SVEUČILIŠTE U ZAGREBU

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Matej Grcić

Detekcija izvandistribucijskih primjera primjenom generativnog modela Real NVP

Zagreb, 2020.

Ovaj rad izrađen je na Zavodu za elektroniku, mikroelektroniku, računalne i inteligentne sustave na Fakultetu elektrotehnike i računarstva, pod vodstvom prof. dr. sc. Siniše Šegvića i predan je na natječaj za dodjelu Rektorove nagrade u akademskoj godini 2019./2020.

Sadržaj

1 Uvod

Duboko učenje (*eng.* Deep learing) je potpodručje strojnog učenja (*eng.* Machine learing). Interes za ovim područjem intenzivno raste nakon pobjede dubokog konvolucijskog modela koji se zove AlexNet [Krizhevsky et al., 2012] na natjecanju ImageNet održanom 2012. godine, ovo područje doživljava rapidan razvoj. Upravo zahvaljujući tom razvoju, danas uspješno rješavamo neke probleme računalnog vida, obrade prirodnog jezika te probleme iz mnogih drugih područja.

Ubrzani razvoj dubokog učenja omogućili su javno dostupni veliki skupovi podataka, napreci u istraživanju optimizacijskih algoritama te naglo povećanje hardverskih mogućnosti uz smanjenje cijene. Konkretno, pojava *CUDA* tehnologije je omogućila smještanje dubokih modela na grafičke procesore te dodatno ubrzanje. Konačno, pojava programskih razvojnih okvira kao što su *PyTorch*, *TensorFlow* i drugi omogućila je jednostavnu implementaciju novih ideja te brzo izvođenje eksperimenata.

Duboki modeli su prediktivni modeli koji sadrže više od jedne nelinearne transformacije. Danas duboki modeli imaju desetke milijuna parametara te zahtijevaju nekoliko gigabajta memorije za uspješno treniranje postupkom propagacije pogreške unatrag (*eng.* Backpropagation). Duboki modeli su našli primjenu u području računalnog vida (*eng.* Computer vision). Računalni vid jest područje računarske znanosti koje želi omogućiti računalima da prepoznaju i obrađuju predmete u slikama i videozapisima na isti način kao što to čine ljudi.

Iako se duboki modeli danas intenzivno primjenjuju za rješavanje niza kompleksnih zadataka, imaju određene nedostatake koji predstavljaju zanimljiv smjer daljnjeg istaživanja. Jedan od glavnih nedostataka dubokih modela jest njihova naglašena samouvjerenost [Lakshminarayanan et al., 2017, Guo et al., 2017]. Naime, ukoliko dubokom modelu predamo primjer koji ne podliježe distribuciji

skupa podataka na kojem je model naučen, model takav primjer klasificira s velikom sigurnošću u krivi razred. Ovakve primjere nazivamo izvandistribucijski primjeri. Slično, ako primjeru iz skupa podataka dodamo pomno odabrau perturbaciju, tada model takav primjer klasificira u drugu klasu s velikom sigurnošću. Takve primjere nazivamo neprijateljskim primjerima. Obje navedene vrste primjera remete normalan rad dubokog modela i potencijalno su veliki sigurnosni problem.

U ovom radu promatramo problem izvandistribucijskih primjera u okviru računalnog vida. U provedenim eksperimentima se fokusiramo na skupove podataka za klasifikaciju. Ovaj rad se temelji na [Lee et al., 2018], u kojem se performanse diskriminativnog modela u detektiranju izvandistribucijskih primjera poboljšavaju uz pomoć generativnih suparničkih modela.

Glavni doprinosi ovog rada su: analiza postojećih procedura za poboljšanje performansi klasifikatora u detekciji izvandistribucijskih primjera, pojednostavljenje i poboljšanje procedure definirane u [Lee et al., 2018] korištenjem normalizirajućih vjerojatnosnih tokova te analiza metoda za generiranje primjera s ruba distribucije. Također uvodimo generativni model Real NVP [Dinh et al., 2017] temeljen na nizu invertibilnih transformacija u učenje klasifikatora sposobnog detektirati izvandistribucijske primjera. Real NVP detaljno objašnjavamo u nastavku rada.

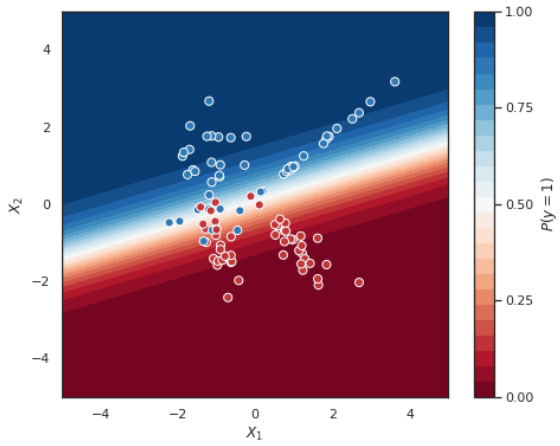
Ostatak rada organiziran je na sljedeći način. U poglavlju 2 definiramo osnovne pojmove kao što su diskriminativni i generativni modeli te način učenja istih. U poglavlju 3 iznosimo problem detekcije izvandistribucijskih primjera te nudimo pregled različitih pristupa pri rješavanju navedenog problema. U poglavlju 4 opisujemo generativne modele temeljene na normalizirajućem vjerojatnosnom toku (*eng.* normalizing flow). Fokusiramo se na autoregresijski tok Real NVP [Dinh et al., 2017] temeljen na invertibilnim transformacijama. U poglavlju

5 opisujemo našu varijantu detekcije izvandistribucijskih primjera koja se temelji na metodi definiranoj u [Lee et al., 2018] te metodi za generiranje primjera s ruba distribucije uz korištenje izvandistribucijskog skupa podataka. Konačno, u poglavlju 6 prikazujemo provedene eksperimente te s dobivenim rezultatima potkrepljujemo iznesene tvrdnje.

2 Osnovni pojmovi

2.1 Diskriminativni modeli

Diskriminativni modeli su modeli za klasifikaciju podataka koji se uče nadziranim učenjem. Karakteristika nadziranog učenja jest da skup podataka za učenje sadrži primjere $\mathbf{x}$ i pripadajuće oznake y iz predefinarnog skupa oznaka. Uz takav skup podataka za učenje moguće je izravno modelirati distribuciju $P(Y = y|\mathbf{x})$.

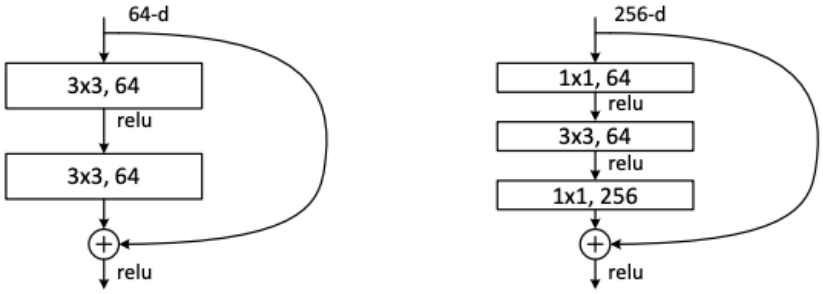


Slika 1: Granica između dvije klase primjera definirana diskriminativnim modelom.

Na temelju naučene distribucije $P(Y = y|\mathbf{x})$ možemo u prostoru primjera iz skupa podataka definirati granice između klasa danog skupa podataka (slika 1). Izlaz modela interpretiramo kao vjerojatnost da dani primjer $\mathbf{x}$ pripada klasi y. Posljedično, granica među klasama se nalazi na mjestima gdje vjerojatnost da primjer pripada određenoj klasi postaje manja od vjerojatnosti da primjer pripada nekoj drugoj klasi. S obzirom da ovakvi modeli izravno modeliraju distribuciju $P(Y = y|\mathbf{x})$, najčešće ih primjenjujemo na klasifikacijske probleme. U fazi zaključivanja dani primjer $\mathbf{x}$ klasificiramo u klasu s najvećom vrijednošću

$P(Y = y|\mathbf{x})$. Diskriminativne arhitekture za razumijevanje slike su ResNet i DenseNet arhitekture, koje detaljnije objašnjavamo u nastavku.

Arhitektura ResNet se temelji na rezidualnim jedinicama (*eng.* Residual unit). Postoje dvije vrste rezidualnih jedinica. Osnovna jedinica se sastoji od dva konvolucijska sloja s jezgrom dimenzije 3×3, sloja s normalizacijom po grupi (*eng.* batch normalization) [Ioffe and Szegedy, 2015] te nelinearne *ReLU* aktivacijske funkcije. Jedinica s uskim grlom (*eng.* bottleneck) se sastoji od tri konvolucijska sloja, tri nelinearne *ReLU* aktivacijske funkcije te tri sloja s normalizacijom po grupi. Prvi konvolucijski sloj kao i zadnji imaju jezgre veličine 1×1 dok srednji konvolucijski sloj koristi jezgru veličine 3×3. Prvi konvolucijski sloj s jezgrom veličine 1×1 projicira ulazni tenzor u tenzor s manjim brojem mapa u semantičkoj dimenziji. Posljednji konvolucijski sloj vraća dimenzije tenzora na dimenzije ulaznog tenzora. Obje vrste rezidualnih jedinica sadrže karakterističnu preskočnu vezu vidljivu na slici 2. Posljedica preskočne veze jest to da izlaznom tenzoru jedinice dodajemo ulazni tenzor (zbrajanje po elementu).



Slika 2: Primjeri rezidualnih blokova. Izvor: [He et al., 2016a]

Spomenimo i rezidualne jedinice s predaktivacijom [He et al., 2016b], gdje *ReLU* aktivacija ide prije rezidualne veze. Prednost korištenja rezidualnih jedi-

nica s predaktivacijom je u tome što gradijent teče modelom nepromijenjen, što dodatno olakšava optimizaciju [He et al., 2016b].

Rezidualne jedinice koje operiraju na istoj rezoluciji tvore rezidualne blokove (*eng.* Residual block). ResNet arhitektura nastaje slaganjem više (tipično četiri) rezidualnih blokova. Konkretne implementacije su ResNet18, ResNet34, ResNet50, ResNet101 te ResNet152 gdje ResNet18 i ResNet34 koriste osnovne rezidualne blokove, a ostale rezidualni blok s uskim grlom. Za detaljniji opis arhitekture ResNet čitatelja upućujemo na [He et al., 2016a, He et al., 2016b].

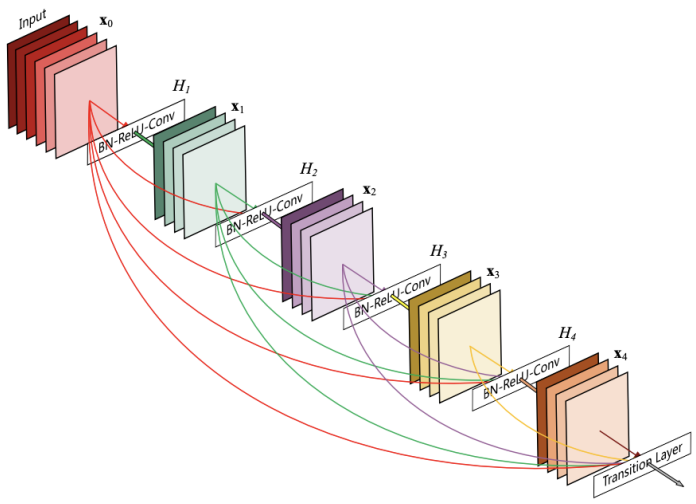
Rezidualne jedinice koje operiraju na istoj rezoluciji tvore rezidualne blokove (*eng.* Residual block). ResNet arhitektura nastaje slaganjem više (tipično četiri) rezidualnih blokova. Konkretne implementacije su ResNet18, ResNet34, ResNet50, ResNet101 te ResNet152, gdje ResNet18 i ResNet34 koriste osnovne rezidualne jedinice, a ostale rezidualne jedinice s uskim grlom. Slika 3 prikazuje arhitekture različitih ResNet arhitektura.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Slika 3: Različite ResNet arhitekture. Izvor: [He et al., 2016a]

Za detaljniji opis ResNet arhitekture čitatelja upućujemo na [He et al., 2016a].

Arhitektura DenseNet se temelji na gustim blokovima (*eng.* dense block). Svaki gusti blok se sastoji od odrenog broja gustih slojeva (*eng.* dense layer). Gusti sloj se sastoji od sljedećeg niza transformacija: normalizacija po grupi, nelinearna transformacija $ReLU$, konvolucija s jezgrom dimenzije 1×1, normalizacija po grupi, nelinearna transformacija $ReLU$ i konačno konvolucija s jezgrom dimenzije 3×3. Isti redosljed transformacija imamo u rezidualnim jedinicama s predaktivacijom [He et al., 2016b]. Slično kao ResNet arhitektura, DenseNet arhitektura koristi preskočne veze. Izlaz svakog gustog sloja se prosljeđuje na ulaz svih sljedećih slojeva gusto povezanog bloka (slika 4).



Slika 4: Gusto povezani blok arhitekture DenseNet. Izvor: [Huang et al., 2017].

Spomenimo još i tranzicijski blok (*eng.* transition block) koji sadrže sljedeći niz transformacija: normalizacija po grupi, nelinearna transformacija $ReLU$, konvolucija s jezgrom dimenzije $1x1$ te sloj sažimanja. Tranzicijski blok se umeće između svaka dva susjedna gusta bloka s ciljem da se veličina prostornih dimenzija tenzora smanji na pola. U DenseNet-BC arhitekturi tranzicijski blok komprimira

broj mapa značajki (tipično za 0.5). Konkretne implementacije DenseNet arhitekture su DenseNet-121, DenseNet-169, DenseNet-201 i konačno DenseNet-264. Slika 5 prikazuje navedene arhitekture.

Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	112 × 112	7 × 7 conv, stride 2			
Pooling	56 × 56	3 × 3 max pool, stride 2			
Dense Block (1)	56 × 56	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	56 × 56	1 × 1 conv			
	28 × 28	2 × 2 average pool, stride 2			
Dense Block (2)	28 × 28	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	28 × 28	1 × 1 conv			
	14 × 14	2 × 2 average pool, stride 2			
Dense Block (3)	14 × 14	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 24$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 48$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 64$
Transition Layer (3)	14 × 14	1 × 1 conv			
	7 × 7	2 × 2 average pool, stride 2			
Dense Block (4)	7 × 7	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 16$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 48$
Classification Layer	1 × 1	7 × 7 global average pool			
		1000D fully-connected, softmax			

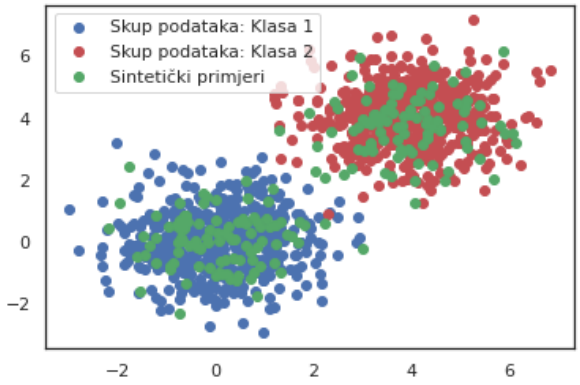
Slika 5: Različite DenseNet arhitekture. Izvor: [Huang et al., 2017]

Za više informacija čitatelja upućujemo na [Huang et al., 2017].

2.2 Generativni modeli

Generativni modeli aproksimiraju zajedničku distribuciju podataka $p(x, y)$ (slika 6). Aproksimaciju distribucije $p(x, y)$ označavamo s $q(x, y)$. Na temelju distribucije $q(x, y)$ moguće je generirati nove primjere, što je ujedno i razlog zašto ove modele nazivamo generativnima. Generativne modele također možemo primijeniti na klasifikacijske probleme. Distribuciju $p(y|x)$ možemo odrediti iz distribucije $p(x, y)$ primjenom Bayesovog pravila:

$$P(y|x) = \frac{p(x, y)}{p(x)} = \frac{p(x|y)p(y)}{\sum_y p(x|y)p(y)} \tag{1}$$



Slika 6: Generativni modeli aproksimiraju distribuciju odabranog skupa podataka. Uzorkovanjem takvog modela dobijemo sintetičke primjere slične primjerima iz skupa podataka za učenje.

Iako je generativnim modelom moguće modelirati distribuciju $p(y|x)$, preferiramo korištenje diskirminativnih modela za klasifikaciju. Razlog leži u broju parametara modela što objašnjavamo uspoređujući logističku regresiju i Bayesov klasifikator na problemu binarne klasifikacije. Krećemo od vjerojatnosti da dani primjer pripada prvoj klasi. Izraz 1 transformiramo na sljedeći način:

$$P(y=1|\mathbf{x}) = \frac{p(\mathbf{x}|y=1)P(y=1)}{\sum_{i=1}^{2} p(\mathbf{x}|y=i)P(y=i)} = \frac{1}{1+\exp(\ln\frac{p(\mathbf{x}|y=2)P(y=2)}{p(\mathbf{x}|y=1)P(y=1)})} \tag{2}$$

iz čega sljedi:

$$P(y=1|\mathbf{x}) = \frac{1}{1+\exp(-\alpha)} = \sigma(\alpha) \quad \alpha = \ln\frac{p(\mathbf{x}|y=1)P(y=1)}{p(\mathbf{x}|y=2)P(y=2)} \tag{3}$$

Ako pretpostavimo dijeljenu kovarijacijsku matricu između distribucija $P(\mathbf{x}|y)$, tada je $\alpha = \mathbf{w}^T\mathbf{x}$, čime smo dobili logističku regresiju. Bayesov klasifikator ima $\frac{n(n+1)}{2} + nK + K - 1$ parametara, gdje n označava broj dimenzija primjera, a K broj klasa. Logistička regresija ima $nK + K$ parametara.

U slučaju kada imamo neoznačeni skup podataka, prelazimo u domenu nenadziranog učenja, a generativni modeli uče distribuciju $P(\mathbf{x})$. Generativne modele možemo promatrati kao nelinearne funkcije latentne varijable z:

$$P(\mathbf{x}) = \int_z P(\mathbf{x}|f_\theta(\mathbf{z}))P(\mathbf{z}) \quad P(\mathbf{z}) = N(\mathbf{z}|0, \mathrm{I}) \tag{4}$$

gdje nelinearna funkcija $f_\theta(\mathbf{z})$ može biti duboki model. Integral u 4 je netraktabilan zbog nelinarne transformacje $f_\theta(\mathbf{z})$. Posljedično, duboke generativne modele dijelimo prema načinu kako rješavaju problem netraktabilnog integrala.

Ograničeni Boltzmannov stroj (*eng.* restricted Boltzmann machines) definira distribuciju $P(\mathbf{x})$ preko energijske funkcije:

$$P(\mathbf{v}) = \sum_{\mathbf{h}} P(\mathbf{v}, \mathbf{h}) = \sum_{\mathbf{h}} \frac{\exp(-E(\mathbf{v}, \mathbf{h})/T)}{Z} \tag{5}$$

gdje je:

$$E(\mathbf{v}, \mathbf{h}) = -\frac{1}{2}\mathbf{v}^T\mathbf{W}\mathbf{h}, \quad Z = \sum_{\mathbf{v}}\sum_{\mathbf{h}} \exp(-E(\mathbf{v}, \mathbf{h})) \tag{6}$$

Ovaj se model se sastoji od dva sloja potpuno povezanih neurona (vidljivi i skriveni sloj). Arhitektura modela je prikazana na slici 7. Model se uči na sljedeći način. $P(\mathbf{x})$ možemo rastaviti kao umnožak nenormalizirane distribucije te normalizirajuće konstante:

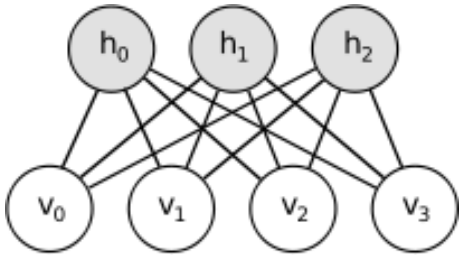
$$P(\mathbf{x}) = \frac{1}{Z}\hat{P}(\mathbf{x}) \tag{7}$$

Model učimo maksimiziranjem logaritma izglednosti navedenog izraza, što je ekvivalentno izrazu:

$$\log P(\mathbf{x}) = \log \hat{P}(\mathbf{x}) - \log Z \tag{8}$$

Ovaj izraz optimiramo u dvije faze. U pozitivnoj fazi maksimiziramo član $\log \hat{P}(\mathbf{x})$. Posljedica ove faze jest to da se distribucija modela prilagođava uzorcima iz skupa

podataka za učenje. Negativna faza učenja minimizira član $\log Z$. Zbog kompleksnosti izračuna normalizirajuće konstante, negativnu fazu možemo samo aproksimirati. Za aproksimaciju najčešće koristimo algoritam kontrastne divergencije [Carreira-Perpiñán and Hinton, 2005]. Efektivno, u negativnoj fazi uzorkujemo distribuciju modela te smanjujemo vjerojatnost uzorkovanih primjera.



Slika 7: Restricted Boltzmann Machine (RBM). Izvor: Google image search

Model uzorkujemo tako da fiksiramo vrijednosti skrivenih neurona te uzorkujemo distribuciju $P(\mathbf{v}|\mathbf{h})$ prema izrazu:

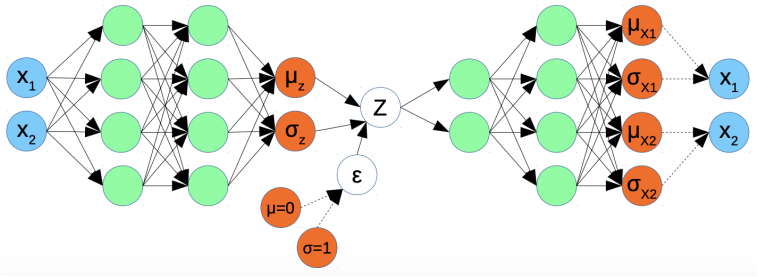
$$P(\mathbf{v}|\mathbf{h}) = \sigma(\mathbf{W}^T\mathbf{h}) \tag{9}$$

Za više informacija čitatelja upućujemo na [Salakhutdinov et al., 2007].

Varijacijski autoenkoderi (*eng.* Variational Autoencoders) aproksimiraju netraktabilan inegral iz 4. Ovaj model modelira latentnu varijablu $\mathbf{z}$ rekonstrukcijom podataka iz skupa za učenje. Enkoderski dio transformira primjer $\mathbf{x}$ iz skupa za učenje u latentnu varijablu z koja se ravna po normalnoj razdiobi, odnosno uči očekivanje i varijancu te distribucije. Dekoderski dio pokušava rekonstruirati dani primjer iz latentne varijable z. Ovaj model optimira gubitak:

$$\mathbb{E}_{q_\psi(z|x)}[\log(p_\theta(\mathbf{x}|z))] - \mathrm{KL}(q_\theta(z|\mathbf{x})||p(z)) \tag{10}$$

što je donja granica od $\log p(\mathbf{x})$. Slika 8 prikazuje arhitekturu dubokog varijacijskog autonekodera. Za više informacija o ovom modelu čitatelja upućujemo na [Kingma and Welling, 2014].



Slika 8: Arhitektura varijacijskog autoenkodera. Izvor: Duboko učenje, Tema 7: Generativni duboki modeli, FER Zagreb

Generativni suparnički modeli (*eng.* generative adversarial networks) uče $P(\mathbf{x})$ zaobilaženjem integrala. Naime, model generator i model diskriminator su suprotstavljeni u optimiranju izraza:

$$\min_{\theta} \max_{\psi} \quad \mathbb{E}_{x \sim P(x)}[\log D_{\psi}(x)] + \mathbb{E}_{z \sim P(z)}[\log(1 - D_{\psi}(G_{\theta}(z)))] \tag{11}$$

Optimiranje navedenog izraza je ekvivalentno minimax igri s optimumom u točki sedla koja odgovara Nashevom ekvilibriju. Diskriminator pokušava razlikovati primjere iz skupa podataka od sintetičkih primjera stvorenih generatorom. Generator pokušava zavarati diskriminator generiranjem što vjernijh primjera. Rezultat optimiranja navedenog gubitka jest sposobnost generatora da generira primjere koji se ravnaju po razdiobi $P(\mathbf{x})$.

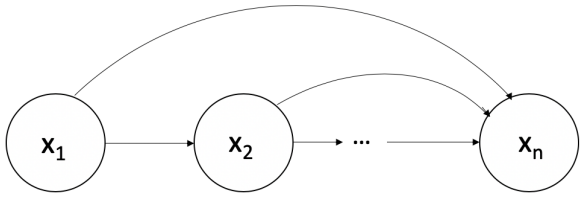
Prema [Lucas et al., 2019], suparničko učenje se fokusira isključivo na kvalitetu primjeraka što rezultira izostavljanjem modova distribucije skupa podataka (*eng.* mode collapse). S druge strane, učenje maksimizacijom izglednosti rezultira distribucijom modela koja obuhvaća sve modove distribucije podataka, ali

određene dijelove vjerojatnosnog volumena smješta u nekarakteristična područja vjerojatnosnog prostora s obzirom na distribuciju skupa podataka. Više informacija o generativnim suparničkim modelima se nalazi u [Goodfellow et al., 2014].

Autoregresijski modeli (*eng.* Autoregressive models) izbjegavaju integral u izrazu 4 tako da se ne oslanjaju na latentnu varijablu. Distribucija $P(\mathbf{x})$ se faktorizira prema pravilu lanca na sljedeći način:

$$P(\mathbf{x}_{1:d}) = P(\mathbf{x}_1) \prod_{i=2}^{D} P(\mathbf{x}_i | \mathbf{x}_{1:i-1}) \tag{12}$$

Modeliranje distribucije na ovakav način ne unosi nikakve apriorne pretpostavke o uvjetnoj nezavisnosti varijabli. Kompleksne distribucije $P(\mathbf{x}_i|\mathbf{x}_{1:i-1})$ modeliramo dubokim modelima. Slika 9 grafički prikazuje Bayesovu mrežu ovako modelirane distribucije. Mana ovakvog pristupa jest sporo uzorkovanje modela. Na primjeru slika, ovakav model uzorkuje piksel po piksel. Tipični primjeri ovakvih modela su objašnjeni u [van den Oord et al., 2016] i [Salimans et al., 2017]. Naglasimo da ova vrsta generativnih modela nema veze s normalizirajućim autoregresijskim tokovima.



Slika 9: Pravilo lanca prikazano grafički kao Bayesova mreža.

Normalizirajući tokovi (*eng.* normalizing flows) uvode restrikciju na funkciju $f_\theta(\mathbf{x})$ u izrazu 4. Funkcija mora biti invertibilna tj. bijekcija te mora čuvati broj

dimenzija. Uz navedene pretpostavke integral iz navedenog izraza postaje traktabilan. Uz iste pretpostavke distribucija $p(\mathbf{x})$ se može definirati preko formule za promjenu varijable distribucije što za posljedicu ima sposobnost egzaktne evaluacije izglednosti. Ova vrsta generativnih modela opisana je detaljnije u poglavlju 5.

Glavni zadaci generativnih modela su evaluacija izglednosti, uzorkovanje aproksimirane distribucije podataka i preslikavanje u latentnu reprezentaciju. Generativni suparnički modeli su sposobni generirati primjere visoke kvalitete. Varijacijski autoenkoderi kodiraju primjere u latentnu reprezentaciju. Autoregresijski modeli egzaktno evaluiraju izglednost danog primjera. Normalizirajući tokovi mogu efikasno obaviti sva tri zadatka. Ipak latentna reprezentacija danog primjera je jednakih dimenzija kao i primjer, dok pri uzorkovanju modela dobijemo vizualno mutnije primjere od onih generiranih GAN-om.

2.3 Učenje dubokih modela

Duboki modeli se uče optimiranjem funkcije gubitka. Funkcija gubitka mjeri pogrešku modela na danom primjeru iz skupa podataka. Tipična funkcija gubitka za klasifikacijske probleme jest unakrsna entropija, a za regresijske probleme kvadratni gubitak. Za razliku od klasičnog strojnog učenja, gdje je funkcija gubitka uvijek konveksna, u dubokom učenju to ne mora biti slučaj. Iz tog razloga, optimizacijski postupak nas ne dovodi uvijek u globalni minimum funkcije gubitka. Ipak, pokazalo se da dostizanje globalnog minimuma nije uvjet za dobro naučen model (lokalni minimumi su dovoljno dobri).

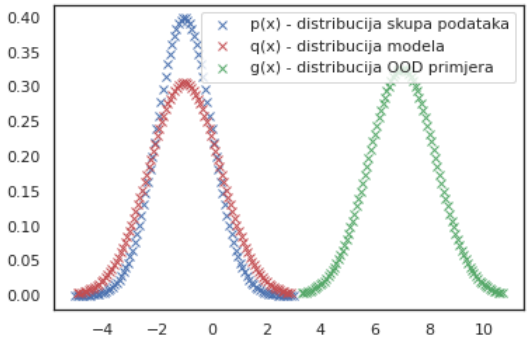
Optimiranje funkcije gubitka se provodi u dva koraka. Prvo gubitak modela propagiramo unatrag kroz sve slojeve modela. Na taj način za svaki parametar modela računamo pripadajući gradijent tj.derivaciju gubitka po odabranom parametru. Ovaj postupak se naziva propagacija greške unatrag (*eng.* back-

propagation). Sljedeći korak jest korištenje izračunatih gradijenata u postupku ažuriranja parametara. To se provodi uz pomoć nekog od optimizacijskih alogoritama. Najjednostavniji takav algoritam jest stohastični gradijentni spust, a koriste se još i AdaGrad [Duchi et al., 2011], RMSProp [Dauphin et al., 2015] i Adam [Kingma and Ba, 2015].

U klasičnom strojnom učenju modele učimo na dva načina. Kod učenja on-line metodom modelu prosljeđujemo jedan po jedan primjer te nakon svakog primjera ažuriramo parametre modela. Drugi način jest da modelu predamo cijelokupan skup podataka pa zatim optimiramo parametre modela. Da bismo naučili duboki model, potreban nam je relativno velik skup podataka. On-line metoda ne dolazi u obzir zbog nestabilnog gradijenta, dok bi ažuriranje parametara nakon cijelog skupa podataka bilo neučinkovito. Učenje dubokih modela provodimo hibridnom metodom. Modelu prikazujemo malu grupu primjera, tipično veličine 32, 64 ili 128 primjera, te nakon svake grupe ažuriramo parametre modela. Intuitivno, metoda se zove učenje s minigrupama (*eng.* Minibatch learning).

2.4 Unutardistribucijski primjeri

Unutardistribucijske primjere dobivamo uzorkovanjem skupa za učenje ili odgovoarajućeg generativnog modela. Ovi primjeri su najčešće generirani uz pomoć generativnog modela koji uči distribuciju odabranoga skupa podataka. Ako pretpostavimo da primjeri iz odabranoga skupa podataka podliježu distribuciji $p(x)$, tada unutardistribucijski primjeri podliježu distribuciji podataka $q(x)$, gdje je $q(x)$ aproksimacija distribucije $p(x)$. Slika 10 ilustrativno prikazuje primjere iz distribucije podataka plavim oznakama, dok su generirani primjeri označeni crvenim oznakama.



Slika 10: Prikaz primjera iz distribucije podataka (plavo), distribucije modela (crveno) i izvandistribucijskih primjera (zeleno).

2.5 Izvandistribucijski primjeri

Za razliku od unutardistribucijskih primjera, izvandistribucijske primjere nije jednostavno egzaktno definirati. U ovom radu, izvandistribucijske primjere definiramo kao primjere koji ne podliježu odabranoj distribuciji podataka kojoj podliježu unutardistribucijski primjeri. Na primjer, ukoliko definiramo skup podataka koji sadrži slike životinja kao unutardistribucijski skup, tada je primjer izvandistribucijskog skupa podataka skup sa slikama brojeva. Bitno je spomenuti i primjere s ruba distribucije. Intuitivno, to su primjeri kojima gustoća vjerojatnosti podataka dodijeljuje malu vjerojatnost. Na slici 10 su izvandistribucijski primjeri označeni zelenom bojom.

3 Pregled literature

3.1 Motivacija

Ubrzo nakon početnog razvoja dubokih modela, otkrivena je njihova naglašena samouvjerenost [Lakshminarayanan et al., 2017, Guo et al., 2017]. Duboki model izvandistribucijske primjere tipično klasificiraju u neku netočnu klasu s visokom sigurnošću. Upravo to je jedan od najvećih sigurnosnih problema sustava temeljenih na dubokom učenju. U literaturi rješenja ovog problema uključuju evaluaciju izglednosti, korištenje dodatnih modela ili dodatnih skupova podataka, što ćemo detaljnije razmotriti u nastavku.

3.2 Evaluacija izglednosti

Motivirani diskusijom na radionici *Advances in Approximate Bayesian Inference (AABI) 2017*, nekolicina znanstvenika je pokušala detektirati izvandistribucijske primjere na temelju izglednosti pojave danog primjera. Intuicija je jednostavna, modelu koji je sposoban evaluirati izglednost pokažemo primjer te ga označimo kao izvandistribucijski primjer ukoliko je dobiveni logaritam izglednosti malen. Za ovaj pristup najčešće koristimo autoregresivne modele ili modele temeljene na normalizirajućim tokovima zbog sposobnosti egzaktnog definiranja izglednosti. Iako intuitivan, prema [Nalisnick et al., 2019] i [Serrà et al., 2019] ovaj pristup nije poručio dobre rezultate.

U [Nalisnick et al., 2019] autori razmatraju pristup temeljen na evaluaciji izglednosti generativnog modela temeljenog na invertirajućem normalizirajućem toku Glow [Kingma and Dhariwal, 2018]. Autori prvo treniraju model na skupu podataka CIFAR10 [Krizhevsky, 2012] pa zatim računaju logaritam izglednost skupa podataka SVHN [Netzer et al., 2011], koji su zapravo izvandistribucijski primjeri. Iznenađujuće, primjeri iz izvandistribucijskog skupa podataka su imali

veću logaritamsku izglednost od primjera iz unutardistribucijskog skupa podataka. Sličnu pojavu autori uočavaju kod autorgresijskog modela te varijacijskog autoenkodera. Konačno, autori nude objašnjenje za uočenu anomaliju u slučaju normalizirajućeg toka, a temelji se na analizi utjecaja transformacije na izglednost. U slučaju normalizirajućeg toka izglednost je jednaka umnošku izglednosti latentne varijable i determinante jakobijana transformacije. Determinanta jakobijana pridjeljuje veću vrijednost primjerima iz izvandistribucijskog skupa podataka što za posljedicu ima veću izglednost izvandistribucijskih primjera. Dodatno, autori zaključuju da maksimizacija logaritma determinante jakobijana transformacije povećava osjetljivost transformacije na perturbacije skupa podataka za učenje.

U [Serrà et al., 2019] autori pokazuju da modeli temeljeni na invertirajućem toku primjerima koji su vizualno jednostavniji pridodaju veću izglednost, bez obzira na skup podataka na kojem je model naučen. Autori izvandistribucijskim primjerima prvo smanje prostorne dimenzije koristeći sažimanje srednjom vrijednosti, a zatim ih naduzorkuju do originalnih dimenzija. Pri sažimanju, korištenje različitih veličina okna utječe na vizualnu složenost konačnog primjera. Zaključno, autori koriste istrenirani model za evaluaciju izglednosti izvandistribucijskih primjera te zaključuju da vizualno najjednostavniji primjeri (nastali korištenjem najvećeg okna) imaju najveću izglednost.

3.3 Napredno korištenje izglednosti

Postoji nekoliko načina kako definirati mjeru koju možemo iskoristiti za detekciju izvandistribucijskih primjera na temelju izglednosti. Općenitu mjeru za detekciju izvandistribucijskih primjera definiramo izrazom:

$$s : X \to \mathbb{R} \tag{13}$$

gdje je $s(\mathbf{x})$ funkcija od primjera $\mathbf{x}$, te sadrži evaluaciju logaritma izglednosti.

[Grathwohl et al., 2019] koristi L2 normu gradijenta logaritma izglednosti po primjeru kao mjeru za detekciju izvandistribucijskih primjera. Mjeru definiramo izrazom:

$$s(\mathbf{x}) = -\|\frac{\partial \log P_\theta(\mathbf{x})}{\partial \mathbf{x}}\|_2 \tag{14}$$

Intuicija iza ovako definirane mjere je sljedeća, izvandistribucijski primjeri s visokom izglednošću su okruženi primjerima s niskom izglednošću. Slijedi da gradijent izvandistribucijskog primjera ima veliki iznos L2 norme gradijenta. Autori tvrde da ova mjera uspješno detektira izvandistribucijske primjere kad se logaritam izglednosti evaluira njihovim modelom temeljenim na energijskoj funkciji, dok egzaktna evaluacija logaritma izglednosti koristeći modele temeljene na normalizirajućim tokovima ne poručuje dobre rezultate.

[Ren et al., 2019] koristi omjer logaritama izglednosti dvaju modela. Jedan model je treniran na primjerima iz skupa podataka, dok drugi model uči isti skup podataka uz dodatak da su primjeri zašumljeni. Mjeru definiramo izrazom:

$$LLR(\mathbf{x}) = \log\frac{P_\theta(\mathbf{x})}{P_{\theta_o}(\mathbf{x})} = \log P_\theta(\mathbf{x}) - \log P_{\theta_o}(\mathbf{x}) \tag{15}$$

gdje je nazivnik model koji je treniran na zašumljenim primjerima. Autori evaluiraju logaritam izglednost koristeći autoregresivne modele. Mana ovakvog pristupa jest potreba za korištenjem dvaju modela.

3.4 Korištenje negativnog skupa podataka

Sljedeći pristup pokušava povećati sposobnost klasifikatora u detekciji izvandistribucijskih primjera korištenjem dodatnog izvandistribucijskog skupa podataka. U [Hendrycks et al., 2019] autori koriste ovaj pristup. Posljedično, gubitak

klasifikatora ima dodatan član koji definiramo kao unakrsnu entropiju između vjerojatnosnog izlaza klasifikatora i uniformne razdiobe za izvandistribucijske primjere. Izraz za gubitak jest:

$$L(\theta) = \mathbb{E}_{(\hat{\mathbf{x}},\hat{y})\sim D_{in}}[-\log P_\theta(y = \hat{y}|\hat{\mathbf{x}})] + \lambda\, \mathbb{E}_{\mathbf{x}\sim D_{out}}[\mathrm{CE}(P_\theta(y|\hat{\mathbf{x}}), U)] \tag{16}$$

gdje CE prestavlja unakrsnu entropiju, a U unifromnu razdiobu. Prema rezultatima, odabir izvandistribucijskih skupova podataka ima veliki utjecaj na konačne performanse klasifikatora. Za detekciju izvandistribucijskih primjera se koristi maksimalna vrijednost softmaksa.

Sličan pristup imaju [Bevandic et al., 2018, Bevandic et al., 2019], gdje autori koriste dva tipa modela za detekciju izvandistribucijskih primjera. Model s jednom glavom detektira izvandistibucijske primjere na temelju maksimalne vrijednosti softmaksa. Tijekom učenja funkcija gubitka ima dodatan član koji zahtjeva da probabilistički izlaz modela teži ka uniformnoj razdiobi za primjere iz izvandistribucijskog skupa podataka. Model s dvije glave ima dodatnu glavu odgovornu za detekciju izvandistribucijskih primjera čiji probabilistički izlaz tumačimo kao vjerojatnost da je dani primjer unutardistribucijski primjer. U fazi učenja funkcija gubitka ima dodatan član koji zahtjeva da probabilistički izlaz modela teži ka nuli za primjere iz negativnog skupa podataka dok za unutardistribucijske primjere teži u jedan. Autori također koriste navedene modele za gustu predikciju slika.

3.5 Korištenje generativnog modela podataka na rubu distribucije

Slično kao prethodni pristup, umjesto dodatnog izvandistribucijskog skupa podataka, možemo iskoristiti generativni model kao generator izvandistribucij-

skih primjera. U [Lee et al., 2018] autori koriste generativni suparnički model (GAN) kao generator primjera s ruba distribucije skupa podataka. Da bi ovaj pristup radio, autori paralelno uče generator, diskriminator i klasifikator. Gubitak klasifikatora ima dodatan član koji pomiče probabilistički izlaz k uniformnoj razdiobi za umjetno generirane primjere s ruba distribucije. Izraz za gubitak klasifikatora definiramo kao:

$$L(\theta) = \mathbb{E}_{(\hat{\mathbf{x}},\hat{y})\sim P_{in}}[-\log P_\theta(y = \hat{y}|\hat{\mathbf{x}})] + \lambda\, \mathbb{E}_{\mathbf{x}\sim P_{out}}[\mathrm{KL}(P_\theta(y|\mathbf{x}), U)] \tag{17}$$

gdje KL predstavlja KL divergenciju, a U unifromnu razdiobu. Generativni model je sposoban generirati primjere s ruba distribucije zbog dodatnog člana u svojoj funkciji gubitka koji je isti kao i dodatan član u gubitku klasifikatora. Posljedično, optimiramo sljedeći izraz:

$$\begin{aligned}\min_\theta \max_\psi \quad & \mathbb{E}_{\mathbf{x}\sim P(\mathbf{x})}[\log D_\psi(\mathbf{x})] + \mathbb{E}_{z\sim P(z)}[\log(1 - D_\psi(G_\theta(z)))] \\ & + \lambda\, \mathbb{E}_{z\sim P(z)}[\mathrm{KL}(P_\theta(y|G(z)), U)]\end{aligned} \tag{18}$$

U ovom radu navedeni pristup poboljšavamo koristeći drugu vrstu generativnog modela.

4 Generativni modeli s normalizirajućim vjerojatnosnim tokom

Normalizirajući vjerojatnosni tokovi su modeli koji se temelje na formuli za promjenu varijable distribucije. Ulazna distribucija prolazi kroz invertibilnu transformaciju, te izlazi kao valjana distribucija latentne varijable. Formula za promjenu varijable distribucije definirana je sljedećom jednadžbom:

$$p_x(\mathbf{x}) = p_z(\mathbf{f}(\mathbf{x})) \left| \det \left(\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} \right) \right| \tag{19}$$

U prikazanoj jednadžbi $|.|$ predstavlja apsolutnu vrijednost, a $\det(X)$ determinantu matrice X. Da bi formula vrijedila funkcija $\mathbf{f}$ mora biti bijekcija, što znači da vrijedi:

$$\mathbf{z} = \mathbf{f}(\mathbf{x}) \equiv \mathbf{x} = \mathbf{f}^{-1}(\mathbf{z}) \tag{20}$$

Posljedica ovako definirane funkcije $\mathbf{f}$ je očuvanje broja dimenzija ulaza. Jednadžba 19 je vrlo intuitivna. S obzirom da distribuciju $p(x)$ transformiramo u neku drugu distribuciju $p(z)$ preko funkcije $\mathbf{f}$, ta funkcija može kontrahirati vjerojatnosni prostor. S obzirom da je vjerojatnosni volumen konstantan, posljedica kontrakcije je promjena gustoće vjerojatnosti. Na primjer, kada uniformnu razdiobu na kontinuiranom intervalu [0, 1] transformiramo funkcijom $Y = 2X$ dobijemo uniformnu razdiobu na intervalu [0, 2]. Vidimo da je funkcija gustoće vjerojatnosti rezultantne distribucije jednaka polovici gustoće vjerojatnosti početne distribucije. Kada govorimo o višedimenzionalnom vjerojatnosnom prostoru, tada promjenu u gustoći vjerojatnosti kompenziramo množenjem s determinantom derivacije funkcije $\mathbf{f}$. Geometrijska interpretacija determinante kvadratne matrice X jest faktor za koji množenje s matricom X kontrahira geometrijski prostor.

Normalizirajući tok nastaje primjenom više invertibilnih transformacija na ulaznu distribuciju što je definirano izrazom:

$$q_0(\mathbf{z}_0) = q_K(\mathbf{z}_K) \prod_{k=1}^{K} \left| \det \left(\frac{\partial \mathbf{f}_k}{\partial \mathbf{z}_{k-1}} \right) \right| \tag{21}$$

gdje vrijedi:

$$\mathbf{z}_K = \mathbf{f}_K \circ ... \circ \mathbf{f}_1(\mathbf{z}_0), \quad \mathbf{z}_0 \sim q_0(\mathbf{z}_0) \tag{22}$$

4.1 Autoregresijski tokovi

Autoregresijski tokovi su podvrsta normalizirajućih tokova. Kod ove vrste tokova želimo povećati ekspresivnost toka vodeći računa o tome da determinanta jakobijana transformacije bude traktabilna. Upravo je traktabilnost izračuna determinante transformacije najveći izazov kod modeliranja arhitekture toka.

Jednadžba proizvoljne transformacije autoregresijskog toka je dana izrazom:

$$\mathbf{z}_i = \mathbf{f}(\mathbf{x}_{1:i}) \tag{23}$$

Ako pretpostavimo da je $\mathbf{x}$ vektor proizvoljne dimenzije D, tada iz navedenog izraza zaključujemo da i-ta dimenzija vektora $\mathbf{z}$ ovisi o prvih i dimenzija ulaznog vektora $\mathbf{x}$. Iz toga slijedi da je derivacija funkcije $\mathbf{f}$ jakobijan donje trokutaste matrice. Determinantu jakobijana je tada vrlo jednostavno izračunati kao umnožak elemenata na dijagonali matrice, što je prikazano izrazom:

$$\det(J) = \prod_{i=1}^{d} J_{ii} \tag{24}$$

Postoji nekoliko vrsta autoregresijskih tokova koje razlikujemo na temelju odabira invertibilne funkcije $\mathbf{f}$.

Masked Autoregressive Flow (MAF) definira invertibilnu funkciju $\mathbf{f}$ kao:

$$\mathbf{z}_1 = \boldsymbol{\mu}_1 + \boldsymbol{\sigma}_1 \mathbf{x}_1 \tag{25}$$

$$\mathbf{z}_i = \boldsymbol{\mu}(\mathbf{z}_{1:i-1}) + \boldsymbol{\sigma}(\mathbf{z}_{1:i-1})\mathbf{x}_i \tag{26}$$

Ovakav odabir transformacije učinkovito evaluira izglednost, ali je uzorkovanje modela izrazito sporo. Za više informacija o ovom modelu čitatelja upućujemo na [Papamakarios et al., 2017].

Inverse Autoregressive Flow (IAF) definira invertibilnu funkciju $\mathbf{f}$ kao:

$$\mathbf{z}_1 = \boldsymbol{\mu}_1 + \boldsymbol{\sigma}_1 \mathbf{x}_i \tag{27}$$

$$\mathbf{z}_i = \boldsymbol{\mu}(\mathbf{x}_{1:i-1}) + \boldsymbol{\sigma}(\mathbf{x}_{1:i-1})\mathbf{x}_i \tag{28}$$

Zbog ovakve formulacije transformacije, uzorkovanje modela je učinkovito. Za više informacija o ovom modelu čitatelja upućujemo na [Kingma et al., 2016].

U okviru ovog rada detaljnije proučavamo tok *Real Non-Volume Preserving Flow* (Real NVP) [Dinh et al., 2017]. Spomenimo još i autoregresijskog tok *Glow* [Kingma and Dhariwal, 2018] koji se temelji na invertibilnim konvolucijskim blokovima s jezgrom veličine 1×1. Slično kao i Real NVP, ovaj tok se sastoji od više slojeva miješanja (*eng.* coupling layer). Sloj miješanja je detaljno objašnjen u nastavku rada.

4.2 Razlike u očuvanju vjerojatnosnog volumena

Determinanta transformacije u izrazu (19) jest koeficijent za koji invertibilna transformacija kontrahira vjerojatnosni prostor. Normalizirajući tok kojem apsolutna vrijednost determinante transformacije iznosi 1 nazivamo tok s očuvanjem

vjerojatnosnog volumena (*eng.* volume preserving flow). Kod takvih tokova evaluacija izglednosti degenerira u:

$$p_x(\mathbf{x}) = p_z(\mathbf{f}(\mathbf{x})) \tag{29}$$

Primjere ovakvih tokova pronalazimo u [Dinh et al., 2015]. Slično, normalizirajući tok kojem apsolutna vrijednost determinante transformacije različita od 1 nazivamo tok bez očuvanja vjerojatnosnog volumena (*eng.* non-volume preserving flow).

4.3 Normalizirajući tok bez očuvanja vjerojatnosnog volumena

Fokusiramo se na vrstu normalizirajućeg autoregresijskog toka koji se naziva Real NVP [Dinh et al., 2017]. Zbog svog dizajna, ovaj model može efikasno izvesti prolaz unaprijed i unatrag te uzorkovati naučenu distribuciju. Osnovnu gradivnu jedinicu ovog modela nazivamo sloj miješanja (*eng.*Coupling layer) te ćemo ga detaljnije proučiti u nastavku.

4.3.1 Sloj miješanja

Pretpostavimo da je ulaz u sloj vektor $\mathbf{x}$ dimenzije D, a izlaz iz sloja vektor $\mathbf{y}$ jednake dimenzije kao i ulazni vektor. Transformacija unutar sloja se provodi na način da odaberemo proizvoljan broj dimenzija d ($d < D$), a zatim odabrane dimenzije propustimo na izlaz. Preostalih $D-d$ dimenzija transformiramo vodeći računa o očuvanju broja dimenzija. Transformacije unutar sloja su definirane formulama 30 i 31.

$$\mathbf{y}_{1:d} = \mathbf{x}_{1:d} \tag{30}$$

$$\mathbf{y}_{d+1:D} = \mathbf{x}_{d+1:D} \odot \exp(\mathbf{s}(\mathbf{x}_{1:d})) + \mathbf{t}(\mathbf{x}_{1:d}) \tag{31}$$

Oznaka $\odot$ predstavlja Hadamardov produkt, a $\mathbf{s}$ i $\mathbf{t}$ mapiraju R^d u R^{D-d}. Ove transformacije imaju jednostavan inverz definiran s:

$$\mathbf{x}_{1:d} = \mathbf{y}_{1:d} \tag{32}$$

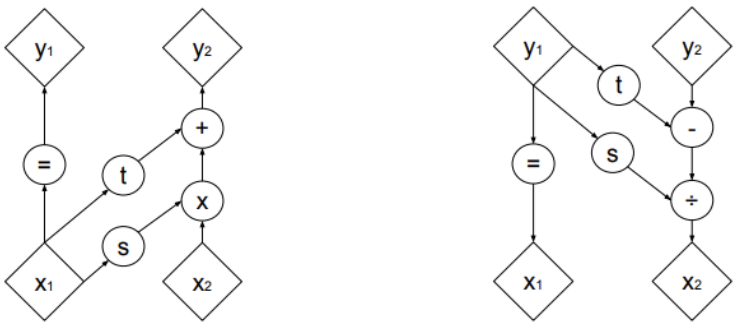
$$\mathbf{x}_{d+1:D} = (\mathbf{y}_{d+1:D} - \mathbf{t}(\mathbf{y}_{1:d})) \odot \exp(-\mathbf{s}(\mathbf{y}_{1:d})) \tag{33}$$

Primjetimo da na kompleksnost inverza ne utječu funkcije $\mathbf{s}$ i $\mathbf{t}$ pa to mogu biti složene funkcije koje ne moraju biti bijekcije. Iz tog razloga se za $\mathbf{s}$ i $\mathbf{t}$ koriste neuronske mreže. Točnije, koristi se proizvoljan broj rezidualnih jedinica s predaktivacijom popularnih ResNet arhitektura [He et al., 2016a].

Zanimljivo je promotriti derivaciju izlaza po ulazu ovog sloja tj. jakobijan koji je definiran sljedećim izrazom:

$$\frac{\delta y}{\delta x} = \begin{bmatrix} I & 0 \\ \frac{\delta y_{d+1:D}}{\delta x_{1:d}} & diag(\exp[s(x_{1:d})]) \end{bmatrix} \tag{34}$$

Uočimo da se još jednom radi o donjoj trokutastoj matrici. Ta informacija će nam dobro doći pri učenju modela. Slika 11 prikazuje računski graf definiranog sloja.

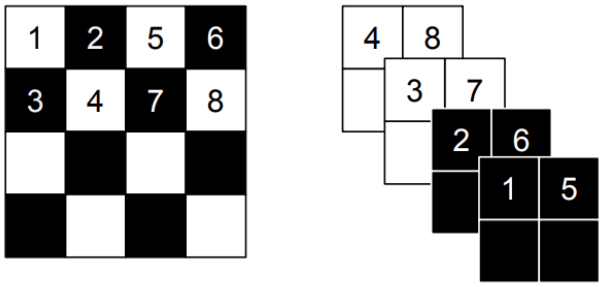


Slika 11: Računski graf sloja. Izvor: [Dinh et al., 2017]

Promotrimo Jakobijan transformacije definiran izrazom 34. Primjećujemo da se determinanta Jakobijana nije jednaka 1 zbog transformacije definirane funkcijom **s**. Ako bismo izbacili izraz koji uključuje funkciju **s** iz transformacije sloja, tada bi naš model degenerirao u model koji čuva vjerojatnosni volumen.

4.3.2 Kompozicija modela Real NVP

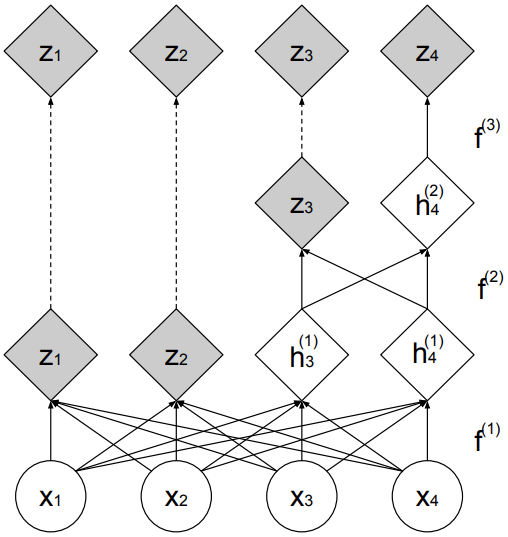
Kako bismo pravilno razumjeli kompozitnu arhitekturu Real NVP-ja, prvo moramo objasniti dva načina odabira dimenzija koje propagiramo bez promjene. Maska šahovske ploče (na slici 12 lijevo) odabire dimenzije koje propagira bez promjene uz uvjet da se ne propagiraju dvije horizontalno ili vertikalno susjedne dimenzije u istom kanalu danog tenzora. Maska po kanalu (na slici 12 desno) propušta čitave kanale tenzora bez promjene. Primjetimo da prije korištenja maske po kanalu tenzor dimenzija $s \times s \times c$ preoblikujemo u tenzor dimenzija $\frac{s}{2} \times \frac{s}{2} \times 4c$.



Slika 12: Dva načina odabira dimenzija koje se propagiraju sljedećem sloju stapanja bez promjene. Izvor: [Dinh et al., 2017]

Real NVP nastaje slaganjem više blokova poduzorkovanja jedan na drugi. Svaki blok poduzorkovanja se sastoji od sljedećeg niza transformacija. Ulazni tenzor prvo prolazi kroz tri sloja stapanja. Između svakog od slojeva stapanja

se odabiru dimenzije koje se propagiraju bez promjene s maskom šahovske ploče vodeći računa da se maska alternira između slojeva. Dobiveni tenzor dimenzija $s \times s \times c$ zatim preoblikujemo u tenzor dimenzija $\frac{s}{2} \times \frac{s}{2} \times 4c$. Sljedeće, tenzor prolazi kroz još tri sloja stapanja. Ovaj put između svakog od slojeva stapanja dimenzije koje se propagiraju bez promjene odabiremo s maskom po kanalu. Konačno, tenzor vraćamo u prvobitan oblik dimenzija $s \times s \times c$. Naglasimo još da se između dvaju blokova poduzorkovanja pola dimenzija izlaznog tezora više ne mijenja, to jest svaki sljedeći blok transformira tenzor s upola manjim brojem dimenzija. Arhitektura Real NVP-ja s više blokova poduzorkovanja nalazi se na slici 13.



Slika 13: Kompozitna arhitektura Real NVP-ja. Izvor: [Dinh et al., 2017]

Propagacija vjerojatnosnog tenzora kroz niz blokova se može opisati sljedećim nizom jednadžbi:

$$h^0 = x \tag{35}$$

$$(z^{i+1}, h^{i+1}) = f^{i+1}(h^i) \tag{36}$$

$$z^L = f^L(h^{L-1}) \tag{37}$$

$$z = (z^1, ..., z^L) \tag{38}$$

gdje f^i predstavlja transformaciju bloka poduzorkovanja. Primjetimo da u jednadžbi (36) pola dimenzija tenzora izostavljamo od daljnje propagacije.

4.3.3 Normalizirajući tok kao generativni model

Generativnim modelima je cilj naučiti distribuciju skupa podataka $p(x)$. Distribucija modela $q(x)$ aproksimira distribuciju skupa podataka $p(x)$. Glavna prednost takvih modela jest što imaju sposobnost generiranja primjera koji su vizualno vrlo slični primjerima iz skupa za učenje. Normalizirajući tok Real NVP možemo promatrati kao generativni model. Model uči aproksimirati distribuciju podataka $p(x)$ tako da je nizom invertibilnih tranformacija preslikava u neku jednostavnu distribuciju latentne varijable. Za distribuciju latentne varijable najčešće koristimo normalnu razdiobom s očekivanjem 0 i varijancom 1.

4.3.4 Učenje modela

Učenje generativnog modela možemo definirati kao minimiziranje KL divergencije između distribucije podataka i distribucije modela:

$$\theta^* = \underset{\theta}{\operatorname{argmin}}\, \mathrm{KL}(P_{data}(\mathbf{x})||P_\theta(\mathbf{x})) \tag{39}$$

što je prema formuli za KL divergenciju ekvivalentno:

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \sum_{\mathbf{x}} P_{data}(\mathbf{x}) \log P_{data}(\mathbf{x}) - \sum_{\mathbf{x}} P_{data}(\mathbf{x}) \log P_\theta(\mathbf{x}) \tag{40}$$

iz čega slijedi:

$$\theta^* = \underset{\theta}{\operatorname{argmin}}\, \mathbb{H}[P_{data}] - \frac{1}{N} \log P_\theta(\mathbf{x}) \tag{41}$$

gdje $\mathbb{H}$ predstavlja entropiju distribucije. S obzirom da je entropija distribucije skupa podataka konstanta, taj član možemo zanemariti. Posljedično, zaključujemo da je minimizacija KL divergencije između distribucije podataka i distribucije modela jednaka maksimizaciji logaritamske izglednosti naučene distribucije modela, što je definirano izrazom:

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \log P_\theta(\mathbf{x}) \tag{42}$$

Logaritamska izglednost je dana sljedećim izrazom:

$$\log(p_x(x)) = \log(p_z(f(x))) + \log\left(\left|\det(\frac{\delta f(x)}{\delta x})\right|\right) \tag{43}$$

Primjetimo da izraz uključuje izračun determinante Jakobijana. To nas ne brine jer je jakobijan svakog sloja donja trokutasta matrica pa je determinanta jednaka umnošku elemenata na dijagonali. Isto tako znamo da je $\det(AB) = \det(A) \cdot \det(B)$ zbog čega se izračun determinante može izvesti sloj po sloj.

4.3.5 Uzorkovanje modela

Uzorkovanje navedenog modela je trivijalno. Podsjetimo se da je model građen od slojeva invertibilnih transformacija koje čuvaju oblik predanog tenzora. Naučenu distribuciju uzorkujemo tako da definiramo vjerojatnosni tenzor proizvoljnih prostornih dimenzija te ga provučemo kroz model u suprotnom smjeru. Tenzor latentne varijable možemo jednostavno uzorkovati jer se latentna varijabla ravna po normalnoj razdiobi s unaprijed definiranim očekivanjem i varijancom. Postupak je definiran sljedećim jednadžbama:

$$\mathbf{z} \sim N(0, I) \tag{44}$$

$$\mathbf{x} = \mathbf{f}^{-1}(\mathbf{z}) \tag{45}$$

Rezultantni tenzor je jednakog oblika kao i predani tenzor. Na primjer, ako želimo generirati sliku dimenzija 64×64 piksela, definiramo tenzor dimenzija $3 \times 64 \times 64$ koji podliježe distribuciji latentne varijable, pa definirani tenzor preslikamo uz pomoć toka u tenzor koji se ravna po distribuciji skupa podataka. Na sličan način, ovaj model može generirati slike proizvoljne veličine. Primjer generiranja slika proizvoljne veličine se nalazi u poglavlju s eksperimentima.

4.3.6 Usporedba s ostalim vrstama generativnih modela

Za razliku od generativnih suparničkih modela, treniranje Real NVP-ja je stabilno. Ipak, slike generirane s GAN-om su kvalitetnije od slika generiranih s Real NVP-em. Za razliku od varijacijskih autoenkodera, Real NVP može egzaktno evaluirati izglednost. U usporedbi s autoregresijskim modelima, uzorkovanje Real NVP modela ima složenost $O(1)$ u ovisnosti o broju dimenzija uzorka, dok je složenost uzorkovanja autoregresijskog modela $O(n)$.

Za razliku od ograničenog Boltzmannovog stroja, distribucija modela normalizirajućeg toka je normalizirana po konstrukciji. Uvođenjem ograničenja da transformacija mora biti invertibilna i čuvati broj dimenzija izbjegavamo potrebu za negativnom fazom pri učenju normalizirajućeg toka.

5 Detekcija izvandistribucijskih primjera

Detekcija izvandistribucijskih primjera se može izvesti na nekoliko načina. Najjednostavniji način uključuje evaluaciju izglednosti pomoću generativnog modela. Ipak, najčešće želimo naučiti diskriminativni model sposoban točno klasificirati unutardistribucijske primjere te detektirati izvandistribucijske primjere. Diskriminativni model učen unakrsnom entropijom nema dobre performanse u potonjem zadatku. Iz tog razloga gubitku diskriminativnog modela dodajemo dodatan član koji tjera probabilistički izlaz modela k uniformnoj razdiobi za primjere iz negativnog skupa podataka ili za primjere s ruba distribucije podataka.

5.1 Izglednosti kao detektor izvandistribucijskih primjera

Najjednostavniji i intuitivan pristup detekciji izvandistribucijskih primjera jest primjena izglednosti. Generativni model naučimo na odabranom skupu podataka. Intuitivno, naučeni model bi trebao imati visoku izglednost za primjere koji podliježu naučenoj distribuciji podataka, dok bi za ostale primjere izglednost trebala biti mala. Dizajniranje generativnog modela koji bi ispunio navedene pretpostavke je aktivno područje istraživanja. Trenutno, autori najčešće koriste autoregresivne modele te modele temeljene na normalizirajućim tokovima zbog mogućnosti egzaktne evaluacije logaritma izglednosti.

[Nalisnick et al., 2019] i [Serrà et al., 2019] pokazuju da izglednost može biti visoka i za izvandistribucijske primjere. Izglednost evaluiraju koristeći autoregresijski tok Glow [Kingma and Dhariwal, 2018] analizirajući skup CIFAR10 [Krizhevsky, 2012] kao unutardistribucijski skup, a SVHN [Netzer et al., 2011] kao izvandistribucijski skup podataka. Dodatno, pokazuju da iznos izglednosti ovisi o vizualnoj složenosti primjera, pri čemu jednostavniji primjeri imaju veću logaritamsku izglednost iako se radi o izvandistribucijskim primjerima. Proveli

smo sličan eksperiment koristeći autoregresijski tok Real NVP i detektirali slično ponašanje.

5.2 Detekcija izvandistribucijskih primjera uz pomoć uzorkovanja s ruba distribucije

Poznati problem u području dubokog učenja jest velika samouvjerenost diskriminativnih modela. To znači da primjere koji su daleko od primjera iz skupa za učenje, naučeni diskriminativni model klasificira u određenu klasu s visokom sigurnošću. Ovu pojavu ne smijemo brkati s težnjom da duboki modeli imaju što veću generalizacijsku sposobnost. Činjenica jest da težimo povećati generalizacijsku sposobnost diskriminativnih modela odnosno klasifikatora ali samo na primjerima koji podliježu distribuciji odabranoga skupa podataka. Sve ostale primjere bi diskriminativni model trebao klasificirati s malom sigurnošću. Kada bi to bio slučaj, tada bismo na temelju sigurnosti klasifikatora pojedine primjere odbili klasificirati tj. označili ih kao izvandistribucijske primjere.

Fokusiramo se na problem klasifikacije primjera u K klasa. Osim što želimo naučiti klasifikator klasificirati zadane primjere sa što boljom točnošću, želimo izvandistribucijske primjere klasificirati s niskom sigurnošću. Takvo ponašanje postižemo uvođenjem primjera s ruba distribucije u postupak učenja klasifikatora. Prema [Lee et al., 2018], optimiramo sljedeći gubitak:

$$L(\theta) = \mathbb{E}_{P_{in}(\hat{x},\hat{y})}[-\log P_\theta(y = \hat{y}|\hat{x})] + \beta\, \mathbb{E}_{P_{out}(x)}[\mathrm{KL}(U||P_\theta(y|x))] \tag{46}$$

KL predstavlja Kullback-Leiblerovu divergenciju, U predstavlja uniformnu razdiobu dok parametar β mora biti veći od 0.

Osim standardnog gubitka unakrsne entropije, navedeni gubitak ima dodatni član kojim se potiče da klasifikator na izlazu daje uniformnu razdiobu za pri-

mjere s ruba distribucije. Takvi primjeri su generirani uz pomoć generativnog modela, stoga ih nazivamo sintetičkim primjerima s ruba distribucije. U originalnom radu [Lee et al., 2018] koristi se generativni suparnički model DCGAN [Radford et al., 2016]. Naše poboljšanje metode umjesto GAN-a koristi generativni model Real NVP, čime bitno pojednostavljujemo proceduru za učenje definiranu u [Lee et al., 2018]. Poboljšanje metode je definirano algoritmom:

Algoritam 1: Procedura za detekciju i generiranje izvandistribucijskih primjera.

Zahtjevaj $\beta > 0$

Definiraj Real NVP: $\mathbf{z} = \mathbf{f}_{\theta_R}(\mathbf{x}), \mathbf{x} = \mathbf{f}_{\theta_R}^{-1}(\mathbf{z})$

Definiraj Klasifikator: $P_{\theta_C}(y|\mathbf{x})$

Definiraj $Optimizator_{\mathrm{R}}, Optimizator_{\mathrm{C}}$

repeat

$\mathbf{x}$, y = obtain_minibatch()

$\mathbf{z}$ = sample $N(0, 1)$

$L_{\mathrm{cls}} = -\mathbb{E}[logP_{\theta_C}(y|\mathbf{x})] + \beta_{\mathrm{cls}}\,\mathbb{E}[\mathrm{KL}(U||P_{\theta_C}(y|\mathbf{f}_{\theta_R}^{-1}(\mathbf{z})))]$

$L_{\mathrm{rnvp}} = -\mathbb{E}[log(p_z(\mathbf{f}_{\theta_R}(\mathbf{x}))) + log\left(\left|det(\frac{\delta \mathbf{f}_{\theta_R}(\mathbf{x})}{\delta \mathbf{x}})\right|\right)]$

$\theta_{\mathrm{R}} += Optimizator_{\mathrm{R}}.update(\nabla L_{\mathrm{rnvp}})$

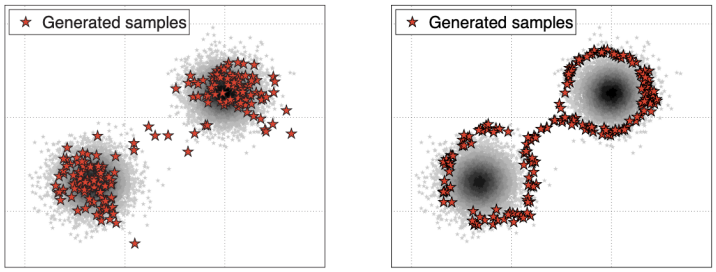
$\theta_{\mathrm{C}} += Optimizator_{\mathrm{C}}.update(\nabla L_{\mathrm{cls}})$

until *maksimalan broj iteracija*

Klasifikator i generativni model rade u sprezi, zajedno optimirajući član gubitka koji potiče visoku nesigurnost za izvandistribucijske primjere. Dodatni član u funkciji gubitka tjera probabilistički izlaz klasifikatora bliže uniformnoj razdiobi za izvandistribucijske primjere. U slučaju Real NVP-ja dodatan član u formuli gubitka omogućava modelu generiranje primjera s ruba distribucije. Hiperpara-

metrom β kontroliramo utjecaj dodanog člana na gubitak.

Slika 14a prikazuje generirane unutardistribucijske primjere označene crvenom bojom. Slika 14b prikazuje generirane primjere s ruba distribucije označene crvenom bojom. U oba slučaja su primjeri iz skupa za učenje označeni sivom bojom.



(a) (b)

Slika 14: Generirani unutardistribucijski primjeri (a) i generirani primjeri s ruba distribucije (b). Generirani primjeri su nastali uzorkovanjem dva različita modela. Izvor: [Lee et al., 2018]

U radu [Hendrycks et al., 2019] autori kao izvandistribucijske primjere koriste primjere iz drugih poznatih skupova podataka. U slučaju kad je izvandistribucijski skup podataka daleko od distribucije podataka skupa za učenje, takvi primjeri ne doprinose poboljšanju performansi klasifikatora u detekciji izvandistribucijskih primjera [Hendrycks et al., 2019]. Slično, u [Bevandic et al., 2019] autori koriste izvandistribucijski skup podataka pri učenju modela za gustu predikciju s dodatnom glavom koja služi za detekciju izvandistribucijskih dijelova slike.

Kako bismo uspješno detektirali izvandistribucijske primjere, potrebno je klasifikator sposoban klasificirati primjere u K klasa transformirati u detektor izvandistribucijskih primjera. Najčešće, vjerojatnost da je dani primjer unutardistribucijski primjer definiramo kao maksimalan iznos probabilističkog izlaza modela

(max-softmax) [Hendrycks and Gimpel, 2017]. Pretpostavimo da se unutardistribucijski primjeri označavaju s 1, a izvandistribucijski primjeri s 0. Sljedeći izraz definira transformaciju klasifikatora u detektor izvandistribucijskih primjera.

$$P_\theta(c = 1|\hat{x}) = \max_y P_\theta(y|\hat{x}) \tag{47}$$

Vjerojatnost da je dani primjer unutardistribucijski primjer možemo definirati i na temelju entropije probabilističkog izlaza klasifikatora:

$$P_\theta(c = 1|\hat{x}) = 1 - \frac{1}{\log K}\,\mathbb{H}[P_\theta] \tag{48}$$

[Hendrycks et al., 2019] prijavljuje da korištenje entropije ima bolje performanse, ali korištenje maksimalne vrijednosti probabilističkog izlaza modela je postalo standard u literaturi. U provedenim eksperimentima koristimo oba pristupa, ali prednost dajemo maksimalnoj vrijednosti probabilističkog izlaza.

5.3 Generiranje primjera s ruba distribucije podataka

Definiramo dva načina generiranja primjera s ruba distribucije. Rezultat obje metode je generativni model koji je sposoban generirati sintetičke primjere s ruba distribucije. Prvi način se temelji na algoritmu 1, gdje generativni i diskriminativni model rade u sprezi. Drugi način se temelji na generativnom modelu i dodatnom negativnom skupu podataka.

5.3.1 Suradnja generativnog i diskriminativnog modela

Ako promijenimo perspektivu i pretpostavimo da nam je glavni cilj generirati primjere s ruba distribucije, tada nam performanse klasifikatora mogu dati dobru ideju o kvaliteti i položaju izvandistribucijskih primjera. Pretpostavimo da imamo klasifikator koji je istreniran na promatranom skupu podataka uz pomoć gubitka unakrsne entropije. Tada performansa klasifikatora može dati signal o

položaju primjera generiranih s Real NVP-om. Naime, ako performansa klasifikatora padne za veliki broj postotnih bodova, tada su sintetički generirani primjeri preblizu primjerima iz skupa podataka. Razlog pada performansi klasifikatora jest to što za sintetičke primjere gubitak klasifikatora tjera probabilistički izlaz klasifikatora na uniformnu razdiobu. S obzirom da su u ovom slučaju sintetički primjeri preblizu primjerima iz skupa podataka za učenje, performanse klasifikatora padaju. U slučaju kada performanse klasifikatora ne opadaju, to može biti znak da su sintetički primjeri daleko od ruba distribucije. Sintetički izvandistribucijski primjeri daleko od ruba distribucije nam nisu zanimljivi. Prema [Lee et al., 2018], sintetički primjeri daleko od ruba distribucije ne doprinose poboljšanju klasifikatora u detekciji izvandistribucijskih primjera.

5.3.2 Učenje generativnog modela na pozitivnim i negativnim podacima

Algoritam 1 koristi diskriminativni model te je zbog toga pogodan samo za označene skupove podataka. U slučaju neoznačenog skupa podataka odnosno u nenadziranom pristupu, model sposoban generirati primjere s ruba distribucije podataka učimo korištenjem dodatnog izvanidstribucijskog skupa podataka. Gubitak generativnog modela ima dodatan član koji zahtjeva da logaritam izglednosti primjera iz izvandistribucijskog skupa podataka bude veći od logaritma izglednosti skupa podataka za učenje. Gubitak je definiran izrazom:

$$L(\theta) = \mathbb{E}_{\mathbf{x}_{in} \sim P_{in}}[-\log P_\theta(\mathbf{x}_{in}) + \lambda\, \mathbb{E}_{\mathbf{x}_{out} \sim P_{out}}[\max(\log P_\theta(\mathbf{x}_{in}) - \log P_\theta(\mathbf{x}_{out}), 0)]] \quad (49)$$

gdje je $\max(x, y)$ maksimalna vrijednost između dvije ponuđene. Hiperparametar λ je veći od nula. Ovaj gubitak pretpostavlja generativni model sposoban egzaktno evaluirati logaritam izglednosti. U našim eksperimentima koristimo ge-

nerativni model Real NVP, te dokazujemo da su generirani primjeri uistinu na rubu distribucije skupa za učenje. Primjetimo da smo nedostatak označenih primjera nadoknadili korištenjem dodatnog (također neoznačenog) skupa podataka.

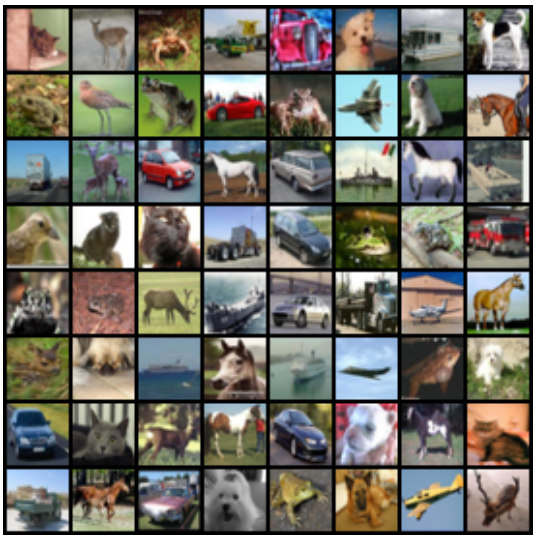
6 Eksperimenti

6.1 Skupovi podataka

U provedenim eksperimentima koristili smo nekoliko skupova podataka. Skupove podataka CIFAR10 [Krizhevsky, 2012], SVHN [Netzer et al., 2011], Tiny-ImageNet [Deng et al., 2009] i LSUN [Yu et al., 2015] opisujemo u nastavku.

6.1.1 CIFAR10

CIFAR10 [Krizhevsky, 2012] je skup podataka koji se sastoji od 60 000 slika rezolucije 32×32 piksela raspoređenih u 10 klasa. 50 000 Slika se nalazi u skupu za učenje, dok se 10 000 slika nalazi u skupu za validaciju. Klase slika su: avion, automobil, ptica, mačka, jelen, pas, žaba, konj, brod i kamion. Slika 15 prikazuje primjere slika iz ovog skupa podataka.



Slika 15: Primjeri slika iz skupa podataka CIFAR10.

6.1.2 SVHN

Skup podataka SVHN [Netzer et al., 2011] (*eng.* Street View House Numbers) se sastoji od 99 289 slika kućnih brojeva podijeljenih u skup za učenje (73 257 slika) i skup za validaciju (26 032 slike). Slike su podijeljene u 10 klasa, a svaka klasa predstavlja broj od 0 do uključivo 9. Slika 16 prikazuje primjere iz navedenog skupa podataka.

Slika 16: Primjeri slika iz skupa podataka SVHN.

6.1.3 LSUN

Skup podataka LSUN [Yu et al., 2015] (*eng.* Large-Scale UNderstaning) jest skup podataka koji se sastoji od 10 milijuna slika raspoređenih u 10 klasa. Odabrane klase su: spavaća soba, most, crkva na otvorenom, učionica, konferencijska soba, blagovaonica, kuhinja, dnevna soba, restoran i toranj. Slike su različitih dimenzija. U provedenim eksperimentima koristimo samo dio skupa podataka koji se sastoji od 10 000 testnih slika smanjenih na rezoluciju 32×32 piksela [Liang et al., 2018]. Slika 17 prikazuje primjere slika iz skupa podataka LSUN.

Slika 17: Primjeri slika iz skupa podataka LSUN.

6.1.4 Tiny-Imagenet

Skup podataka Tiny-Imagenet se sastoji od 110 000 slika raspoređenih u 200 klasa. 100 000 slika se koristi za učenje dok se 10 000 slika nalazi u skupu za validaciju. Slika 18 prikazuje primjere slika iz skupa podataka Tiny-Imagenet.

Slika 18: Primjeri slika iz skupa podataka Tiny-Imagenet.

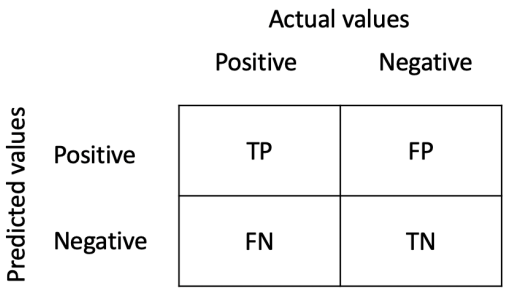
6.2 Metrike

Kako bismo pravilno evaluirali provedene eksperimente, koristimo standardizirane metrike koje nam daju bolji uvid u performanse modela. Korištene metrike objašnjavamo u nastavku rada.

6.2.1 Matrica zabune

Matrica zabune predstavlja intuitivan način prikaza rezultata klasifikacije. Ako se fokusiramo na binarnu klasifikaciju, tada vrijednosti matrice zabune definiramo na sljedeći način. Točno klasificirane pozitivne primjere nazivamo stvarni pozitiv (*eng.* true positive - TP), točno klasificirane negativne primjere nazivamo stvarni negativ (*eng* true negative - TN), netočno klasificirane pozitivne primjere nazivamo lažni negativ (*eng.* false negative - FN) i konačno netočno klasificirane negativne primjere nazivamo lažni pozitiv (*eng.* false positive - FP). Slika 19 prikazuje matricu zabune za binarnu klasifikaciju.

S matricom zabune povezujemo nekoliko mjera koje nam daju uvid u kvalitetu klasifikatora. U ovom radu definiramo mjere: točnost, preciznost, odziv, fall-out, specifičnost i F1 mjeru.



Slika 19: Matrica zabune.

Točnost (*eng.* accuracy) definiramo kao udio točno klasificiranih primjera u skupu svih primjera. Naglasimo da točnost nije dobra mjera u slučaju neurav-

noteženih klasa. Točnost računamo izrazom:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{50}$$

Preciznost (*eng.* precision) definiramo kao udio pozitivno klasificiranih primjera u skupu pozitivno klasificiranih primjera. Preciznost računamo izrazom:

$$P = \frac{TP}{TP + FP} \tag{51}$$

Odziv (*eng.* recall, true positive rate) definiramo kao udio pozitivno klasificiranih primjera u skupu svih pozitivnih primjera. Odziv računamo izrazom:

$$R = TPR = \frac{TP}{TP + FN} \tag{52}$$

Fall-out (false positive rate) definiramo kao udio primjera pogrešsno proglašsenih pozitivnim. Fall-out računamo izrazom:

$$FPR = \frac{FP}{FP + TN} \tag{53}$$

Specifičnost (*eng.* (specificit) definiramo kao udio negativno klasificiranih primjera u skupu svih negativnih primjera. Specifičnost računamo izrazom:

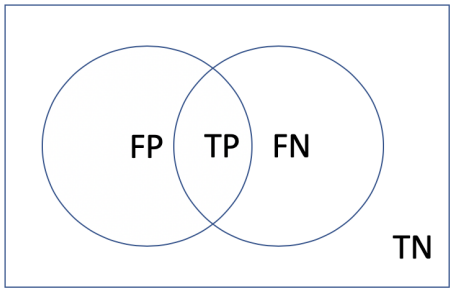
$$S = \frac{TN}{FP + TN} \tag{54}$$

F1 mjera je harmonijska sredina preciznosti i odziva. Ova mjera nam daje dobar uvid u performanse klasifikatora u slučaju neuravnoteženih klasa. F1 mjeru

računamo izrazom:

$$F_1 = \frac{2PR}{P+R} \tag{55}$$

Matricu zabune možemo prikazati i preko Vennovog dijagrama:



Slika 20: Vennov dijagram s elementima iz matrice zabune.

Sljedeće mjere računamo za binarni klasifikator s pomičnim klasifikacijskim pragom. Klasifikator označava primjer u klasu 1 kada je sigurnost klasifikatora veća od nekog praga δ. U suprotom, primjer je klasificiran u klasu 0. Ovakvu klasifikaciju definiramo izrazom:

$$g(\mathbf{x}) = \begin{cases} 1 & \text{ako} \quad s(\mathbf{x}) \geqslant \delta \\ 0 & \text{inače} \end{cases} \tag{56}$$

gdje je $s(\mathbf{x})$ probabilistički izlaz klasifikatora, $q(\mathbf{x})$ mapira primjere u vrijednost iz skupa $\{0, 1\}$, a δ pomični prag.

6.2.2 FPR at 95% TPR

Klasifikatoru s pomičnim klasifikacijskim pragom postavljamo prag na vrijednost za koju je odziv jednak 95% te uspoređujemo performanse na temelju specifičnosti. Bolji klasifikator ima veću mjeru specifičnosti. Ovu mjeru možemo

interpretirati kao vjerojatnost da se izvandistribucijski primjer klasificira u točnu klasu kada odziv (TPR) iznosi 95%.

6.2.3 AUROC

AUROC (*eng.* Area Under the Receiver Operating Characteristic) je mjera za površinu ispod ROC krivulje. ROC krivulja definira vezu između odziva i fall-outa, tj. između TPR i FPR. Mjeru AUROC možemo interpretirati kao vjerojatnost da unutardistribucijski primjer ima veću vjerojatnost detekcije nego izvandistribucijski primjer.

6.2.4 Točnost klasifikacije

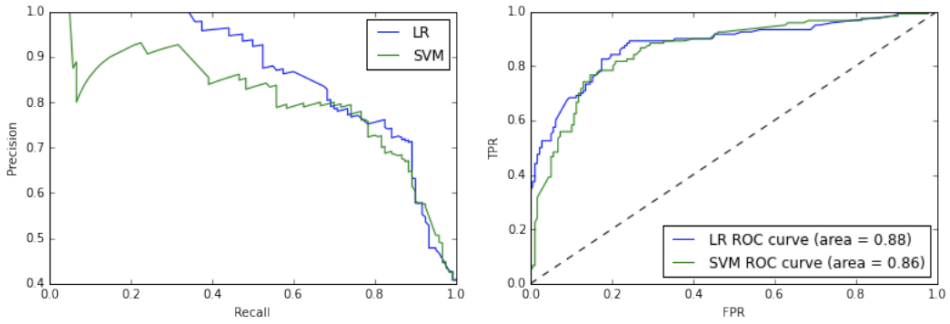
Točnost klasifikacije (*eng.* detection accuracy) je mjera koja odgovara iznosu najveće vjerojatnosti točne detekcije izvandistribucijskih primjeraka za neki prag δ uz pretpostavku da je jednak udio unutar- i izvandistribucijskih primjera u testnom skupu podataka. Mjeru računamo prema izrazu:

$$1 - \min_{\delta}\{P_{in}(s(\mathbf{x}) \leqslant \delta)P(\mathbf{x} \text{ iz } P_{in}) + P_{out}(s(\mathbf{x}) > \delta)P(\mathbf{x} \text{ iz } P_{out})\} \qquad (57)$$

gdje P_{in} predstavlja unutardistribucijske primjere, P_{out} izvandistribucijske primjere.

6.2.5 AUPR

AUPR (*eng.* Area under the Precision-Recall curve) mjeri površinu ispod PR krivulje. PR krivulja definira odnos između preciznosti i odziva za razne vrijednosti klasifikacijskog praga δ. U eksperimentima koristimo mjere AUPR-In i AUPR-Out a razlika jest u tome što za AUPR-In definiramo unutardistribucijske primjere kao pozitivne dok za mjeru AUPR-Out definiramo izvandistribucijske primjere kao pozitivne.



Slika 21: Na lijevoj slici se nalazi PR krivulja, dok na desnoj slici vidimo ROC krivulju. Izvor: Strojno učenje, Sažeci predavanja, FER Zagreb.

6.3 Generiranje primjera modelom Real NVP

U ovom potpoglavlju prikazujemo eksperimente u kojima koristimo generativni model Real NVP kao generator unutardistribucijskih primjera. Eksperimenti prikazuju rezultate generiranje slika različitih rezolucija. Real NVP generira mutnije primjere od generativnog suparničkog modela.

Kvalitetu primjera mjerimo evaluiranjem logaritma izglednosti. Vizualno bolji primjeri imaju veći logaritam izglednosti. S obzirom da je log-izglednost koju računa Real NVP izračunata nad diskretnim vrijednostima, dobivena vrijednost nije direktno usporediva s vrijednostima drugih generativnih modela. Iz tog razloga, slično kao [Dinh et al., 2017], uvodimo mjeru broj bitova po dimenziji (*eng.* bits per dimension):

$$BPD(\mathbf{x}) = \frac{-(\frac{\log p_\theta(\mathbf{x})}{3 \cdot 32 \cdot 32} - \log 128)}{\log 2} \tag{58}$$

6.3.1 Generiranje umjetnih unutardistribucijskih primjera

U ovom eksperimentu smo trenirali model Real NVP na skupovima podataka CIFAR10 i SVHN. Model je učen na način da maksimiziramo logaritam izgled-

nosti podataka iz skupa za učenje. Model se sastoji od 2 rezidualna bloka s 32 skrivene mape značajki u prvom sloju miješanja. Poduzorkovanje je provedeno 3 puta.

Slika 22 prikazuje umjetno generirane unutardistribucijske primjere iz skupa podataka CIFAR10. Model je učen na svim klasama skupa podataka istovremeno.

Slika 22: Generirani primjeri iz skupa podataka CIFAR10.

Slika 23: Generirani primjeri iz skupa podataka SVHN.

Slika 23 prikazuje umjetno generirane unutardistribucijske primjere iz skupa podataka SVHN. Model se sastoji od 2 rezidualna bloka s 32 skrivene mape značajki u prvom sloju miješanja. Poduzorkovanje je provedeno 3 puta. Model je učen na svim klasama skupa podataka istovremeno.

6.3.2 Generiranje umjetnih unutardistribucijskih primjera velike rezolucije

U ovom eksperimentu koristimo generativni model Real NVP za generiranje slika veće rezolucije od slika korištenih u skupu za učenje. Odabrani skup podataka je CIFAR10. Koristimo model iz prethodnog eksperimenta.

Slike 24 i 25 prikazuju primjere dvostruko odnosno četverostruko veće rezolucije (64×64 i 128×128 piksela). Uvećanje se postiže tako da definiramo tenzor željene dimenzije koji se ravna po distribuciji latentne varijable (najčešće normalna razdioba), pa zatim provučemo kroz invertiranu transformaciju. Postupak je detaljnije opisan u potpoglavlju 4.3.5.

Slika 24: Generirani primjeri iz skupa podataka CIFAR10 (uvećanje 2×).

Slika 25: Generirani primjeri iz skupa podataka CIFAR10 (uvećanje 4×).

Slike 26 i 27 prikazuju primjere dvostruko odnosno četverostruko veće rezolucije. Model je učen na skupu podataka SVHN.

Slika 26: Generirani primjeri iz skupa podataka SVHN (uvećanje 2×).

Slika 27: Generirani primjeri iz skupa podataka SVHN (uvećanje 4×).

6.4 Detekcija izvandistribucijskih primjera uz pomoć izglednosti

Slično kao [Nalisnick et al., 2019] i [Serrà et al., 2019], u ovom potpoglavlju prikazujemo eksperimente u kojima pokušavamo detektirati izvandistribucijske primjere pomoću izglednosti. Izglednost egzaktno evaluiramo uz pomoć generativnog modela Real NVP.

6.4.1 Rezultati na skupu podataka CIFAR10

Model Real NVP koji koristimo u ovom eksperimentu se sastoji od 2 rezidualna bloka s 32 skrivene mape značajki u prvom sloju miješanja. Poduzorkovanje se provodi 3 puta. Model prvo treniramo na skupu podataka CIFAR10, a zatim računamo logaritam izgldnosti skupova podataka LSUN, Tiny-Imagenet i SVHN koje koristimo kao izvandistribucijske skupove podataka. U tablici 1 vidimo vrijednosti logaritma izglednosti za navedene skupove podataka. Iz dobivenih rezultata vidimo da primjeri SVHN skupa podataka imaju veći logaritam izglednost od primjera iz skupa CIFAR10.

Tablica 1: Log izglednost izvandistribucijskih skupova podataka. Model je treniran na skupu podataka CIFAR10.

Skup podataka	Bitova/dimenzija	Log-izglednost
CIFAR10 (inlier)	3.621	9324.226
SVHN	2.474	11766.116
LSUN 32x32	4.880	6642.587
Tiny-Imagenet	4.890	6623.051

6.4.2 Rezultati na skupu podataka SVHN

U sljedećem slučaju model Real NVP učimo na skupu podatka SVHN, a zatim računamo logaritam izglednost za izvandistribucijske skupove podataka CIFAR10, LSUN i Tiny-Imagenet. Rezultati se nalaze u tablici 2. Iz tablica 1 i 2 možemo zaključiti da izglednost ovisi o vizualnoj kompleksnosti primjera nego o skupu podataka koji je korišten za učenje modela. Dobiveni rezultati se slažu s rezultatima u [Nalisnick et al., 2019] i [Serrà et al., 2019].

Tablica 2: Log izglednost izvandistribucijskih skupova podataka. Model je treniran na skupu podataka SVHN.

Dataset	Bitova/dimenzija	Log likelihood
SVHN (inlier)	2.194	12363.104
CIFAR10	4.086	8334.588
LSUN 32x32	7.227	1645.605
Tiny-Imagenet	7.343	1397.920

6.5 Detekcija izvandistribucijskih primjera pomoću diskriminativnog modela

U ovom potpoglavlju prikazujemo eksperimente u kojima treniramo diskriminativni model sposoban za detekciju izvandistribucijskih primjera. Prikazujemo rezultate nekoliko metoda uključujući i bazni model učen unakrsnom entropijom. Rezultati uključuju performanse diskriminativnog modela u klasifikaciji unutardistribucijskog skupa podataka te detekciju izvandistribucijskih primjera. U našim eksperimentima koristimo metrike slične kao i [Liang et al., 2018].

Korištene metrike su opisane u potpoglavlju 6.2. Detekcija izvandistribucijskih primjera se svodi na binarnu diskriminaciju gdje se unutardistribucijski primjeri označavaju s 1, a izvandistribucijski primjeri s 0. Izlaz modela koji klasificira primjere u K klasa transformiramo u binarni klasifikator, tj. detektor izvandistribucijskih prema jednadžbama (47) i (48). U svim eksperimentima koristimo model ResNet18 [He et al., 2016a] kao klasifikator.

6.5.1 Rezultati na skupu podataka SVHN

Tablica 3 prikazuje rezultate za klasifikator učen gubitkom unakrsne entropije na skupu podataka SVHN. Skupovi podataka CIFAR10, LSUN i Tiny-Imagenet služe kao izvandistribucijski skupovi podataka. Rezultati ovog modela služe kao referentna točka te ih želimo poboljšati. Model je transformiran u detektor izvandistribucijskih primjera koristeći maksimalnu vrijednost softmaksa (izraz 47).

Tablica 3: Performanse klasifikatora ResNet18 [He et al., 2016a] učenog s gubitkom unakrsne entropije. Klasifikator je transformiran u detektor izvandistribucijskih primjera koristeći maksimalnu vrijednost softmaksa (izraz 47). Dobivene vrijednosti koristimo kao referentnu točku za usporedbu svih ostalih metoda u klasifikaciji unutardistribucijskih primjera i detekciji izvandistribucijskih primjera.

Unakrsna entropija			
Točnost na skupu SVHN: 91.91			
OOD dataset	CIFAR10	LSUN	Tiny-Imagenet
TNR at TPR 95%	37.52	36.62	38.54
AUROC	89.92	89.53	89.98
Točnost detekcije	84.03	83.68	84.17
AUPR-In	91.85	91.46	91.78
AUPR-Out	86.74	86.30	86.87

Tablica 4 prikazuje performanse klasifikatora treniranog procedurom definiranom u [Lee et al., 2018]. Ovako treniran klasifikator ima bolje performanse u detekciji izvandistribucijskih primjera od baznog klasifikatora (tablica 3). Uočavamo pad klasifikacijske sposobnosti ovako treniranog klasifikatora u odnosu na bazni model. Za učenje ovog modela koristimo javno dostupan kod[1].

Tablica 4: Performanse ResNet18 [He et al., 2016a] klasifikatora učenog s kompleksnim gubitkom definiranim u [Lee et al., 2018]. Klasifikator je transformiran u detektor izvandistribucijskih primjera koristeći maksimalnu vrijednost softmaksa (izraz 47). Vrijednosti u zagradama predstavljaju razliku u odnosu na bazni model (tablica 3). Opažamo smanjenje klasifikacijskih sposobnosti klasifikatora uz povećanje performansi u detekciji izvandistribucijskih primjera.

Kompleksni gubitak + GAN [Lee et al., 2018]			
Točnost na skupu SVHN: 90.05 (-1.86)			
OOD dataset	CIFAR10	LSUN	Tiny-Imagenet
TNR at TPR 95%	46.48 (+8.96)	45.33 (+8.71)	45.69 (+7.15)
AUROC	90.02 (+0.10)	89.68 (+0.15)	90.00 (+0.02)
Točnost detekcije	83.19 (-0.84)	82.68 (-1.00)	83.16 (-1.01)
AUPR-In	90.83 (-1.02)	90.70 (-0.76)	91.06 (-0.72)
AUPR-Out	87.86 (+1.12)	87.42 (+1.12)	87.68 (+0.81)

Tablica 5 prikazuje performanse klasifikatora treniranog našom metodom. Klasifikator učen našom metodom (Algoritam 1) ima bolje performanse u detekciji izvandistribucijskih primjera od baznog modela. Uočavamo da ovako treniran klasifikator gubi dio klasifikacijske sposobnosti. Arhitektura modela Real NVP

[1] `https://github.com/alinlab/Confident_classifier`

korištenog u ovom eksperimentu ima 2 rezidualna bloka s 32 mape značajki u prvom sloju. Poduzorkovanje se provodi tri puta. Parametar β je postavljen na 5. Veličina minigrupe u fazi učenja je 128.

Tablica 5: Performanse ResNet18 [He et al., 2016a] klasifikatora učenog našom metodom (Algoritam 1). Klasifikator je transformiran u detektor izvandistribucijskih primjera koristeći maksimalnu vrijednost softmaksa (izraz 47). Vrijednosti u zagradama predstavljaju razliku u odnosu na bazni model (tablica 3). Opažamo znatno smanjenje klasifikacijskih sposobnosti klasifikatora uz osjetno povećanje performansi u detekciji izvandistribucijskih primjera.

Kompleksni gubitak + Real NVP			
Točnost na skupu SVHN: 88.01 (-3.90)			
OOD dataset	CIFAR10	LSUN	Tiny-Imagenet
TNR at TPR 95%	63.53 (+26.01)	60.39 (+23.77)	61.30 (+22.76)
AUROC	93.76 (+3.84)	93.03 (+3.50)	93.43 (+3.45)
Točnost detekcije	86.98 (+2.95)	86.24 (+2.56)	86.65 (+2.48)
AUPR-In	94.54 (+2.69)	93.83 (+2.37)	94.37 (+2.59)
AUPR-Out	92.48 (+5.74)	91.65 (+5.35)	91.93 (+5.06)

Tablica 6 prikazuje rezultate klasifikatora učenog sa sintetičkim primjerima s ruba distribucije podataka. Sintetički primjeri s ruba distribucije su dobiveni koristeći gubitak definiran izrazom (49), gdje je skup podataka TrafficSign korišten kao izvandistribucijski skup podataka. Arhitektura modela Real NVP korištenog u ovom eksperimentu ima 3 rezidualna bloka s 32 mape značajki u prvom sloju. Poduzorkovanje se radi tri puta. Parametar λ je postavljen na vrijednost 20. Pri učenju klasifikatora parametar β postavljamo na vrijednost 1. Parametri modela

su nasumično inicijalizirani. Veličina minigrupe u fazi učenja je 128. Ova metoda postiže bolje rezultate u detekciji izvandistribucijskih primjera od baznog klasifikatora uz smanjenje klasifikacijske sposobnosti (tablica 3).

Tablica 6: Performanse ResNet18 [He et al., 2016a] klasifikatora učenog optimiranjem gubitka definiranog izrazom 46. Klasifikator je transformiran u detektor izvandistribucijskih primjera koristeći maksimalnu vrijednost softmaksa (izraz 47). Primjeri s ruba distribucije su dobiveni optimiranjem gubitka (49). Vrijednosti u zagradama predstavljaju razliku u odnosu na bazni model (tablica 3). Opažamo znatno smanjenje klasifikacijskih sposobnosti klasifikatora uz osjetno povećanje performansi u detekciji izvandistribucijskih primjera.

Primjeri s ruba distribucije (49)			
Točnost na skupu SVHN: 87.45 (-4.46)			
OOD dataset	CIFAR10	LSUN	Tiny-Imagenet
TNR at TPR 95%	49.04 (+11.52)	44.95 (+8.33)	49.51 (+10.97)
AUROC	91.42 (+1.50)	90.05 (+0.52)	91.31 (+1.33)
Točnost detekcije	84.66 (+0.63)	83.34 (-0.34)	84.74 (+0.57)
AUPR-In	92.82 (+0.97)	91.33 (-0.13)	92.43 (+0.65)
AUPR-Out	89.01 (+2.27)	87.29 (+0.99)	88.99 (+2.12)

Tablica 7 prikazuje rezultate klasifikatora učenog uz pomoć izvandistribucijskih primjera metodom definiranom u [Hendrycks et al., 2019]. Parametar β je postavljen na vrijednost 1. Parametri modela su nasumično inicijalizirani. Skup podataka TrafficSign je korišten kao negativni skup podataka. Ovaj skup podataka je manje raznovrstan od izvandistribucijskih skupova podataka u originalnom radu, ipak i ovakav skup podataka doprinosi poboljšanju klasifikatora u

detekciji izvandistribucijskih primjera. Još jednom opažamo pad klasifikacijskih sposobnosti u odnosu na bazni klasifikator (tablica 3).

Tablica 7: Performanse ResNet18 [He et al., 2016a] klasifikatora učenog izvandistribucijskim primjera metodom definiranom u [Hendrycks et al., 2019]. Korišteni izvandistribucijski skup podataka je TrafficSign. Klasifikator je transformiran u detektor izvandistribucijskih primjera koristeći maksimalnu vrijednost softmaksa (izraz 47). Vrijednosti u zagradama predstavljaju razliku u odnosu na bazni model (tablica 3). Opažamo znatno smanjenje klasifikacijskih sposobnosti klasifikatora uz osjetno povećanje performansi u detekciji izvandistribucijskih primjera.

Izvandistribucijski primjeri [Hendrycks et al., 2019]			
Točnost na skupu SVHN:88.61 (-3.30)			
OOD dataset	SVHN	LSUN	Tiny-Imagenet
TNR at TPR 95%	43.60 (+6.03)	37.09 (+0.47)	41.47 (+2.93)
AUROC	91.02 (+1.10)	89.54 (+0.01)	90.38 (+0.40)
Točnost detekcije	84.59 (+0.56)	83.09 (-0.59)	83.84 (-0.33)
AUPR-In	92.92 (+1.07)	91.64 (+0.18)	92.22 (+0.44)
AUPR-Out	87.60 (+0.86)	85.43 (-0.87)	86.75 (-0.12)

Tablica 8 prikazuje rezultate klasifikatora učenog uz pomoć izvandistribucijskih primjera metodom definiranom u [Bevandic et al., 2019]. U eksperimentu koristimo model s dvije glave. Probabilistički izlaz dodatne glave interpretiramo kao vjerojatnost da dani primjer pripada skupu podataka za učenje. Također, dodatnu glavu koristimo za detekciju izvandistribucijskih primjera. Skup podataka TrafficSign je korišten kao negativni skup podataka. Parametar β je postavljen na vrijednost 1. Tijekom učenja koristimo minigrupe koje sadrže unutar- i izvandis-

tribucijske primjere. U usporedbi s baznim klasifikatorom (tablica 3), opažamo povećanje klasifikacijskih sposobnosti za primjere iz unutardistribucijskog skupa podataka te smanjenje performansi u detekciji izvandistribucijskih primjera.

Tablica 8: Performanse ResNet18 [He et al., 2016a] klasifikatora s dodatnom glavom za detekciju izvandistribucijskih primjera [Bevandic et al., 2019]. Probabilistički izlaz dodatne glave interpretiramo kao vjerojatnost da primjer pripada unutardistribucijskom skupu podataka. Korišteni izvandistribucijski skup je TrafficSign. Opažamo povećanje klasifikacijskih sposobnosti uz smanjenje performansi u detekciji izvandistribucijskih primjera.

Izvandistribucijski primjeri [Bevandic et al., 2019]			
Točnost na skupu SVHN: 92.25 (+0.34)			
OOD dataset	SVHN	LSUN	Tiny-Imagenet
TNR at TPR 95%	33.32 (-4.20)	51.18 (+14.56)	52.93 (+14.39)
AUROC	66.75 (-23.17)	77.42 (-12.11)	78.65 (-11.30)
Točnost detekcije	64.70 (-15.33)	73.44 (-10.24)	74.45 (-9.72)
AUPR-In	62.80 (-29.05)	71.73 (-19.73)	73.42 (-18.36)
AUPR-Out	74.03 (-12.71)	83.51 (-2.79)	84.37 (-0.60)

Transformacija klasifikatora u detektor izvandistribucijskih primjera može se iskoristiti entropija probabilističkog izlaza modela i uniformne razdiobe (izraz 48). Detekcija izvandistribucijskih primjera uz pomoć entropije daje slične rezultate kao i rezultati u prethodno navedenim tablicama.

6.5.2 Rezultati na skupu podataka CIFAR10

Tablica 9 prikazuje rezultate ResNet18 [He et al., 2016a] klasifikatora učenog gubitkom unakrsne entropije na skupu podataka CIFAR10. Sposobnost detekcije izvandistribucijskih primjera klasifikatora testiramo na skupovima podataka SVHN, LSUN i Tiny-ImageNet koji služe kao izvandistribucijski skupovi podataka. Rezultati ovog modela služe kao referentna točka te ih želimo poboljšati. Klasifikator je transformiran u detektor izvandistribucijskih primjera koristeći maksimalnu vrijednost probabilističkog izlaza klasifikatora (izraz 47). Parametri modela su nasumično inicijalizirani.

Tablica 9: Performanse ResNet18 [He et al., 2016a] klasifikatora učenog s gubitkom unakrsne entropije. Klasifikator je transformiran u detektor izvandistribucijskih primjera koristeći maksimalnu vrijednost probabilističkog izlaza modela (izraz 47).

Unakrsna entropija			
Točnost na skupu CIFAR10: 71.18			
OOD dataset	SVHN	LSUN	Tiny-Imagenet
TNR at TPR 95%	10.61	11.96	10.71
AUROC	68.58	67.32	66.82
Točnost detekcije	64.56	62.99	63.13
AUPR-In	73.68	69.67	68.95
AUPR-Out	63.65	64.00	63.10

Tablica 10 prikazuje performanse klasifikatora treniranog procedurom definiranom u [Lee et al., 2018]. U fazi učenja koristimo iste hiperparametre kao u originalnom radu. Za razliku od originalnog rada [Lee et al., 2018], koristimo

ResNet18 [He et al., 2016a] klasifikator. Parametri modela su nasumično inicijalizirani. Ovako treniran klasifikator ima bolje performanse u detekciji izvandistribucijskih primjera od baznog klasifikatora. Osim toga, uočavamo porast klasifikacijske sposobnosti ovako treniranog klasifikatora u odnosu na bazni model (tablica 9).

Tablica 10: Performanse ResNet18 [He et al., 2016a] klasifikatora učenog s kompleksnim gubitkom definiranim u [Lee et al., 2018]. Klasifikator je transformiran u detektor izvandistribucijskih primjera koristeći maksimalnu vrijednost probabilističkog izlaza modela (izraz 47). Opažamo povećanje klasifikacijskih sposobnosti uz povećanje performansi u detekciji izvandistribucijskih primjera (tablica 9).

Kompleksni gubitak + GAN [Lee et al., 2018]			
Točnost na skupu CIFAR10: 71.53 (+0.35)			
OOD dataset	SVHN	LSUN	Tiny-Imagenet
TNR at TPR 95%	17.70 (+7.09)	14.10 (+2.14)	12.86 (+2.15)
AUROC	78.20 (+9.62)	71.92 (+4.60)	69.61 (+2.79)
Točnost detekcije	71.75 (+7.19)	66.54 (+3.55)	64.74 (+1.61)
AUPR-In	81.92 (+8.24)	74.00 (+4.33)	75.85 (+6.90)
AUPR-Out	72.92 (+9.27)	67.79 (+3.79)	65.73 (+2.63)

Tablica 11 prikazuje performanse klasifikatora treniranog metodom definiranom algoritmom 1. Arhitektura modela Real NVP korištenog u ovom eksperimentu ima 3 rezidualna bloka s 32 mape značajki u prvom sloju. Poduzorkovanje se radi tri puta. Parametar β je postavljen na 3. Veličina minigrupe je 128. Klasifikator je transformiran u detektor izvandistribucijskih primjera koristeći maksimalnu vrijednost probabilističkog izlaza modela (izraz 47). Naučeni klasifikator

ima bolje performanse u detekciji izvandistribucijskih primjera od baznog modela. Također, opažamo da ovako treniran klasifikator ima bolje klasifikacijske performanse od baznog modela (tablica 9). U ovom eksperimentu kao i u svim ostalim eksperimentima slike pretprocesiramo tako da primjenimo vodoravni zaokret s vjerojatnošću 0.5.

Tablica 11: Performanse ResNet18[He et al., 2016a] klasifikatora učenog našom metodom (Algoritam 1). Klasifikator je transformiran u detektor izvandistribucijskih primjera koristeći maksimalnu vrijednost probabilističkog izlaza modela (izraz 47). Opažamo povećanje klasifikacijskih sposobnosti uz povećanje performansi u detekciji izvandistribucijskih primjera (tablica 9).

Kompleksni gubitak + Real NVP			
Točnost na skupu CIFAR10: 75.74 (+4.56)			
OOD dataset	SVHN	LSUN	Tiny-Imagenet
TNR at TPR 95%	18.28 (+7.67)	22.90 (+10.94)	14.23 (+3.52)
AUROC	82.50 (+13.92)	78.54 (+11.22)	70.23 (+3.41)
Točnost detekcije	77.33 (+12.77)	71.71 (+8.72)	65.20 (+2.07)
AUPR-In	86.70 (+13.02)	80.62 (+10.95)	71.61 (+2.66)
AUPR-Out	75.51 (+11.86)	75.18 (+11.18)	66.89 (+3.79)

Tablica 12 prikazuje rezultate klasifikatora učenog sa sintetičkim primjerima s ruba distribucije podataka. Sintetički primjeri s ruba distribucije su dobiveni koristeći gubitak definiran izrazom (49), gdje je skup podataka Tiny-ImageNet korišten kao izvandistribucijski skup podataka. Arhitektura modela Real NVP korištenog u ovom eksperimentu ima 2 rezidualna bloka s 32 mape značajki u prvom sloju. Poduzorkovanje se radi tri puta. Parametar λ je postavljen na

vrijednost 15. Pri učenju klasifikatora parametar β postavljamo na vrijednost 2. Parametri modela su nasumično inicijalizirani. Veličina minigrupe u fazi učenja je 128. Ova metoda postiže bolje rezultate u detekciji izvandistribucijskih primjera od baznog klasifikatora uz smanjenje klasifikacijske sposobnosti (tablica 9).

Tablica 12: Performanse ResNet18 [He et al., 2016a] klasifikatora učenog optimiranjem gubitka definiranog izrazom (46). Klasifikator je transformiran u detektor izvandistribucijskih primjera koristeći maksimalnu vrijednost probabilističkog izlaza modela (izraz 47). Opažamo povećanje klasifikacijskih sposobnosti uz povećanje performansi u detekciji izvandistribucijskih primjera (tablica 9).

Primjeri s ruba distribucije (49)			
Točnost na skupu CIFAR10: 71.29 (+0.11)			
OOD dataset	SVHN	LSUN	Tiny-Imagenet
TNR at TPR 95%	17.12 (+6.51)	24.83 (+12.87)	19.08 (+8.37)
AUROC	78.73 (+10.15)	79.16 (+11.84)	73.81 (+6.99)
Točnost detekcije	72.82 (+8.26)	71.59 (+8.60)	67.43 (+4.30)
AUPR-In	81.81 (+8.13)	81.30 (+11.63)	74.90 (+5.95)
AUPR-Out	72.79 (+9.14)	75.96 (+11.96)	70.85 (+7.75)

Tablica 13 prikazuje rezultate klasifikatora učenog uz pomoć izvandistribucijskih primjera metodom definiranom u [Hendrycks et al., 2019]. Parametar β je postavljen na vrijednost 1. Parametri modela su nasumično inicijalizirani. Skup podataka Tiny-ImageNet je korišten kao negativni skup podataka. Ovaj skup podataka je manje raznovrstan od izvandistribucijskih skupova podataka u originalnom radu, ipak i ovakav skup podataka doprinosi poboljšanju klasifikatora u detekciji izvandistribucijskih primjera. Još jednom opažamo pad klasifikacijskih

sposobnosti u odnosu na bazni klasifikator (tablica 9).

Tablica 13: Performanse ResNet18 [He et al., 2016a] klasifikatora učenog izvandistribucijskim primjera metodom definiranom u [Hendrycks et al., 2019]. Korišteni izvandistribucijski skup podataka je TrafficSign. Klasifikator je transformiran u detektor izvandistribucijskih primjera koristeći maksimalnu vrijednost softmaksa (izraz 47). Vrijednosti u zagradama predstavljaju razliku u odnosu na bazni model (tablica 9). Opažamo znatno povećanje klasifikacijskih sposobnosti klasifikatora uz osjetno povećanje performansi u detekciji izvandistribucijskih primjera.

Izvandistribucijski primjeri [Hendrycks et al., 2019]			
Točnost na skupu CIFAR10: 76.79 (+5.61)			
OOD dataset	SVHN	LSUN	Tiny-Imagenet
TNR at TPR 95%	12.15 (+1.54)	14.46 (+2.50)	13.39 (+2.68)
AUROC	72.45 (+3.87)	71.05 (+3.73)	69.84 (+3.02)
Točnost detekcije	68.01 (+3.45)	66.35 (+3.36)	64.85 (+1.72)
AUPR-In	77.05 (+3.37)	71.85 (+2.18)	71.58 (+2.63)
AUPR-Out	66.64 (+2.99)	67.38 (+3.38)	65.98 (+2.88)

Tablica 14 prikazuje rezultate klasifikatora učenog uz pomoć izvandistribucijskih primjera metodom definiranom u [Bevandic et al., 2019]. U eksperimentu koristimo model s dvije glave. Probabilistički izlaz dodatne glave interpretiramo kao vjerojatnost da dani primjer pripada skupu podataka za učenje. Također, dodatnu glavu koristimo za detekciju izvandistribucijskih primjera. Skup podataka Tiny-ImageNet je korišten kao negativni skup podataka. Parametar β je postavljen na vrijednost 1. Tijekom učenja koristimo minigrupe koje sadrže

unutar- i izvandistribucijske primjere. U usporedbi s baznim klasifikatorom (tablica 9), opažamo značajno povećanje klasifikacijskih sposobnosti za primjere iz unutardistribucijskog skupa podataka te povećanje performansi u detekciji izvandistribucijskih primjera.

Tablica 14: Performanse ResNet18 [He et al., 2016a] klasifikatora s dodatnom glavom za detekciju izvandistribucijskih primjera [Bevandic et al., 2019]. Probabilistički izlaz dodatne glave interpretiramo kao vjerojatnost da primjer pripada unutardistribucijskom skupu podataka. Korišteni izvandistribucijski skup je Tiny-ImageNet. Opažamo povećanje klasifikacijskih sposobnosti uz povećanje performansi u detekciji izvandistribucijskih primjera.

Izvandistribucijski primjeri [Bevandic et al., 2019]			
Točnost na skupu CIFAR10: 76.79 (+5.61)			
OOD dataset	SVHN	LSUN	Tiny-Imagenet
TNR at TPR 95%	20.53 (+9.92)	99.90 (+87.94)	100.00 (+89.92)
AUROC	55.68 (-12.90)	99.96 (+32.64)	100.00 (+33.18)
Točnost detekcije	58.72 (+5.84)	99.93 (+36.94)	100.00 (+36.87)
AUPR-In	56.75 (+16.93)	99.96 (+30.29)	100.00 (31.05)
AUPR-Out	62.57 (-1.08)	99.98 (+35.98)	100.00 (+36.9)

Još jednom smo pokušali transformacirati klasifikator u detektor izvandistribucijskih primjera uz pomoć entropije probabilističkog izlaza modela i uniformne razdiobe (izraz 48). Dobiveni rezultati su slični kao i rezultati u prethodno navedenim tablicama.

6.6 Generiranje sintetičkih primjera s ruba distribucije

U ovom potpoglavlju koristimo metodu definiranu algoritmom 1 za generiranje primjera s ruba distribucije. Uzorkovanjem generativnog modela dobijemo sintetičke primjere s ruba distribucije.

6.6.1 Rezultati na skupu podataka CIFAR10

Odabrani klasifikator jest ResNet18 [He et al., 2016a]. Arhitektura modela Real NVP se sastoji od 3 rezidualna bloka s 32 mape značajki u prvom sloju. Poduzorkovanje se izvršava tri puta. Parametar β je postavljeni na 3. Veličina minigrupe je 128.

Slika 28 prikazuje primjere generiranih primjera s ruba distribucije na skupu podataka CIFAR10.

Slika 28: Primjeri s ruba distribucije za skup podataka CIFAR10.

6.6.2 Rezultati na skupu podataka SVHN

Odabrani klasifikator jest ResNet18 [He et al., 2016a]. Arhitektura modela Real NVP se sastoji od 2 rezidualna bloka s 32 mape značajki u prvom sloju.

Poduzorkovanje se izvršava tri puta. Parametar β je postavljeni na 5. Veličina minigrupe je 128.

Slika 29 prikazuje primjere generiranih primjera s ruba distribucije na skupu podataka SVHN.

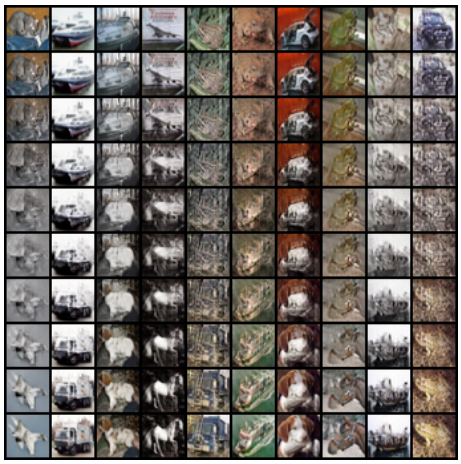
Slika 29: Primjeri s ruba distribucije za skup podataka SVHN.

6.7 Interpolacija između primjera skupa podataka

U ovom eksperimentu prikazujemo interpolaciju između parova slika iz skupa podataka za učenje. Slike 31 i 30 prikazuju primjere takve interpolacije. U slikama, prvi i zadnji redak prikazuju primjere iz skupa podataka, dok retci između pokazuju primjere generirane modelom Real NVP. U ovom slučaju radi se o unutardistribucijskim primjerima iz skupa za validaciju. Primjer u i-tom elementu svakog stupca je generiran koristeći izraz:

$$\mathbf{x}_i = \mathbf{f}^{-1}(a\mathbf{f}(\mathbf{x}_1) + (1-a)\mathbf{f}(\mathbf{x}_2)) \quad a \in [0,1] \tag{59}$$

Slika 30: Interpolacija između primjera iz skupa podataka SVHN.



Slika 31: Interpolacija između primjera iz skupa podataka CIFAR10.

7 Zaključak

U okviru ovog rada smo obradili pristupe detekciji izvandistribucijskih primjera. Nakon definiranja osnovnih pojmova i pregleda literature, objašnjavamo generativni model Real NVP koji je primjer modela temeljenog na autoregresijskim normalizirajućim tokovima. Osim detaljnog definiranja arhitekture temeljene na sloju miješanja, povlačimo paralele između normalizirajućeg toka i ostalih generativnih modela. Zbog svoje pogodne arhitekture, ovaj model ima sposobnost generiranja slika veće rezolucije od slika za učenje, što kod drugih generativnih modela nije slučaj.

Slično kao [Nalisnick et al., 2019], u radu opisujemo neuspješan pokušaj detektiranja izvandistribucijskih primjera na temelju logaritma izglednosti te razloge zbog kojih pristup ne radi. Za razliku od [Nalisnick et al., 2019], u našim eksperimentima koristimo generativni model Real NVP. Rezultati provedenih eksperimentata se slažu s rezultatima dobivenima u [Nalisnick et al., 2019].

Također, unapređujemo metodu opisanu u radu [Lee et al., 2018] korištenjem generativnog modela Real NVP. Kao i procedura definirana u [Lee et al., 2018], unaprijeđena metoda generira sintetičke primjere s ruba distribucije, a klasifikator učen metodom koja koristi Real NVP postiže bolje rezultate u detekciji izvandistribucijskih primjera od baznog klasifikatora kao i od klasifikatora učenog metodom [Lee et al., 2018]. Konačno, uspoređujemo performanse konkurentnih metoda u detekciji izvandistribucijskih primjera definiranima u [Bevandic et al., 2019], [Lee et al., 2018] te [Hendrycks et al., 2019].

Sva opažanja i tvrdnje definirane u ovom radu smo potkrijepili vlastitim eksperimentima koristeći skupove podataka CIFAR10 [Krizhevsky, 2012] i SVHN [Netzer et al., 2011]. Rezultati eksperimenata se slažu s opažanjima i definiranim tvrdnjama.

U budućem radu predlažemo primjenu razvijene metode na odgovarajuće pro-

bleme iz domene računalnog vida. Npr. klasifikator treniran našom metodom ima povećanu sposobnost detektiranja izvandistribucijskih primjera, što može biti korak ka rješenju sigurnosnih problema dubokih modela. Umjetno generirani primjeri s ruba distribucije se mogu varirati u rezoluciji, što otvara mogućnosti za primjenu u gustoj predikciji slika uz istovremenu detekciju izvandistribucijskih dijelova slike [Bevandic et al., 2018, Bevandic et al., 2019] i mnoge druge.

Literatura

[Bevandic et al., 2018] Bevandic, P., Kreso, I., Orsic, M., and Segvic, S. (2018). Discriminative out-of-distribution detection for semantic segmentation. *CoRR*, abs/1808.07703.

[Bevandic et al., 2019] Bevandic, P., Kreso, I., Orsic, M., and Segvic, S. (2019). Simultaneous semantic segmentation and outlier detection in presence of domain shift. In Fink, G. A., Frintrop, S., and Jiang, X., editors, *Pattern Recognition - 41st DAGM German Conference, DAGM GCPR 2019, Dortmund, Germany, September 10-13, 2019, Proceedings*, volume 11824 of *Lecture Notes in Computer Science*, pages 33–47. Springer.

[Carreira-Perpiñán and Hinton, 2005] Carreira-Perpiñán, M. Á. and Hinton, G. E. (2005). On contrastive divergence learning. In Cowell, R. G. and Ghahramani, Z., editors, *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics, AISTATS 2005, Bridgetown, Barbados, January 6-8, 2005*. Society for Artificial Intelligence and Statistics.

[Dauphin et al., 2015] Dauphin, Y. N., de Vries, H., Chung, J., and Bengio, Y. (2015). Rmsprop and equilibrated adaptive learning rates for non-convex optimization. *CoRR*, abs/1502.04390.

[Deng et al., 2009] Deng, J., Dong, W., Socher, R., Li, L., Li, K., and Li, F. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*

(CVPR 2009), 20-25 June 2009, Miami, Florida, USA, pages 248–255. IEEE Computer Society.

[Dinh et al., 2015] Dinh, L., Krueger, D., and Bengio, Y. (2015). NICE: nonlinear independent components estimation. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings*.

[Dinh et al., 2017] Dinh, L., Sohl-Dickstein, J., and Bengio, S. (2017). Density estimation using real NVP. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

[Duchi et al., 2011] Duchi, J. C., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, 12:2121–2159.

[Goodfellow et al., 2014] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. C., and Bengio, Y. (2014). Generative adversarial networks. *CoRR*, abs/1406.2661.

[Grathwohl et al., 2019] Grathwohl, W., Wang, K., Jacobsen, J., Duvenaud, D., Norouzi, M., and Swersky, K. (2019). Your classifier is secretly an energy based model and you should treat it like one. *CoRR*, abs/1912.03263.

[Guo et al., 2017] Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. (2017). On calibration of modern neural networks. In Precup, D. and Teh, Y. W., editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR.

[He et al., 2016a] He, K., Zhang, X., Ren, S., and Sun, J. (2016a). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society.

[He et al., 2016b] He, K., Zhang, X., Ren, S., and Sun, J. (2016b). Identity mappings in deep residual networks. In Leibe, B., Matas, J., Sebe, N., and Welling, M., editors, *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV*, volume 9908 of *Lecture Notes in Computer Science*, pages 630–645. Springer.

[Hendrycks and Gimpel, 2017] Hendrycks, D. and Gimpel, K. (2017). A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

[Hendrycks et al., 2019] Hendrycks, D., Mazeika, M., and Dietterich, T. G. (2019). Deep anomaly detection with outlier exposure. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

[Huang et al., 2017] Huang, G., Liu, Z., van der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 2261–2269. IEEE Computer Society.

[Ioffe and Szegedy, 2015] Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Bach, F. R. and Blei, D. M., editors, *Proceedings of the 32nd International*

Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 448–456. JMLR.org.

[Kingma and Ba, 2015] Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

[Kingma and Dhariwal, 2018] Kingma, D. P. and Dhariwal, P. (2018). Glow: Generative flow with invertible 1x1 convolutions. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, pages 10236–10245.

[Kingma et al., 2016] Kingma, D. P., Salimans, T., and Welling, M. (2016). Improving variational inference with inverse autoregressive flow. *CoRR*, abs/1606.04934.

[Kingma and Welling, 2014] Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. In Bengio, Y. and LeCun, Y., editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

[Krizhevsky, 2012] Krizhevsky, A. (2012). Learning multiple layers of features from tiny images. *University of Toronto*.

[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Bartlett, P. L., Pereira, F. C. N., Burges, C. J. C., Bottou, L., and Weinberger, K. Q.,

editors, *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, pages 1106–1114.

[Lakshminarayanan et al., 2017] Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 6402–6413.

[Lee et al., 2018] Lee, K., Lee, H., Lee, K., and Shin, J. (2018). Training confidence-calibrated classifiers for detecting out-of-distribution samples. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

[Liang et al., 2018] Liang, S., Li, Y., and Srikant, R. (2018). Enhancing the reliability of out-of-distribution image detection in neural networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

[Lucas et al., 2019] Lucas, T., Shmelkov, K., Alahari, K., Schmid, C., and Verbeek, J. (2019). Adaptive density estimation for generative models. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32:*

Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada, pages 11993–12003.

[Nalisnick et al., 2019] Nalisnick, E. T., Matsukawa, A., Teh, Y. W., Görür, D., and Lakshminarayanan, B. (2019). Do deep generative models know what they don't know? In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

[Netzer et al., 2011] Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. (2011). Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*.

[Papamakarios et al., 2017] Papamakarios, G., Murray, I., and Pavlakou, T. (2017). Masked autoregressive flow for density estimation. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 2338–2347.

[Radford et al., 2016] Radford, A., Metz, L., and Chintala, S. (2016). Unsupervised representation learning with deep convolutional generative adversarial networks. In Bengio, Y. and LeCun, Y., editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

[Ren et al., 2019] Ren, J., Liu, P. J., Fertig, E., Snoek, J., Poplin, R., DePristo, M. A., Dillon, J. V., and Lakshminarayanan, B. (2019). Likelihood ratios for out-of-distribution detection. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R., editors, *Advances in Neural*

Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada, pages 14680–14691.

[Salakhutdinov et al., 2007] Salakhutdinov, R., Mnih, A., and Hinton, G. E. (2007). Restricted boltzmann machines for collaborative filtering. In Ghahramani, Z., editor, *Machine Learning, Proceedings of the Twenty-Fourth International Conference (ICML 2007), Corvallis, Oregon, USA, June 20-24, 2007*, volume 227 of *ACM International Conference Proceeding Series*, pages 791–798. ACM.

[Salimans et al., 2017] Salimans, T., Karpathy, A., Chen, X., and Kingma, D. P. (2017). Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

[Serrà et al., 2019] Serrà, J., Álvarez, D., Gómez, V., Slizovskaia, O., Núñez, J. F., and Luque, J. (2019). Input complexity and out-of-distribution detection with likelihood-based generative models. *CoRR*, abs/1909.11480.

[van den Oord et al., 2016] van den Oord, A., Kalchbrenner, N., Espeholt, L., Kavukcuoglu, K., Vinyals, O., and Graves, A. (2016). Conditional image generation with pixelcnn decoders. In Lee, D. D., Sugiyama, M., von Luxburg, U., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 4790–4798.

[Yu et al., 2015] Yu, F., Zhang, Y., Song, S., Seff, A., and Xiao, J. (2015). LSUN: construction of a large-scale image dataset using deep learning with humans in the loop. *CoRR*, abs/1506.03365.

Sažetak

Detekcija izvandistribucijskih primjera primjenom generativnog modela Real NVP

Matej Grcić

Ovaj rad proučava detekciju izvandistribucijskih primjera diskriminativnim modelom koji istovremeno obavlja i klasifikaciju podataka. Glavni doprinos rada je poboljšanje pristupa utemeljenog na generiranju podataka na rubu distribucije skupa za učenje. Naše poboljšanje temelji se na generativnom modelu Real NVP koji je baziran na normalizirajućem toku, a ima nekoliko važnih prednosti u odnosu na suparničke generativne modele koji su korišteni ranije. Prednosti naše formulacije su jednostavnije učenje, bolje modeliranje distribucije podataka za učenje te bolji rezultati u empirijskim eksperimentima. Naš pristup postiže točniju detekciju izvandistribucijskih primjera od osnovnog pristupa koji ne koristi generativno modeliranje, od pristupa utemeljenog na izračunavanju izglednosti podataka te od pristupa koji se temelji na suparničkom generiranju rubnih podataka. Svi doprinosi vrednovani su iscrpnim eksperimentima.

Ključne riječi: detekcija izvandistribucijskih primjera, Real NVP, primjeri s ruba distribucije

Summary

Out-of-distribution detection by using generative model Real NVP

Matej Grcić

This work examines the problem of out-of-distribution detection with a deep discriminative model capable of classification. The main contribution of this work is the enhancement of the method based on generative sampling at the data distribution border. Our enhancement of the method rests on flow-based generative model Real NVP. The main advantages of our enhancement are smoother training phase, better data distribution coverage and superior empirical results. Our approach has superior results in outlier detection over both the baseline model and the model trained with samples at the distribution border generated by adversarial networks. All contributions are evaluated on exhaustive experiments.

Keywords: out-of-distribution detection, Real NVP, samples at the distribution border