

**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I
INFORMACIJSKIH TEHNOLOGIJA OSIJEK**

Sveučilišni diplomski studij Računarstva

**TRENIRANJE CONDITIONAL VAE NA
MIKROSKOPSKIM SLIKAMA**

Projektni zadatak - seminarski rad

Lukrecija Međimorec

Osijek, 2024.

Sadržaj

1. Uvod.....	3
2. Teorijska pozadina.....	4
2.1. VAE.....	4
2.2. CVAE.....	4
3. Metodologija.....	5
3.1. Skup podataka.....	5
3.2. Arhitektura mreže.....	7
3.2.1. Enkoder.....	7
3.2.2. Dekoder.....	8
3.3. Treniranje modela.....	10
4. Rezultati.....	12
5. Zaključak.....	15
6. Izvori.....	16

1. Uvod

Mikroskopska analiza krvi je dijagnostička metoda pregleda i analize krvnih stanica upotrebom mikroskopa, a pruža detaljne informacije o krvnim stanicama, što je ključno za dijagnosticiranje i praćenje mnogih bolesti i medicinskih stanja, uključujući leukemiju, anemiju i razne infekcije.

Conditional Variational Autoencoder (CVAE dalje u tekstu) je generativni model dubokog učenja koji može učiti složene distribucije podataka i generirati nove, realistične uzorke na temelju tih distribucija, što ga čini izuzetno korisnim za različite primjene gdje je potrebno generirati specifične vrste podataka.

Ovaj seminarski rad bavit će se primjenom CVAE modela za generiranje mikroskopskih slika krvnih stanica. Cilj rada jest istražiti sposobnost modela za učenje karakteristika različitih tipova krvnih stanica iz skupa podataka i generiranje novih slika na osnovu naučenog. Generirane slike mogu se koristiti za treniranje drugih modela strojnog učenja, simulaciju rijetkih stanja ili u obrazovne svrhe.

U radu će se opisati metodologija korištena za treniranje CVAE modela, analizirat će se dobiveni rezultati, pregledati relevantna literatura te razmotriti moguće primjene i ograničenja ove tehnike. Cilj rada je pokazati kako generativni modeli, poput CVAE, mogu doprinijeti napretku u medicinskoj dijagnostici i istraživanju, pružajući nove alate za analizu i razumijevanje složenih bioloških podataka.

2. Teorijska pozadina

2.1. VAE

Varijacijski autoenkoderi (VAE) vrsta su generativnih modela koji uče prikazati složene distribucije podataka koristeći metode iz područja vjerojatnosti. Opisani autoenkoderi kombinacija su neuronskih mreža i teorije varijacijskih metoda, a za cilj imaju generirati nove podatke slične onima iz skupa za treniranje.

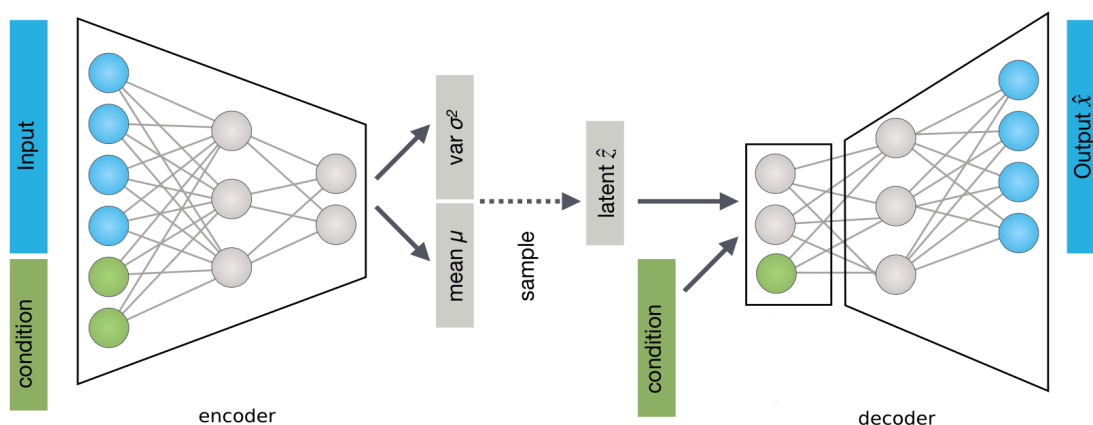
Varijacijski autoenkoder sastoji se od enkodera i dekodera. Enkoder transformira ulazne podatke u vektor značajki, odnosno točke u latentnom prostoru koje slijede određenu distribuciju. Za slične su slike točke u latentnom prostoru blizu, a za različite su slike točke udaljene. Vektor značajki sadrži manje informacija od originalne slike, a cilj dekodera jest iz vektora značajki rekonstruirati originalnu sliku, odnosno točke iz latentnog prostora transformirati u izvornu domenu podataka.

2.2. CVAE

Conditional VAE predstavlja napredno proširenje varijacijskog autoenkodera koje omogućuje generiranje podataka uz dodatnu informaciju, odnosno uvjet ili oznaku. Drugim riječima, osim slike, enkoderu i dekoderu predaje se i oznaka, pri čemu oznaka utječe na rezultat koji mreža generira što omogućuje eksplicitno generiranje slike određene klase. CVAE arhitektura prikazana je slikom 2.2.1.

Dodavanjem uvjeta, modelu je omogućeno učenje načina generiranja složenijeg latentnog prikaza koji uzima u obzir dodatne informacije. To poboljšava kvalitetu generiranih uzoraka u odnosu na VAE, i omogućuje modelu preciznije praćenje složenih odnosa između ulaznih podataka i uvjeta.

Budući da CVAE omogućuje kontrolirano generiranje podataka prema specifičnim uvjetima, može se koristiti u svrhu poboljšanja kvalitete skupova podataka što često pronalazi primjenu kod neuravnoteženih skupova podataka. Uz to, CVAE može pomoći u stvaranju sintetičkih slika tkiva, simulaciji bioloških procesa i generiranju podataka za rijetke bolesti čime se poboljšava istraživanje i povećava opseg edukativnih materijala.



Slika 2.2.1. CVAE arhitektura

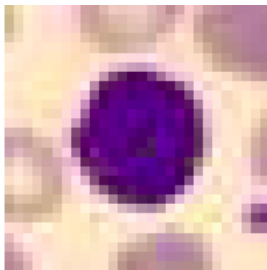


3. Metodologija






3.1. Skup podataka

Model je treniran koristeći BloodMNIST skup podataka iz MedMNIST kolekcije standardiziranih biomedicinskih slika.

Slike u skupu prikazuju krvne stanice zdravih pojedinaca bez infekcija te hematoloških i onkoloških bolesti koji u trenutku uzimanja uzorka nisu primali nikakvu farmakološku terapiju.

Skup se sastoji od ukupno 17,092 mikroskopske slike krvnog brisa organizirane u 8 klasa koje predstavljaju različite vrste krvnih stanica. Značenja pojedinih klasa (oznaka), njihov broj u skupu i primjer dani su tablicom 3.1.1. Iz originalnih slika rezolucije 3x360x360 piksela izrezane su nove slike rezolucije 3x200x200 piksela nakon čega je veličina pojedine slike smanjena na 3x28x28 piksela. Skup je zatim dodatno podijeljen na skup za trening, validaciju i test.

Klasa	Značenje	Broj slika u trening skupu	Primjer
0	bazofili	852	
1	eozinofili	2181	
2	eritroblasti	1085	

3	nezreli granulociti (mijelociti, metamijelociti, promijelociti)	2026	
4	limfociti	849	
5	monociti	993	
6	neutrofili	2330	
7	trombociti	1643	

Tablica 3.1.1. Primjeri slika pojedine klase iz ulaznog skupa podataka

3.2. Arhitektura mreže

Mreža koja je izrađena u sklopu ovog projektnog zadatka sastoji se od enkodera i dekodera implementiranih koristeći PyTorch biblioteku otvorenog koda. Pytorch nudi intuitivno i fleksibilno radno okruženje i funkcije za rad s tenzorima koje se mogu izvršavati na grafičkoj procesorskoj jedinici, što omogućuje brzu i učinkovitu obradu velikih količina podataka.

3.2.1. Enkoder

Enkoder je dio mreže odgovoran za transformaciju ulaznih podataka u vektor značajki koji sadrži manje informacija od originalne slike. Enkoder se sastoji od nekoliko konvolucijskih slojeva koji postupno smanjuju dimenzionalnost ulaznih slika i istovremeno povećavaju broj kanala.

Ulazne su slike u boji, točnije, imaju tri kanala. Ulaznoj se slici kao dodatni kanali dodaju i one-hot enkodirane oznake proširene na prostorne dimenzije slike. Drugim riječima, oznake se pretvaraju u vektor vrijednosti, pri čemu su sve vrijednosti jednake nuli osim one koja odgovara promatranoj oznaci i koja u vektoru ima vrijednost jednaku 1. Vektor se zatim proširuje na tenzor kojemu dimenzije odgovaraju dimenzijama ulazne slike.

Enkoder koristi šest konvolucijskih slojeva koji postupno smanjuju dimenzije slike uz povećanje broja kanala. Svaki sloj koristi kernel veličine 3×3 i stride veličine 2, osim posljednjeg kojemu je stride jednak 1.

Početni broj ulaznih kanala je 11, od čega su tri kanala osnovni kanali slike dok ostale kanale čini 8 oznaka. Prvi sloj povećava broj kanala na 16, drugi na 32, treći na 64, četvrti na 128, a peti i šesti na 256, odnosno 512 kanala.

Izlazu posljednjeg konvolucijskog sloja se smanjuje dimenzija i prolazi kroz dva linearna sloja kako bi se dobili parametri distribucije latentnog prostora, srednja vrijednost i logaritamska varijanca. Ovi parametri koriste se za uzorkovanje i generiranje latentne varijable z .

Latentna varijabla z generira se koristeći funkciju reparametrizacije. Točnije, varijabla z dobiva se kao

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

pri čemu μ predstavlja srednju vrijednost, σ standardnu devijaciju dobivenu iz logaritamske varijance, a ϵ uzorak iz standardne normalne distribucije.

3.2.2. Dekoder

Dekoder je odgovoran za rekonstrukciju ulaznih podataka iz latentnih varijabli pomoću uvjeta, odnosno oznaku slika.

Ulaz u dekodeer jest latentna varijabla z kombinirana s one-hot enkodiranim oznakama kako bi se osiguralo uvjetno generiranje. Takav ulaz prolazi kroz linearni sloj koji transformira latentni prostor natrag u oblik koji odgovara dimenzijama posljednjeg sloja enkodera.

Nakon linearnog sloja koristi se šest transponiranih konvolucijskih slojeva kako bi se postupno povećala prostorna dimenzija i smanjio broj kanala, čime se dimenzije ulaznih slika vraćaju na originalne. Na opisani se način kroz slojeve smanjuje broj kanala s 512 do 11, od kojih su 3 osnovni slike i 8 oznaka.

Nakon svakog konvolucijskog sloja enkodera i transponiranog konvolucijskog sloja dekodeera (osim posljednjeg) koristi se Leaky ReLU aktivacijska funkcija za uvođenje nelinearnosti. Ako je ulazna vrijednost pozitivna, izlaz će biti jednak ulaznoj vrijednosti, a ako je ulazna vrijednost negativna, izlaz će biti mali negativan broj koji je proporcionalan ulaznoj vrijednosti. Budući da navedena aktivacijska funkcije ne postavlja negativne vrijednosti na nulu, njen je izlaz mali gradijent za negativne vrijednosti, čime je omogućena bolja propagacija informacija kroz dublje slojeve mreže.

Nakon posljednjeg sloja dekodeera koristi se sigmoidna aktivacijska funkcija koja ograničava izlaz na vrijednosti između 0 i 1.

3.2.3. Hiperparametri

Prilikom kreiranja modela potrebno je definirati i hiperparametre modela. Definirani hiperparametri uključuju random seed, stopu učenja, broj epoha, veličinu bloka i parametri arhitekture samog modela poput broja klasa, značajki i dimenzija latentnog prostora.

Postavljanjem random seed hiperparametra na određenu vrijednost osigurava se mogućnost reprodukcije rezultata. Kada se koristi jednaka vrijednost, nasumični procesi u programu daju jednake nizove nasumičnih brojeva svaki put kada se program pokrene. Vrijednost opisanog hiperparametra je proizvoljna, ali je čest odabir vrijednosti 0 zbog čega je u ovom slučaju odabrana upravo ta vrijednost.

Stopa učenja određuje koliko brzo će se model prilagođavati promjenama u podacima tijekom treninga te upravlja brzinom konvergencije i stabilnošću treniranja modela. Vrijednost je proizvoljna, no česta je praksa odabrati malu vrijednost stope učenja te istu mijenjati tijekom treninga u ovisnosti o performansama. U ovom slučaju inicijalna vrijednost stope učenja postavljena je na 0.001 te je korišten Adam optimizacijski algoritam za optimizaciju.

Broj epoha definira koliko puta cijeli skup podataka prolazi kroz model tijekom treninga, a u ovom je slučaju broj epoha postavljen na 300.

Veličina bloka (batch size) određuje koliko će se primjera iz skupa obraditi prije nego što se ažuriraju parametri modela. Odabirom veće vrijednosti veličine bloka moguće je ubrzati trening, dok manja vrijednost dovodi do stabilnijeg učenja i zahtijeva manje memorije.

Broj klasa predstavlja broj oznaka u skupu podataka. Budući da se u skupu nalazi osam različitih vrsta krvnih stanica, broj klasa postavljen je na 8 .

Broj značajki odnosi se na dimenzionalnost ulaznih podataka. U ovom je slučaju postavljen na vrijednost 2,352 budući da su dimenzije ulaznih slika 3x28x28 piksela.

Odabirom dimenzionalnosti latentnog prostora upravlja se složenošću prikaza. Povećanjem ovog parametra u konačnici se povećava složenost rekonstruirane slike, pri čemu rekonstruirana slika sadrži više detalja, no za sam trening potrebna je veća količina podataka i više računalnih resursa. Nakon više testiranih vrijednosti, navedeni je parametar postavljen na 128.

3.3. Treniranje modela

Budući da model radi sa slikama, od kojih svaka nosi veliku količinu informacija u obliku piksela, treniranje modela izvodi se koristeći grafičku procesorsku jedinicu koja ima sposobnost paralelnog izvršavanja. Paralelno izvršavanje ubrzava proces treniranja jer omogućuje istovremenu obradu više podataka, čime se smanjuje vrijeme potrebno za svaku iteraciju treniranja.

Treniranje modela vrši se tijekom 300 epoha. Tijekom svake epohe računa se Kullback-Leibler divergencija, dodaju se one-hot enkodirani uvjeti, računa ukupan gubitak i ažuriraju parametri modela.

Kullback-Leibler divergencija mjeri razliku između dvije distribucije. U ovom kontekstu, prva distribucija (posteriorna distribucija) latentnog prostora modelira se pomoću srednje vrijednosti i logaritamske varijance, a druga distribucija latentnog prostora predstavlja standardnu normalnu distribuciju. KL divergencija mjeri koliko distribucija latentnog prostora odstupa od normalne distribucije.

Matematički, KL divergencija računa se prema izrazu

$$D_{KL}(q(z|x)||p(z)) = \frac{1}{2} \sum_{j=1}^J (1 + \log((\sigma_j^{(x)})^2 - (\mu_j^{(x)})^2 - (\sigma_j^{(x)})^2))$$

pri čemu $\mu_j^{(x)}$ predstavlja srednju vrijednost posteriorne distribucije, $\sigma_j^{(x)}$ predstavlja varijancu posteriorne distribucije, a $\log((\sigma_j^{(x)})^2)$ predstavlja logaritam omjera varijance posteriorne distribucije i varijance standardne Gaussove distribucije, koja iznosi 1.

Tijekom epohe vrši se one-hot kodiranje oznaka. One-hot kodiranje pretvara oznake u binarni format, pri čemu je svaka oznaka predstavljena binarnim vektorom. U ovom je slučaju svaka oznaka pretvorena u vektor duljine 8, a svaki element unutar njega jednak je nuli, osim elementa na indeksu koji odgovara promatranoj oznaci, čija je vrijednost jednaka jedan. Kreiran je vektor zatim proširen do prostornih dimenzija ulaznih podataka s kojima je zatim spojen kako bi se modelu uz ulazni podatak predao i uvjet.

Ukupan se gubitak računa kao zbroj KL divergencije i binarne unakrsne entropije (engl. binary cross-entropy, BCE). BCE mjeri razliku između originalne i rekonstruirane slike na razini piksela. Svaki piksel promatra se kao zaseban element, a razlike se u konačnici zbrajaju kako bi se izmjerila točnost rekonstrukcije. Matematički se BCE računa po izrazu

$$BCE(p, q) = -\frac{1}{N} \sum_{i=1}^N [p_i \log(q_i) + (1 - p_i) \log(1 - q_i)]$$

gdje je p_i stvarna vrijednost piksela i , q_i vrijednost piksela koji je rekonstruirao model, a N ukupan broj piksela.


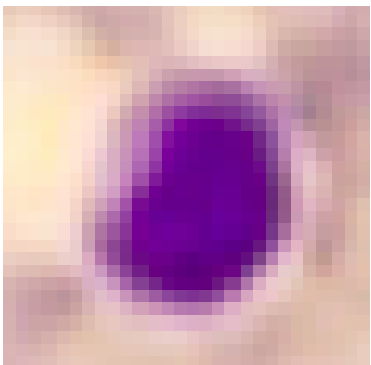

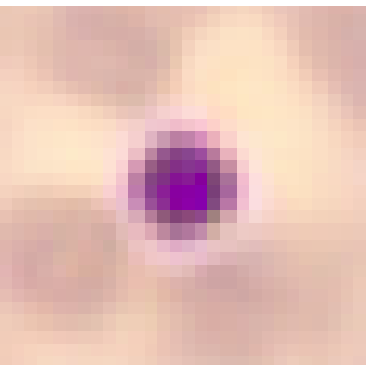

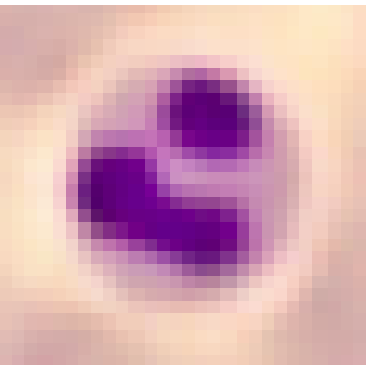
U konačnici se parametri modela ažuriraju kako bi se smanjio gubitak. Za optimizaciju parametara koristi se Adam optimizacijski algoritam koji računa prilagodljive stope učenja za svaki parametar modela. Budući da se gradijenti akumuliraju, isti se prije optimizacije prvo resetiraju tako što se

postave na nulu. Gradijenti ukupnog gubitka se zatim računaju za svaki parametar modela te se na osnovu dobivenih vrijednosti ažuriraju parametri modela s ciljem smanjenja ukupnog gubitka.

4. Rezultati

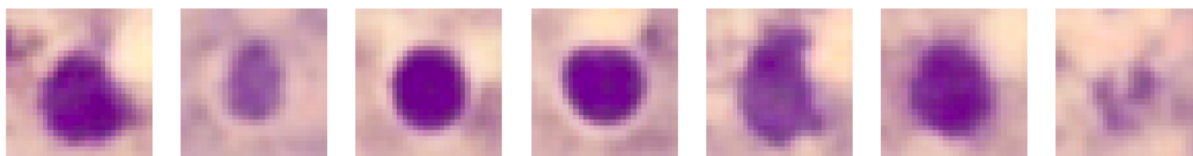
Budući da se CVAE fokusira na veće nositelje informacije u slici, a zanemaruje sitne detalje, očekivano je da će rekonstruirane slike biti zamućene u usporedbi s originalnim slikama. Jednako tako, nove generirane slike bit će mutne i bez puno detalja.

U tablici 4.1. prikazani su parovi originalnih slika i rekonstruiranih slika.

Originalna slika	Rekonstruirana slika
	
	
	

Tablica 4.1. Parovi originalnih i rekonstruiranih slika

Model osim rekonstruiranja ulaznih slika može generirati i nove slike. Primjeri generiranih slika za svaku od osam klasa dani su slikama u nastavku.



Slika 4.1. Klasa 0, bazofili



Slika 4.2. Klasa 1, eozinofili



Slika 4.3. Klasa 2, eritroblasti



Slika 4.4. Klasa 3, nezreli granulociti



Slika 4.5. Klasa 4, limfociti



Slika 4.6. Klasa 5, monociti



Slika 4.7. Klasa 6, neutrofili



Slika 4.8. Klasa 7, trombociti

4.1. Moguća poboljšanja

Lako je uočiti kako su generirane slike klase 6 koje predstavljaju neutrofile najbližnje originalnim slikama, kao i slike klase 1 koje predstavljaju eozinofile. Za navede klase postoji najviše slika u trening skupu pa je jedna od opcija za poboljšanje rezultata modela povećanje opsega trening skupa podataka. Povećanje opsega trening skupa podataka moguće je postići korištenjem različitih transformacija poput rotacije i okretanja slike, pojačavanja određene boje u slici, brisanja dijela originalne slike i zamućivanja ili izoštravanja slike.

Povećanje složenosti modela dodavanjem konvolucijskih slojeva također poboljšava sposobnost modela za uočavanje složenijih uzoraka u podacima. Moguće je dodatno poboljšati sposobnost učenja modela promjenom hiperparametara modela. Primjerice, povećanje dimenzija latentnog prostora povećava složenost rezultata, odnosno omogućuje modelu uočavanje više detalja u slici koji su u konačnici vidljivi na rekonstruiranim i novim generiranim slikama.

U svrhu je generiranja mikroskopskih slika krvnih stanica, osim CVAE, moguće koristiti i druge modele kao što su Generative Adversarial Network (GAN) i difuzijske modele koji kao rezultat daju oštre i realistične slike, ali je njihovo treniranje znatno sporije i zahtjevnije. GAN istovremeno optimizira dvije mreže što može biti nestabilno i zahtijevati mnogo resursa za podešavanje parametara, dok difuzijski modeli zahtijevaju velik broj iterativnih koraka za generiranje pojedine slike što povećava vrijeme treniranja.

5. Zaključak

Generiranje mikroskopskih slika krvnih stanica izvedeno je koristeći Conditional Variational Autoencoder. Analizom rezultata modela vidljivo je da CVAE rekonstruira ulazne slike s velikom točnošću, ali su rekonstruirane slike sklone zamućenju. Isto vrijedi za nove generirane slike, gdje je vidljiva veća sličnost ulaznim podacima kod klasa s većim brojem slika u skupu podataka za treniranje. Treniranje modela može se poboljšati povećanjem opsega skupa podataka, povećanjem složenosti arhitekture modela, dodavanjem slojeva i promjenom parametara modela. Bolje istreniran model može generirati slike s više detalja koje su sličnije slikama u ulaznom skupu podataka. Takve se slike mogu koristiti za treniranje drugih modela strojnog učenja, proširenje opsega neuravnoteženih skupova podataka ili za generiranje slika krvnih stanica s rijetkim anomalijama koje se mogu koristiti u obrazovne svrhe.

6. Izvori

- [1] Jiancheng Yang, Rui Shi, Bingbing Ni. *MedMNIST Classification Decathlon: A Lightweight AutoML Benchmark for Medical Image Analysis*. IEEE 18th International Symposium on Biomedical Imaging (ISBI), 2021. Imaging retrieved from: <https://medmnist.com/>
- [2] S. Le Groux. *The (Conditional Variational) Auto Encoder*, November 2022. Accessed: July 5, 2024. Available: <https://ccrma.stanford.edu/~slegroux/blog/posts/cvae/>
- [3] S. Raschka. *Deep Learning Models*,
https://github.com/rasbt/deeplearning-models/blob/master/pytorch_ipynb/autoencoder/ae-cnn-cvae.ipynb, Accessed: July 5, 2024.
- [4] I. Goodfellow, Y. Bengio, A. Courville. *Deep Learning*. MIT Press, 2016.
www.deeplearningbook.org
- [5] C. Doersch. *Tutorial on Variational Autoencoders*. UC Berkeley, August 2016. arXiv:1606.05908
- [6] C.M. Bishop. *Pattern Recognition and Machine Learning*. 2006.
- [7] D.P. Kingma, M. Welling. *Auto-Encoding Variational Bayes*. December 2022. arXiv:1312.6114
- [8] A. Takyar. *Optimize to actualize: The impact of hyperparameter tuning on AI*, Accessed: July 5, 2024. Available: <https://www.leewayhertz.com/hyperparameter-tuning/>
- [9] I. Dykeman. *Conditional Variational Autoencoders*. December, 2016. Accessed: July 5, 2024. Available: <https://ijdykeman.github.io/ml/2016/12/21/cvae.html>