wikipedia.org/wiki/Evidence_lower_bound

68. Wikipedia. (n.d.). Generative adversarial network Preuzeto 4. rujna 2024. s https://en.wikipedia.org/wiki/Generative_adversarial_network

69. Wikipedia. (n.d.). Text-to-image model. Preuzeto 4. rujna 2024. s https://en.wikipedia.org/wiki/Text-to-image_model

70. Wikipedia. (n.d.). Transformer (deep learning architecture). Preuzeto 4. rujna 2024. s https://en.wikipedia.org/wiki/Transformer_(deep_learning_architecture)

71. Wikipedia. (n.d.). Variation Bayesian methods https://en.wikipedia.org/wiki/Variational_Bayesian_methods [pristupljeno 18.8.2024.]

72. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., et al. (2020). Transformers: State-of-the-art natural language processing. U *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations* (str. 38–45).

73. Xiong, R., Yang, Y., He, D., Zheng, K., Zheng, S., Xing, C., Zhang, H., Lan, Y., Wang, L., & Liu, T.-Y. (2020). On layer normalization in the transformer architecture. *arXiv*. https://doi.org/10.48550/arXiv.2002.04745

74. Zhao, S., Song, J., & Ermon, S. (2019). Infovae: Balancing learning and inference in variational autoencoders. U *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 33, str. 5885–5892).

Poveznica na repozitorij s modelom i ostalim datotekama praktičnog dijela. https://huggingface.co/LeonStrakos/Zavrsni_rad