SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 2004

Detekcija izvan-distribucijskih primjeraka suparničkim učenjem

Hrvoje Bušić

Zagreb, lipanj 2019.

Umjesto ove stranice umetnite izvornik Vašeg rada.
Da bi ste uklonili ovu stranicu obrišite naredbu `\izvornik`.

SADRŽAJ

1. Uvod

Klasifikacija slika je važan zadatak računalnog vida s mnogim zanimljivim primjenama. U posljednje vrijeme najbolji rezultati u tom području postižu se diskriminativnim konvolucijskim modelima (Girshick, 2015). Međutim, diskriminativni modeli su skloni neopravdanom optimizmu, što znači da primjerci izvan domene ekspertize modela često bivaju neispravno klasificirani s velikom procjenom pouzdanosti (Hendrycks i Gimpel, 2016; Goodfellow et al., 2014b; Amodei et al., 2016). Ovaj rad proučava mogućnost rješavanja problema detekcije izvan-distribucijskih primjeraka od strane diskriminativnih modela primjenom suparničkog učenja.

U istraživanju se oslanjamo na rad Hendrycks i Gimpel (2016) koji kvantificiraju pouzdanost modela u predikciju preko maksimalne aktivacije funkcije softmax. Problemu detekcije izvan-distribucijskih primjeraka se pristupa kroz izgradnju detektora zasnovanih na pragu povrh naučenih klasifikatora koji u obzir uzimaju pouzdanost modela u predikciju. Lee et al. (2017) je rad u središtu našeg istraživanja koji opisuje novi pristup učenju robusnih klasifikatora kroz učenje različitih prediktivnih distribucija za unutar- i izvan-distribucijske primjere. Autori proširuju funkciju cijene unakrsne entropije klasifikatora komponentom koja prediktivnu distribuciju izvan-distribucijskih primjeraka približava uniformnoj, efektivno čineći klasifikator manje pouzdanim u vlastita predviđanja nad tim primjerima, dok se generativne suparničke mreže koriste za uzorkovanje najkorisnijih izvan-distribucijskih primjeraka. Različite prediktivne distribucije, gdje je prediktivna-distribucija izvan-distribucijskih primjeraka bliska uniformnoj, otvaraju prostor izgradnji kvalitetnih detektora zasnovanih na pragu prema Hendrycks i Gimpel (2016). Pristup je praktično predstavljen kroz združeni postupak učenja središnjeg klasifikatora i generativnog suparničkog modela s prilagođenim funkcijama cilja, gdje se komponente u hodu međusobno potpomažu.

U prvom poglavlju predstavljamo osnovne gradivne dijelove dubokih konvolucijskih mreža korištenih u ovom radu i način njihovog učenja. Drugo poglavlje opisuje koncept generativnih suparničkih mreža i argumentira korištenje njihovih dubokih konvolucijskih inačica. U trećem poglavlju formalno uvodimo problem detekcije izvan-

distribucijskih primjeraka oslanjajući se na Hendrycks i Gimpel (2016), s pregledom doprinosa rješavanju problema kroz proširenje funkcije cilja klasifikatora i suparničkog modela pod novim združenim režimom učenja od strane Lee et al. (2017). Konačno, predstavljamo rezultate iscrpnih eksperimenata u kojima smo ispitali uspješnost združenog pristupa učenju pod izmijenjenim ciljem, gdje performanse detekcije izvan-distribucijskih primjeraka mjerimo metrikama koje slijede osnovicu Hendrycks i Gimpel (2016) preko više različitih izvan-distribucijskih skupova različite težine.

2. Duboke neuronske mreže

Duboko učenje je grana strojnog učenja koja je zabilježila značajne uspjehe u primjeni na problemima iz raznih područja. Diskriminativni modeli postižu izvrsne rezultate na različitim klasifikacijskim zadatcima poput prepoznavanja govora (Hannun et al., 2014), klasifikaciji slika (Girshick, 2015) i procjeni određenih liječničkih dijagnoza (Caruana et al., 2015). Napredak u modeliranju i treniranju njihovih dubokih inačica s različitim vrstama funkcionalnih slojeva, dostupnost velikih podatkovnih skupova različite težine preko više domena, mogućnost iskorištavanja grafičkih kartica za paralelizaciju izračuna i ubrzanje procesa učenja su samo neki od razloga koji stoje iza ovog uspjeha.

Na visokom nivou, neuronske mreže su složeni sustavi koji se sastoje od mnoštva međusobno povezanih osnovnih jedinica - neurona. Formalno, izlaz jednog neurona funkcija je vektora parametara neurona $\vec{w}$, pomaka b i vektora aktivacija prethodnog sloja neuronske mreže $\vec{x}$ s kojima je neuron direktno povezan:

$$y = f(\vec{x}^\top \cdot \vec{w} + b) = f(\sum_{i=1}^{n} x_i \cdot w_i + b) \tag{2.1}$$

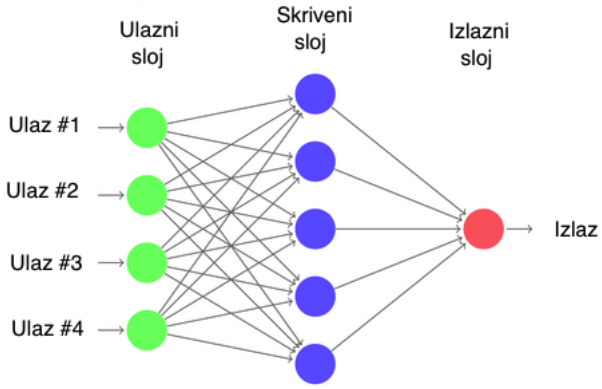
gdje f označava aktivacijsku funkciju neurona, čija je uloga uvođenje nelinearnosti u transformaciju. $\vec{w}$ i b su parametri neurona koji se uče kroz proces treniranja.

Neuronske mreže su organizirane u slojevima koji okupljaju grupe neurona sa zajedničkom zadaćom. Slojevi su međusobno povezani u lančanu strukturu, gdje izlazi jednog sloja imaju ulogu aktivacija sloju koji slijedi. Formalno, sloj jednostavne neuronske mreže tada možemo opisati funkcijom:

$$\vec{h}^{(i)} = f^{(i)}(\vec{h}^{(i-1)\top}\mathbf{W}^{(i)} + \vec{b}^{(i)}) \tag{2.2}$$

gdje $\vec{h}^{(i)}$ predstavlja izlaz i-tog sloja mreže, $f^{(i)}$ je aktivacijska funkcija, a $\mathbf{W}^{(i)}$ i $\vec{b}^{(i)}$ parametri sloja koji se uče.

Neuronske mreže koje sadrže dva ili više skrivena sloja - sloja koji slijede ulaz neuronske mreže (2.1) - nazivamo dubokim neuronskim mrežama, dok neuronske mreže



Slika 2.1: Jednostavna neuronska mreža. Preuzeto iz Zelić (2018).

koje u svom složaju imaju konvolucijske slojeve (opisane u nastavku) nazivamo konvolucijskim neuronskim mrežama.

U ovom poglavlju na visokom nivou predstavljamo slojeve dubokih konvolucijskih neuronskih mreža i algoritme učenja koji su korišteni u ovom radu.

2.1. Slojevi dubokog konvolucijskog modela

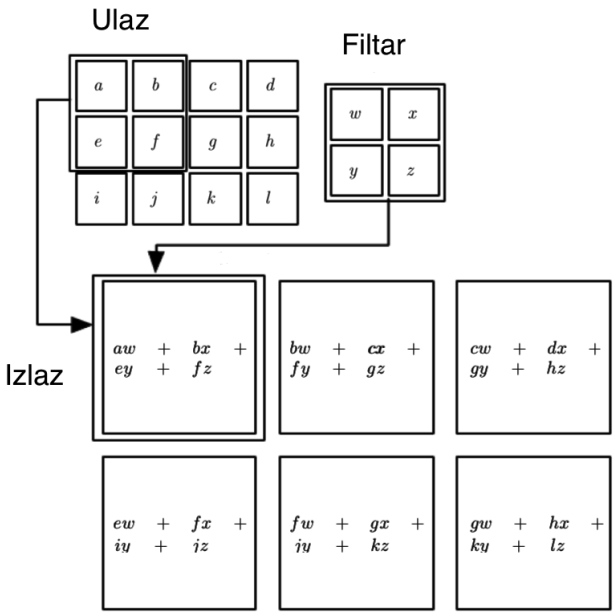
Potpuno povezani sloj

Potpuno povezani sloj neuronske mreže vrši afinu transformaciju ulaza. Ulazne aktivacije sloja se množe s matricom težina te im se dodaje pomak. Matrica težina je parametar potpuno povezanog sloja koji se uči, dok je korištenje pomaka opcionalno te on može biti fiksan ili učen tijekom treniranja. Rezultati afine transformacije se najčešće propuštaju kroz aktivacijsku funkciju prije prolaza prema sljedećem sloju neuronske mreže (2.2).

Glavna karakteristika potpuno povezanog sloja jest povezanost svih izlaznih aktivacija sa svim ulaznim aktivacijama, što omogućuje učenje značajki visokog reda. Iz ovog razloga u dubokim konvolucijskim diskriminativnim modelima potpuno povezani slojevi često slijede konvolucijske slojeve mreže, kako bi se uz pomoć značajki visokog reda pruženih od strane konvolucijskih slojeva opisala kvalitetna klasifikacijska funkcija.

Konvolucijski sloj

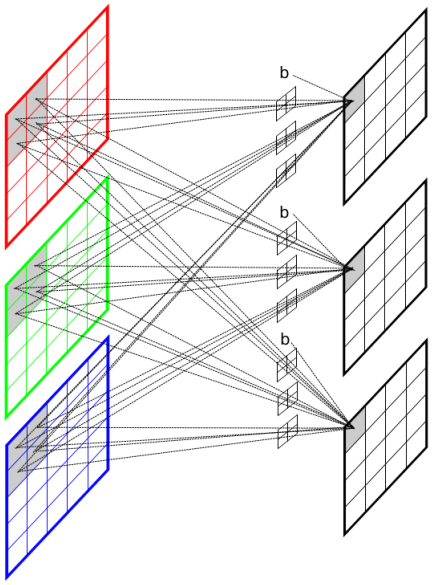
Konvolucijski sloj je temeljni građevni blok duboke konvolucijske neuronske mreže. Konvolucijski sloj je parametriziran skupom filtara (konvolucijskih jezgri) koji se uče



Slika 2.2: Primjer računanja aktivacija izlazne mape značajki konvolucijskog sloja preuzet iz Goodfellow et al. (2016)

tijekom procesa treniranja neuronske mreže. Filtri posjeduju malo receptivno polje, ali se protežu punom dubinom ulaznog volumena te prilikom unaprijednog prolaza prelaze preko pune širine i visine korespondentne ulazne mape značajki. Segmenti ulaznih mapi značajki se množe s korespondentnim konvolucijskim jezgrama te se dobiveni produkti zbrajaju, formirajući izlaznu aktivaciju (2.2). Posljedica postepenog prelaska jezgri preko ulaznih mapi jest preklapanje receptivnih polja bliskih izlaznih aktivacija. Konačan rezultat prijelaza jezgri preko ulaznih mapi značajki je aktivacijska mapa, odnosno izlazna mapa značajki. Prolaskom više različitih skupova filtara preko svih ulaznih aktivacijskih mapa dobivamo različite izlazne aktivacijske mape (2.3). Tijekom treniranja mreža uči filtre koji se aktiviraju pri detekciji određene značajke na određenoj prostornoj poziciji ulaza. Svaka vrijednost izlaznog volumena se stoga može interpretirati i kao izlaz neurona koji gleda na malu regiju u ulazu i dijeli parametre s neuronima u istoj izlaznoj mapi značajki (2.3).

Dimenzije izlazne mape značajki su određene veličinom konvolucijske jezgre $DJ = k * k$ (engl. *kernel size*), korakom konvolucije K (engl. *stride*), ispunom do rubova I (engl. *padding*) i dimenzijama ulaza h i w, gdje je h visina i w širina. Dimenzije konvolucijskih jezgri se pretežito postavljaju na neparnu vrijednost k, korak konvolucije određuje iznos pomaka konvolucijske jezgre po dimenziji ulaza, dok ispuna omogućuje manipuliranje dimenzijama izlazne mape značajki. Izračun dimenzija izlazne



Slika 2.3: Vizualizacija dijeljenja parametara pri unaprijednom prolazu preko ukupnog volumena ulaza od strane aktivacija u istoj izlaznoj mapi značajki. Preuzeto iz Vukotić (2014).

mape značajki je definiran formulama:

$$\hat{h} = \left\lfloor \frac{h + 2I - k}{K} + 1 \right\rfloor \qquad \hat{w} = \left\lfloor \frac{w + 2I - k}{K} + 1 \right\rfloor \tag{2.3}$$

gdje je $\hat{h}$ visina i $\hat{w}$ širina izlazne mape značajki.

Prednost konvolucijskog sloja u usporedbi s potpuno povezanim slojem jest signifikantno manji broj parametara koji se koriste te koje treba naučiti, što je posljedica korištenja konvolucijskih jezgara s malim receptivnim poljem. Za ulaznu sliku koja može imati milijun i više piksela, relevantne regije poput rubova mogu zauzimati više desetaka ili stotina piksela. Tijekom unaprijednog prolaza kroz slojeve dubokog konvolucijskog modela, konvolucijski filtri izlučuju bitne značajke obrađujući regije originalne slike. Manji broj parametara potreban za ovakvu ekstrakciju rezultira manjim memorijskim zauzećem i većom statističkom učinkovitošću konvolucijskog sloja.

Formalno, neka je ulazni volumen konvolucijskog sloja dimenzija $c * h * w$, gdje c označava broj ulaznih mapi značajki, a h i w visinu i širinu svake od mapi značajki. Neka konvolucijski sloj sadrži jezgre veličine k koje računaju o izlaznih mapi značajki. Ukupan broj parametara konvolucijskog sloja iznosi $c \cdot k \cdot k \cdot f + f$ ukoliko se koristi pomak pri izračunu aktivacija, odnosno $c \cdot k \cdot k \cdot f$ ako se pomak ne koristi. Primjećujemo kako broj parametara ne ovisi o veličini ulazne mape značajki.

Sloj unatražne konvolucije

Sloj unatražne konvolucije (engl. *transposed convolution*) obavlja transformaciju suprotnog smjera od one konvolucijskog sloja, što omogućuje učenje prostornog povećanja mapi značajki. Unatražna konvolucija jednu od primjena posjeduje u dubokim generativnim suparničkim modelima za parametrizirano povećanje rezolucije mapi značajki pri generiranju primjera iz vektora šuma (Goodfellow et al., 2014a; Radford et al., 2015).

Dimenzije izlazne mape značajki su određene veličinom konvolucijske jezgre $DJ = k * k$, korakom konvolucije K, ispunom do rubova ulaza I_{in}, ispunom do rubova izlaza I_{out} i dimenzijama ulaza h i w. Izračun dimenzija izlazne mape značajki je definiran formulom:

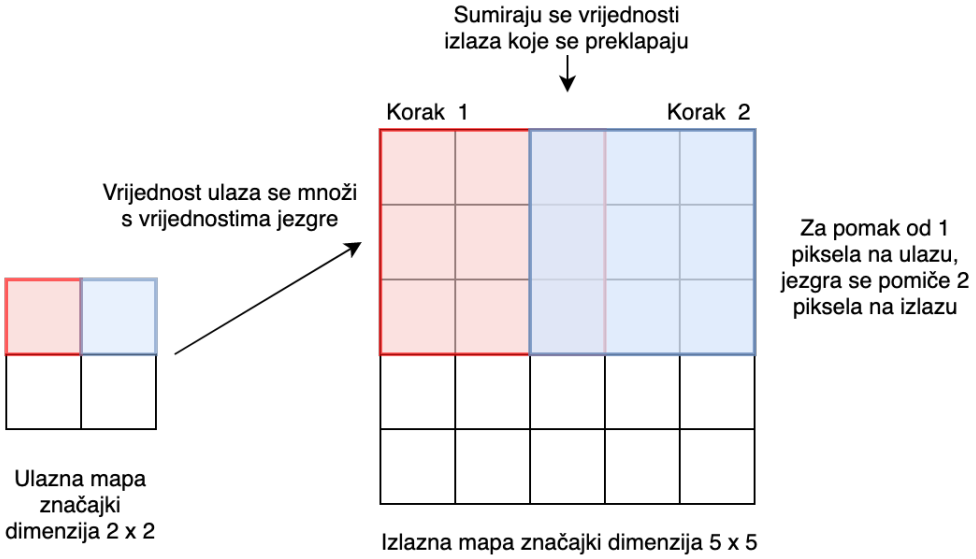
$$\hat{h} = (h-1) * K - 2I_{in} + k + I_{out} \qquad \hat{w} = (w-1) * K - 2I_{in} + k + I_{out} \quad (2.4)$$

gdje je k dimenzija konvolucijske jezgre, $\hat{h}$ visina i $\hat{w}$ širina izlazne mape značajki.

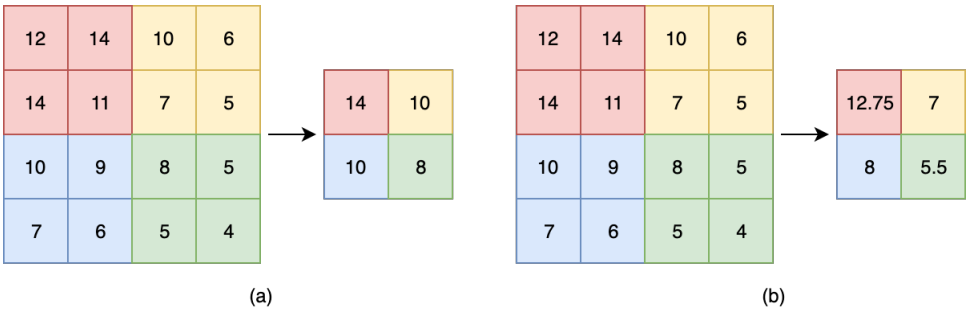
Slika 2.4 prikazuje 2 koraka unatražne konvolucije za ulaznu mapu značajki dimenzija 2x2, s dimenzijom konvolucijske jezgre $k = 3$, korakom konvolucije $K = 2$, bez nadopune. Vrijednost ulazne mape značajki u crvenoj boji se množi s parametrima konvolucijske jezgre te se dobivene vrijednosti zapisuju na mjesta unutar crvenog kvadrata izlazne mape značajki. Zbog koraka konvolucijske jezgre $K = 2$, pomak od 1 piksela na ulazu rezultira pomakom konvolucijske jezgre od 2 piksela na izlazu. Postupak se ponavlja za 2. korak unatražne konvolucije označen plavom bojom. Na mjestima gdje se izlazi preklapaju dolazi do sumiranja vrijednosti. Postupak se vrši za sve vrijednosti ulazne mape značajki, što za prikazani primjer rezultira izlaznom mapom značajki dimenzija 5x5. Postupak unatražne konvolucije idejno odgovara unatražnom prolazu obične konvolucije.

Sloj sažimanja

Slojevima sažimanja neuronske mreže ostvaruje se poduzorkovanje ulazne mape značajki. Poduzorkovanjem se ostvaruje invarijantnost na lokalne translacije, što je korisno kada želimo detektirati prisutnost određene značajke u ulaznoj mapi neovisno o njenoj lokaciji (Goodfellow et al., 2016). Poduzorkovanje se vrši propuštanjem dijelova ulazne mape značajki kroz pre-definiranu funkciju. Primjer su sažimanje maksimalnom vrijednošću (engl. *max pooling*) koje uzima samo maksimalnu vrijednost unutar prozora sažimanja i sažimanje srednjom vrijednošću (engl. *average pooling*) koje uzima srednju vrijednost značajki unutar prozora (2.5).



Slika 2.4: Unatražna konvolucija s konvolucijskom jezgrom dimenzije $k = 3$, korakom $K = 2$, bez ispune do rubova.



Slika 2.5: Slike prikazuju sažimanje a) maksimalnom vrijednošću b) srednjom vrijednošću za prozor sažimanja dimenzije $k = 2$ s korakom $K = 2$.

Dimenzije izlazne mape značajki su određene veličinom prozora sažimanja $DJ = k * k$ (engl. *pool size*), korakom sažimanja K (engl. *stride*), ispunom do rubova I (engl. *padding*) i dimenzijama ulaza h i w. Uloge pojedinih parametara sloja sažimanja slijede one konvolucijskog sloja te izračun dimenzija izlazne mape značajki slijedi jednadžbu 2.3.

Normalizacija po grupi

Gradijenti neuronske mreže se propagiraju kroz slojeve uslijed unatražnog prolaza, stoga visoke vrijednosti ažuriranja parametara u posljednjim slojevima mogu dovesti do iznimno visoke vrijednosti ažuriranja parametara u početnim slojevima mreže zbog množenja gradijenata. Obrat je također moguć s malim iznosima gradijenata. Ovakve pojave nazivamo eksplodirajućim i nestajućim gradijentima respektivno te su jedan od uzročnika nestabilnog procesa učenja mreža i nemogućnosti učenja. Normalizacija po grupama (engl. *batch normalization*) (Ioffe i Szegedy, 2015) pruža elegantan način ublažavanja navedenih problema. Neuronske mreže lakše uče kada su podaci na ulazu slojeva normalizirani, odnosno njihova srednja vrijednost je 0 i varijanca 1 (Goodfellow et al., 2016). Normalizacija po grupama (1) vrši normalizaciju mini-grupe oko srednje vrijednosti 0 s varijancom 1, te provodi parametrizirano skaliranje i pomak mini-grupe kako ne bi došlo do gubitka bitnih informacija o značajkama. Nadalje, Ioffe i Szegedy (2015) navode kako normalizacija po grupama vrši regularizaciju dubokog modela te omogućuje korištenje veće stope učenja.

Normalizacije po grupi na izlazu konvolucijskih slojeva se vrše za svaku mapu značajki zasebno.

Aktivacijske funkcije

Slojevima nelinearnih aktivacijskih funkcija osigurava se nelinearnost transformacija između slojeva neuronske mreže. Sposobnost vršenja nelinearnih transformacija kroz slojeve neuronskoj mreži omogućuje učenje nelinearnosti koje postoje u podacima. Ovdje pružamo matematičke definicije, prikaze i kratak komentar onih aktivacijskih funkcija koje su korištene za ostvarenje dubokih modela u radu. Čitatelja usmjeravamo prema (Goodfellow et al., 2016; Xu et al., 2015) za dublju analizu važnosti aktivacijskih funkcija i studiju uspješnosti različitih aktivacijskih funkcija iz porodice $ReLU$.

Sigmoidalna funkcija je monotono rastuća funkcija s kodomenom $(0, 1)$, gdje se izlaz asimptotski približava 0 kako se vrijednosti ulaza približavaju $-\infty$, odnosno 1 kako se vrijednosti ulaza približavaju $+\infty$ (2.5). Glavni problem pri uporabi sigmo-

Algoritam 1 Normalizacija po grupama. Parametri koji se uče su β i γ. η osigurava stabilnost operacije dijeljenja.

1: Pripremi mini-grupu B primjera $\{x^{(1)}, ..., x^{(M)}\}$

2: Izračunaj srednju vrijednost mini-grupe:

$$\mu_B \leftarrow \frac{1}{M}\sum_{i=1}^{M} x_i.$$

3: Izračunaj varijancu mini-grupe:

$$\sigma_B^2 \leftarrow \frac{1}{M}\sum_{i=1}^{M}(x_i - \mu_B)^2.$$

4: Provedi normalizaciju nad primjerima:

$$\hat{x}^{(i)} \leftarrow \frac{x^{(i)} - \mu_B}{\sqrt{\sigma_B^2 + \eta}}$$

5: Provedi skaliranje i pomak primjera:

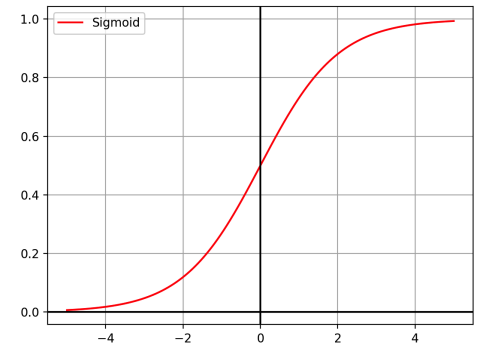
$$y^{(i)} \leftarrow \gamma\hat{x}^{(i)} + \beta$$

idalne aktivacijske funkcije predstavlja nestajući gradijent koji se javlja za jako velike vrijednosti ulaza u pozitivnom ili negativnom smjeru. Tada kažemo da je funkcija u zasićenju. Spomenimo još kako izlaz funkcije nije centriran oko 0 te je računanje izlaza eksponencijalne funkcije računalno skupa matematička operacija.

Funkcija tangens hiperbolni je monotono rastuća funkcija s kodomenom $(-1, 1)$, gdje se izlaz asimptotski približava -1 kako se vrijednosti ulaza približavaju $-\infty$, odnosno 1 kako se vrijednosti ulaza približavaju $+\infty$ (2.6). Tangens hiperbolni ne rješava problem nestajućeg gradijenta sigmoidalne funkcije, no posjeduje prednosti poput lakšeg prolaz gradijenata unatrag za negativne vrijednosti ulaza i centriranost izlaza funkcije oko 0.

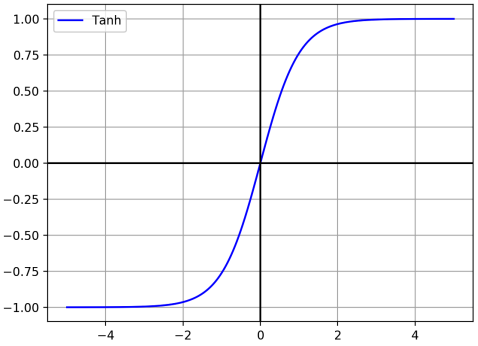
ReLU (engl. *rectified linear unit*) aktivacijska funkcija je rastuća funkcija s kodomenom $(0, +\infty)$, gdje sve vrijednosti ulaza < 0 na izlazu poprimaju vrijednost 0, dok se za vrijednosti ulaza ≥ 0 funkcija ponaša kao identiteta (2.7). Glavna razlika ReLU aktivacijske funkcije naspram sigmoidalne funkcije i tangensa hiperbolnog jest nepostojanje faze zasićenja funkcije pri određenim vrijednostima ulaza. Ovo za posljedicu ima dvije glavne prednosti: rješenje problema nestajućeg gradijenta i brža konvergencija pri učenju (Nair i Hinton, 2010). ReLU aktivacija omogućava računalno

$$\sigma(x) = \frac{1}{1 + e^{-x}} \qquad (2.5)$$



Slika 2.6: Sigmoidalna funkcija

$$tanh(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}} \qquad (2.6)$$



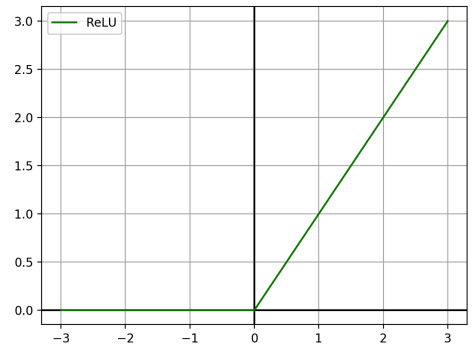
Slika 2.7: Tangens hiperbolni

jednostavan izračun izlaza pri prolazu unaprijed, no za sve ulaze < 0 onemogućuje tok gradijenata pri unatražnom prolazu.

Porodica LeakyReLU (engl. *leaky rectified linear unit*) aktivacijskih funkcija čine funkcije slične ReLU-u, s različitim strategijama izračuna kodomene za negativne vrijednosti ulaza. Ovdje navodimo samo LeakyReLU aktivaciju s fiksnim faktorom množenja negativnih vrijednosti ulaza α (2.8). Ova strategija omogućuje protok gradijenata negativnih vrijednosti ulaza pri unatražnom prolazu skaliranih za faktor α, što dovodi do znatnog poboljšanja performansi određenih arhitektura (Xu et al., 2015).

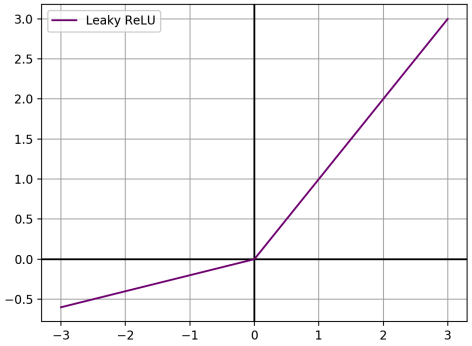
Softmax funkcija (2.9) kao ulaz prima vektor realnih vrijednosti s K elemenata te ga normalizira u vektor vjerojatnosne distribucije s K vjerojatnosti. Vektor realnih vrijednosti po ulazu u softmax sloj može imati negativne vrijednosti i vrijednosti veće od 1 te suma svih vrijednosti ne treba iznositi 1. Vektor realnih vrijednosti na izlazu softmax sloja ima sve pozitivne vrijednosti iz intervala $[0, 1]$, čija ukupna suma iznosi 1, kako bi mogle biti interpretirane kao iznosi vjerojatnosti. Softmax funkciju interpretiramo kao glatku aproksimaciju indikatorske funkcije.

$$ReLU(x) = \max\{0, x\} \quad (2.7)$$



Slika 2.8: ReLU

$$LReLU(x) = \max\{\alpha x, x\} \quad (2.8)$$



Slika 2.9: Leaky ReLU

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z^j}}; \quad i = 1, ..., K; \quad \mathbf{z} = (z_1, ..., z_K) \in \mathbb{R}^K \quad (2.9)$$

2.2. Optimizacijski postupak

Optimizacijski postupak podrazumijeva traženje parametara modela koji zadovoljavaju funkciju cilja koja opisuje željeno konačno ponašanje modela. Postupak optimizacije se obavlja do zadovoljavanja uvjeta konvergencije ili dostizanja određenog uvjeta zaustavljanja.

Učenje širenjem gradijenata unatrag (engl. *backpropagation*) kroz slojeve je osnovni optimizacijski postupak korišten kod učenja neuronskih mreža. Gradijentni spust je iterativan način optimizacije funkcije cilja kada je istu potrebno minimizirati. Funkciju cilja tada često zovemo funkcijom gubitka ili funkcijom cijene $J(\theta)$ parametriziranu parametrima modela θ. Gradijentni spust ažurira parametre modela u

suprotnom smjeru od gradijenata funkcije cijene skalirane stopom učenja η. Formalno:

$$\theta^{(i+1)} = \theta^{(i)} - \eta \nabla_\theta J(x^{(i)}, y^{(i)}; \theta). \tag{2.10}$$

Stopa učenja je hiperparametar koji određuje do koje mjere novonastale informacije nadjačavaju stare informacije. Previsoka stopa učenja može rezultirati skokom parametara preko minimuma funkcije, dok preniska stopa može dovesti do predugog učenja koje nikada ne konvergira, ili ostaje zarobljeno u nepoželjnom lokalnom minimumu. Stalne stope učenja su uvijek manje od 1 kako bi postupak učenja mogao konvergirati. Kako bi se postigla brža konvergencija, spriječile oscilacije i izbjegli nepoželjni lokalni minimumi, stopa učenja se često mijenja tijekom treninga, bilo u skladu s rasporedom učenja ili korištenjem adaptivne stope učenja. Optimizacijski postupci u ovom radu koriste algoritam ADAM za ažuriranje vrijednosti parametara modela tijekom učenja. ADAM izračunava adaptivnu stopu učenja za različite parametre iz procjena prvog i drugog momenta gradijenata, te kombinira metode momenta i adaptivnog pomaka što pospješuje učenje dubokih modela (Kingma i Ba, 2014).

Konačno, optimizacijski postupci se dijele na postupke koji koriste čitav skup primjera pri izračunu gradijenta funkcije cijene (engl. *batch*), samo jedan uzorkovani primjer (engl. *stohastic*) ili mini-grupu uzorkovanih primjera (engl. *mini-batch*). Pri optimizaciji modela dubokog učenja najviše se koristi učenje nad mini-grupama. Aproksimacija gradijenta skupa mini-grupom brže vodi do konvergencije i unosi dodatni šum koji je koristan pri optimizaciji ne-konveksne funkcije cilja, jer može izvući učenje iz nepoželjnih lokalnih minimuma ili sedlastih područja u kojima je zapeo. Svi algoritmi u ovom radu koriste učenje nad mini-grupama.

3. Generativni suparnički modeli

Učenje karakterističnih značajki s potencijalom za višestruku uporabu iz velikih neoznačenih skupova podataka je aktivno područje istraživanja. U kontekstu računalnog vida potencijal leži u velikim neoznačenim skupovima slika i videa koji mogu poslužiti za formiranje relevantnih posrednih značajki koje kasnije mogu doprinijeti različitim nadziranim zadacima poput klasifikacije slika.

Generativni suparnički modeli (Goodfellow et al., 2014a) omogućuju izlučivanje značajki iz slikovnih podataka s potencijalom ponovnog korištenja komponenti modela za izvlačenja bitnih karakteristika pri nadziranom učenju (Radford et al., 2015). Zbog izostanka klasične funkcije cilja isti se nude kao atraktivan odabir za potrebe učenja kompleksnih reprezentacija prisutnih u skupu podataka za učenje.

U ovom poglavlju predstavljamo koncept generativnih suparničkih modela, generativne suparničke mreže i duboke konvolucijske generativne suparničke mreže koje su naprednije inačice potonjih te postižu izvrsne rezultate na zadatku uzorkovanja novih primjera iz distribucije faze učenja u području računalnog vida.

3.1. Generativne suparničke mreže

Generativni suparnički model (Goodfellow et al., 2014a) je okvir koji se sastoji od generatorske i diskriminatorske komponente, gdje su generator i diskriminator opisani diferencijabilnim parametriziranim funkcijama. Zadatak generatora je što bolje opisati (rekreirati) distribuciju podataka za učenje, dok je zadatak diskriminatora specijalizirati se u ocijeni da li određeni primjer pripada distribuciji podataka za učenje ili distribuciji generatora. Postupak učenja generativnog suparničkog modela odgovara igri između dva igrača (3.1), gdje generator konstantno pokušava nadmudriti diskriminatora proizvodeći primjere koji su sve sličniji podatcima distribucije učenja, dok diskriminator mora održati korak s generatorom kroz učenje visoko-diskriminativnih značajki koje karakteriziraju distribuciju podataka za učenje. U prostoru svih proizvoljnih diferencijabilnih parametriziranih funkcija beskonačnog kapaciteta jedinstveno rješenje

postoji (Goodfellow et al., 2014a) te se postiže u trenutku kada diskriminator više ne može razlikovati stvarne od umjetnih primjera te vjerojatnost ispravne procjene za dani primjer iznosi 0.5. Ova netočnost diskriminatora proizlazi iz sposobnosti generatora da stvori primjere koji su vrlo slični primjerima faze učenja, odnosno uspješno opisuje distribuciju podataka faze učenja.

$$\min_G \max_D V(D,G) = \mathbb{E}_{\mathbf{x}\sim p_{data}(\mathbf{x})}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z})}[\log(1 - D(G(\mathbf{z})))] \tag{3.1}$$

3.1: Kako jedna komponenta prilikom učenja ne može mijenjati parametre druge komponente, učenje se provodi kroz igranje minimax igre između dva igrača.

Navedeni okvir omogućuje širinu pri odabiru specifičnih algoritama za modele i optimizacijske algoritme. Ovaj rad slijedi Goodfellow et al. (2014a) u razmatranju posebnog slučaja generativnih suparničkih modela, gdje generator stvara primjere propuštajući slučajan šum kroz višeslojni perceptron, dok je diskriminator također višeslojni perceptron. Ovakva konfiguracija omogućuje treniranje obaju modela uz pomoć algoritma unatražnog učenja (engl. *backpropagation algorithm*), dok se uzorkovanje primjera iz naučene distribucije vrši unaprijednim prolazom kroz generatorsku mrežu. Ovu inačicu modela stoga nazivamo generativnim suparničkim *mrežama* (engl. ***G**enerative **A**dversarial **N**etworks*).

Generator detaljnije opisujemo kao diferencijabilnu funkciju G parametriziranu s θ_g koja na ulazu prima vektor šuma $\mathbf{z}$ uzorkovan iz uniformne distribucije p_z te na izlazu daje $\widetilde{\mathbf{x}}$. Pojednostavljeno, zadatak generatora je jednostavnu distribuciju p_z što više približiti stvarnoj distribuciji podataka p_{data}. Jednadžba 3.2 (Goodfellow et al., 2014a) opisuje funkciju cilja koju generator želi minimizirati:

$$\frac{1}{M}\sum_{i=1}^{M}\log(1 - D(G(\mathbf{z}^{(i)}))). \tag{3.2}$$

U praksi jednadžba (3.2) u fazi učenja rezultira gradijentima koji nisu zadovoljavajući za učenje generatora G. Rano u fazi učenja, dok je G loš u generiranju kvalitetnih primjera, diskriminator D može odbaciti generirane primjere s visokom pouzdanošću, jer su znatno drugačiji od primjeraka za učenje. U tom slučaju kod $\log(1 - D(G(\mathbf{z})))$ dolazi do zasićenja (Goodfellow et al., 2014a). Funkciju cilja generatora stoga mijenjamo u (3.3) koju generator želi maksimizirati. Ovakva funkcija cilja odražava originalnu ideju pri učenju suparničkih mreža, dok rezultira s boljim gradijentima za G u

ranim fazama učenja:

$$\frac{1}{M}\sum_{i=1}^{M}\log D(G(\mathbf{z}^{(i)})). \tag{3.3}$$

Diskriminator detaljnije opisujemo kao diferencijabilnu funkciju D parametriziranu s θ_d koja na ulazu prima primjer x nad kojim vrši binarnu klasifikaciju, gdje je izlaz funkcije vjerojatnost pripadnosti distribuciji podataka za učenje. Jednadžba (3.4) (Goodfellow et al., 2014a) opisuje funkciju cilja koju diskriminator želi minimizirati:

$$\frac{1}{M}\sum_{i=1}^{M}[\,\log D(\mathbf{x}^{(i)}) + \log(1 - D(G(\mathbf{z}^{(i)}))\,]. \tag{3.4}$$

Treniranje suparničkog para mreža se zasniva na igri sa sumom nula, odnosno minimax igri, gdje postoji samo jedan pobjednik. Model konvergira kada generator i klasifikator dođu u poziciju Nasheve ravnoteže. Nashova ravnoteža igre je skup strategija, jedna za svakog igrača, takva da igrači nemaju poticaj da promjene svoju strategiju u odnosu na drugog igrača. Pronalaženje Nasheve ravnoteže funkcije cilja 3.1 teži je zadatak od standardne optimizacije funkcije cilja. Algoritam učenja generativnih suparničkih mreža stohastičkim gradijentnim spustom (2) predložen od strane Goodfellow et al. (2014a) naizmjenično trenira diskriminator i generator. Korištenje gradijentnog spusta s ciljem optimizacije funkcija cilja pojedinih komponenti suparničkog para ne garantira konvergenciju minimax igre te je glavni uzrok nestabilnosti prisutne pri učenju generativnih suparničkih mreža.

Velika prednost generativnih suparničkih modela naspram ostalih generativnih metoda leži u odijeljenosti generatorske komponente i podataka za učenje. Generator ne biva ažuriran gradijentima rekonstrukcijskog gubitka, već gradijentima diskriminatora. Iz ovog razloga, bitne značajke skupa za učenje ne mogu biti direktno kopirane u parametre generatora kako bi se ostvarile bolje performanse pri sintezi novih primjera (Goodfellow et al., 2014a).

S druge strane, nedostatak generativnih suparničkih modela jest nemogućnost robusne provjere poklapanja između p_g i p_{data}, odnosno nemogućnost robusne provjere pojavljivanja test seta pod p_g, odnosno nemogućnost eksplicitne reprezentacije $p_g(x)$ (Goodfellow et al., 2014a). Također, distribucije podataka iz stvarnog svijeta su kompliciranе i višemodalne, odnosno mogu imati više različitih vrhova s koncentriranim podgrupama uzoraka. Generativni suparnički modeli zbog postupka učenja opisanog kroz minimax igru mogu završiti u fazi degradacije učenja (engl. *mode collapse*), gdje generator za proizvoljan šum generira isti uzorak, odnosno nauči generirati samo jedan mod podataka kako bi zadovoljio uvjete igre (Goodfellow et al., 2014a).

Algoritam 2 Treniranje generativnih suparničkih mreža algoritmom stohastičkog gradijentnog spusta s mini-grupama.

1: **for** broj iteracija učenja **do**

2: Uzorkuj mini-grupu vektora šuma $\{\mathbf{z}^{(1)}, ..., \mathbf{z}^{(M)}\}$ iz distribucije p_z.

3: Pripremi mini-grupu primjera $\{\mathbf{x}^{(1)}, ..., \mathbf{x}^{(M)})\}$ iz distribucije p_{data}.

4: Ažuriraj parametre θ_d diskriminatora D pomicanjem njihovog stohastičkog gradijenta uzlazno:

$$\nabla_{\theta_d} \frac{1}{M} \sum_{i=1}^{M} [\, \log D(\mathbf{x}^{(i)}) + \log(1 - D(G(\mathbf{z}^{(i)}))) \,].$$

5: Uzorkuj mini-grupu vektora šuma $\{\mathbf{z}^{(1)}, ..., \mathbf{z}^{(M)}\}$ iz distribucije p_z.

6: Ažuriraj parametre θ_g generatora G pomicanjem njihovog stohastičkog gradijenta uzlazno:

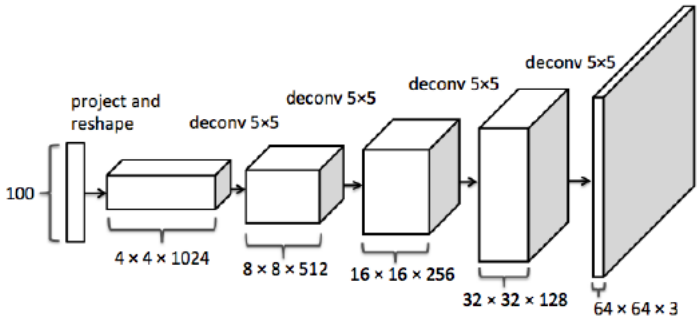
$$\nabla_{\theta_g} \frac{1}{M} \sum_{i=1}^{M} [\, \log(D(G(\mathbf{z}^{(i)})) \,].$$

3.2. Duboke konvolucijske inačice generativnih suparničkih mreža

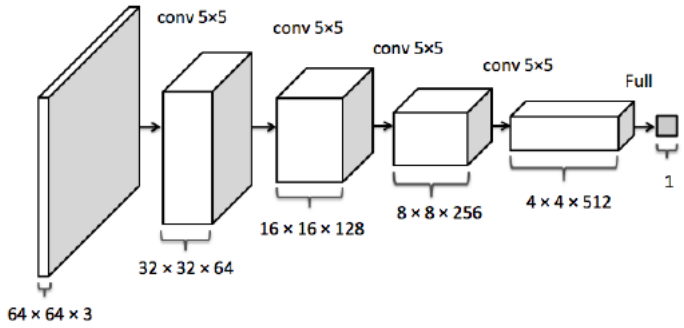
Generator i diskriminator su parametrizirane diferencijabilne funkcije često implementirane kao duboke neuronske mreže. Radford et al. (2015) predlažu ograničenja nad arhitekturom dubokih konvolucijskih neuronskih mreža koje upotrijebljene pri dizajnu komponenti suparničkog para osiguravaju stabilniji postupak učenja. Duboke konvolucijske neuronske mreže kod generatora omogućuju učenje prostornog povećanja kroz korištenje slojeva s unatražnom konvolucijom (engl. *transpose convolution*), dok se kod diskriminatora konvolucijskim slojevima ostvaruje prostorno sažimanje.

Prvo ograničenje sugerira korištenje konvolucijskih slojeva s pomakom umjesto determinističkih slojeva sažimanja. Ideja je omogućiti neuronskoj mreži da sama nauči najbolji režim sažimanja tijekom treninga.

Drugo ograničenje eliminira potpuno-povezane slojeve u generatoru i diskriminatoru. Generator je na ulazu predstavljen s vektorom $\mathbf{z}$ uzorkovanim iz uniformne distribucije p_z koji odmah sudjeluje u procesu transponirane konvolucije. Nadalje, tenzor na izlazu posljednjeg konvolucijskog sloja diskriminatora se izravnava i povezuje s posljednjim slojem kojeg čini sigmoidalna aktivacijska funkcija.



(a)



(b)

Slika 3.1: Slika a) prikazuje arhitekturu dubokog konvolucijskog generatora, dok slika b) prikazuje arhitekturu dubokog konvolucijskog diskriminatora. Preuzeto iz Yeh et al. (2017)

Posljednje ograničenje uvodi uporabu normalizacije po grupama (Ioffe i Szegedy, 2015) koja stabilizira postupak učenja normalizirajući ulaz u sloj mreže oko srednje vrijednosti nula i varijance jedan. Normalizacija po grupama pomaže s problemima u učenju koji se javljaju zbog loše inicijalizacije parametara modela te omogućuje lakši protok gradijenata kroz duboki model. Normalizacija po grupama se izostavlja u posljednjem sloju mreže generatora i prvom sloju mreže diskriminatora.

U nastavku rada su korištene samo duboke konvolucijske inačice generativnih suparničkih mreža pri čemu točni opisi arhitekture prethode objavu rezultata.

4. Problem detekcije izvan-distribucijskih primjeraka

4.1. Procjena nesigurnosti predikcije

Duboke neuronske mreže postižu najbolje klasifikacijske rezultate u domenama poput klasifikacije slika (Girshick, 2015), prepoznavanja govora (Hannun et al., 2014) i procjeni određenih medicinskih dijagnoza (Caruana et al., 2015). Međutim, takvi diskriminativni modeli ne nude informaciju o pouzdanosti modela u vlastiti klasifikacijski rezultat. Nemogućnost mnogih klasifikatora da naprave procjenu pouzdanosti vlastite predikcije predstavlja ozbiljan problem za njihovo usvajanje i primjenu u stvarnom svijetu. Kao prirodan primjer se nudi model za procjenu medicinskih dijagnoza na temelju podataka o pacijentima, gdje visoka točnost modela nije jedini imperativ, već i njegova mogućnost prepoznavanja zahtjevnih slučajeva kod kojih ne može donijeti pouzdanu odluku, gdje takvi slučajevi trebaju biti proslijeđeni na daljnje razmatranje od strane liječnika. Mali broj pogrešnih dijagnoza može ozbiljno naštetiti perspektivi modela strojnog učenja u području medicine. Za mnogo iscrpnije razmatranje konkretnih problema sigurnosti sustava umjetne inteligencije usmjeravamo čitatelja na Amodei et al. (2016).

Kod razmatranja problema neosnovanog optimizma diskriminativnih modela pri procjeni, Hendrycks i Gimpel (2016) nude osnovicu za daljnje istraživanje stavljanjem fokusa na prediktivnu distribuciju koju model proizvodi te definiraju standardizirane mjere za određivanje performansi detekcije pogrešno klasificiranih i izvan-distribucijskih primjeraka.

Softmax (2.9) je brzo-rastuća eksponencijalna funkcija kod koje male promjene na ulazu za posljedicu imaju značajne promjene na izlazu funkcije. Kako je funkcija softmax glatka aproksimacija indikatorske funkcije, iznimno rijetko izlaz softmax sloja poprima oblik uniformne distribucije za izvan-distribucijske primjerke. Hendrycks i Gimpel (2016) daju primjer dubokog diskriminativnog modela za klasifikaciju MNIST

znamenki (LeCun et al., 2010) koji daje samouvjerenu predikciju od 91% za slučajan šum na svom ulazu, gdje se najviši iznos na izlazu softmax sloja tumači kao razina pouzdanosti modela u predikciju.

Hendrycks i Gimpel (2016) ukazuju na trend nižih vrijednosti vjerojatnosti predikcije za netočno klasificirane i izvan-distribucijske primjere naspram točno klasificiranih primjera. Istraživanje stoga može biti usmjereno ka modeliranju prediktivnih distribucija točno klasificiranih primjera za efektivnu identifikaciju i odjeljivanje netočnih primjera, iako promatranje iznosa predikcije točno klasificiranog primjerka u izolaciji može biti zavaravajuće. Ovo otvara put uvođenju jednostavnog detektora izvan-distribucijskih primjeraka zasnovanog na pragu. Detektor $g(\mathbf{x}) : X \rightarrow \{0, 1\}$ pridružuje oznaku 1, ako je iznos povjerenja u predikciju $g(\mathbf{x})$ iznad nekog praga δ, i oznaku 0 inače:

$$g(\mathbf{x}) = \begin{cases} 1 & \text{ako je } q(\mathbf{x}) \geq \delta \\ 0 & \text{inače} \end{cases} \tag{4.1}$$

Preciznije, detektor može biti izgrađen nad naučenim klasifikatorom, gdje kao iznos povjerenja u predikciju promatramo iznos najviše vrijednosti prediktivne distribucije i donosimo odluku o prihvaćanju ili odbacivanju predikcije kao nepouzdane. Na ovoj ideji se zasniva doprinos rada Lee et al. (2017), razmatranje kojega je središnji dio ovog rada.

4.2. Funkcija cijene dvostruke pouzdanosti

Lee et al. (2017) problem neopravdanog optimizma *dubokih neuronskih mreža* povezuju s problemom detekcije izvan-distribucijskih primjeraka, gdje neopravdani optimizam proizlazi iz nemogućnosti procjene modela pripada li primjerak domeni faze treniranja ili se radi izvan-distribucijskom primjerku. Ovo formalno možemo predstaviti kao problem binarne klasifikacije. Neka primjer $\mathbf{x} \in X$ pripada razredu $y \in Y = \{1, ..., K\}$, gdje su $\mathbf{x}$ i y slučajne varijable koje slijede zajedničku distribuciju $P_{in}(\mathbf{x}, y) = P_{in}(y|\mathbf{x})\ P_{in}(\mathbf{x})$. Pretpostavljamo da je klasifikator $P_\theta(y|\mathbf{x})$ naučen nad podacima koji slijede distribuciju $P_{in}(\mathbf{x}, y)$, gdje θ označava parametre naučenog modela. S $P_{out}(\mathbf{x})$ označavamo stranu distribuciju koja je "daleko" od distribucije iz faze učenja $P_{in}(\mathbf{x})$. Središnji problem je predvidjeti da li primjer $\mathbf{x}$ pripada P_{in} ili P_{out} kroz uporabu dobro kalibriranog klasifikatora $P_\theta(y|\mathbf{x})$. Odnosno, želimo dobiti detektor $g(\mathbf{x}) : X \rightarrow \{0, 1\}$ koji pridaje oznaku 1 primjeru koji pripada domeni faze učenja i oznaku 0 inače.

Doprinos rješavanju problema detekcije izvan-distribucijskih primjeraka P_{out} od strane Lee et al. (2017) čini predložak za izgradnju klasifikatora koji bolje odjeljuje prediktivne distribucije unutar- i izvan-distribucijskih primjeraka i omogućuje izgradnju uspješnijeg detektora zasnovanog na pragu koji iznos najviše vrijednosti prediktivne distribucije tumači kao iznos povjerenja u predikciju. Predložak se sastoji od tri različita doprinosa:

Prvo, funkcija cijene klasifikatora proširuje se komponentom koja na predstavljenim izvan-distribucijskim primjerima minimizira Kullback-Leibler divergenciju (4.2) između prediktivne distribucije i uniformne distribucije, kako bi model davao manje pouzdana predviđanja za izvan-distribucijske primjerke. Očekivanje je kako će ovime distribucije P_{in} i P_{out} postati odjeljivije. Ovakvu funkciju cijene u nastavku rada zovemo funkcijom dvostruke pouzdanosti[1], originalno engl. *confidence loss*.

Nadalje, za treniranje gore navedenog klasifikatora potrebno je koristiti izvan-distribucijske primjere u fazi učenja koje je vrlo teško uzorkovati. Takvi primjeri često nisu a priori poznati za problem koji model pokušava riješiti ili je nemoguće (neisplativo) uzorkovati čitav prostor koji prekrivaju izvan-distribucijski primjeri. Ovdje se koriste generativne suparničke mreže za generiranje najkorisnijih primjera iz P_{out}. Najkorisniji primjeri su oni koji leže u području niske gustoće distribucije P_{in}, odnosno granični primjeri distribucije P_{in} prema P_{out}.

Konačno, autori predlažu postupak učenja koji uči središnji klasifikator i suparnički par mreža u tandemu, minimizirajući naizmjenice novo-predložen gubitak središnjeg klasifikatora i novo-predložen gubitak suparničkog para mreža, u shemi učenja gdje se kroz treniranje mreže međusobno unaprjeđuju.

4.2.1. Klasifikator

Lee et al. (2017) proširuju funkciju cijene unakrsne entropije klasifikatora komponentom Kullback-Leiblerove divergencije (4.2) kako bi model optimizacijskim postupkom prediktivnu distribuciju izvan-distribucijskih primjera P_{out} što više približio uniformnoj distribuciji, dok bi prediktivna distribucija primjera faze učenja P_{in} ostala nepromijenjena (4.3). Kullback-Leiblerova divergencija je mjera koliko određena vjerojatnosna distribucija $Q(x)$ odudara od očekivane vjerojatnosne distribucije $P(x)$, gdje su $Q(x)$ i $P(x)$ diskretne vjerojatnosne distribucije i X skup svih mogućih vrijednosti od

[1] Prijevod "dvostruka pouzdanost" je odabran u duhu cilja gdje želimo dobiti središnji klasifikator pouzdan u predviđanja za primjere iz domene faze učenja i pouzdan u prepoznavanje izvan-distribucijskih primjera za koje nije specijaliziran te ih ne može klasificirati.

x. Kullback-Leiblerova divergencija se može interpretirati kao mjera gubitka informacije kada se distribucija $Q(x)$ koristi za aproksimaciju distribucije $P(x)$.

$$KL\left(P \mid\mid Q\right) = \sum_{x \in X} P(x) \log\left(\frac{P(x)}{Q(x)}\right) \tag{4.2}$$

Jednadžba 4.3 predstavlja funkciju cijene dvostruke pouzdanosti, gdje KL označava Kullback-Leiblerovu divergenciju, $\mathcal{U}(y)$ označava uniformnu distribuciju i $\beta > 0$ hiperparametar koji određuje udio komponente Kullback-Leiblerove divergencije u ukupnoj funkciji cijene. Minimizacija ovakve funkcije cijene rezultira klasifikatorom koji pridaje veće vrijednosti predikcije $\max_y P_\theta(y|\mathbf{x})$ primjerima iz domene faze učenja nego izvan-distribucijskim primjerima.

$$\min_\theta \mathbb{E}_{P_{in}(\hat{\mathbf{x}},\hat{y})}[\, -\log P_\theta\left(y = \hat{y}|\hat{\mathbf{x}}\right)\,] + \beta\, \mathbb{E}_{P_{out}(\mathbf{x})}[\, KL\left(\mathcal{U}(y) \mid\mid P_\theta\left(y|\mathbf{x}\right)\right)\,] \tag{4.3}$$

4.2.2. Generativne suparničke mreže

Lee et al. (2017) proširuju funkciju cilja koja opisuje minimax igru između generativnih suparničkih mreža (Goodfellow et al., 2014a) komponentom kojom naučeni klasifikator s dvostrukom pouzdanošću sudjeluje u oblikovanju generatora za stvaranje najkorisnijih primjera iz P_{out}, odnosno primjera koji leže na samoj granici distribucija P_{in} i P_{out} (4.4). U 4.4, θ označava parametre naučenog klasifikatora s dvostrukom pouzdanošću, KL označava Kullback-Leiblerovu divergenciju, $\mathcal{U}(y)$ označava uniformnu distribuciju i $\beta > 0$ hiperparametar koji određuje udio komponente Kullback-Leiblerove divergencije u ukupnoj funkciji cilja. Optimizacija ovakve funkcije cilja rezultira generatorom koji generira primjere za koje klasifikator s dvostrukom pouzdanošću daje prediktivnu distribuciju sličnu uniformnoj, implicitno ih karakterizirajući kao izvan-distribucijske, a koji zbog djelovanja diskriminatora nisu daleko u prostoru primjera od unutar-distribucijskih.

$$\begin{aligned}\min_G \max_D \beta\, \mathbb{E}_{P_G(\mathbf{x})}\left[KL\left(\mathcal{U}(y) \mid\mid P_\theta\left(y|\mathbf{x}\right)\right)\right] \\ + \mathbb{E}_{P_{in}(\mathbf{x})}\left[\log D(\mathbf{x})\right] + \mathbb{E}_{P_G(\mathbf{x})}\left[\log\left(1 - D(\mathbf{x})\right)\right]\end{aligned} \tag{4.4}$$

4.2.3. Pristupi treniranju

Odjeljak 4.2.2 sugerira korištenje naučenog klasifikatora s dvostrukom pouzdanošću za unaprjeđenje generatora izvan-distribucijskih primjeraka tijekom učenja. Obrat je

također moguć, i čini veliki dio motivacije rada, a radi se o korištenju naučenog generatora izvan-distribucijskih primjeraka prilikom učenja središnjeg klasifikatora. Očigledno je da se modeli međusobno nadopunjuju u fazi učenja, što prirodno vodi ka združenoj shemi učenja klasifikatora s dvostrukom pouzdanošću i generatora izvan-distribucijskih primjeraka - odnosno generativnog suparničkog para mreža čiji je on dio. Združena funkcija cilja (Lee et al., 2017) pritom izgleda:

$$\min_{G}\max_{D}\min_{\theta} \underbrace{\mathbb{E}_{P_{in}(\hat{\mathbf{x}},\hat{y})}[\,-\log P_\theta\,(y=\hat{y}|\hat{\mathbf{x}})\,]}_{\text{(a)}} + \beta \underbrace{\mathbb{E}_{P_G(\mathbf{x})}[\,KL\,(\mathcal{U}(y)\,||\,P_\theta\,(y|\mathbf{x}))\,]}_{\text{(b)}} + \underbrace{\mathbb{E}_{P_{in}(\hat{\mathbf{x}})}[\,\log D(\hat{\mathbf{x}})\,] + \mathbb{E}_{P_G(\mathbf{x})}[\,\log(1-D(\mathbf{x}))\,]}_{\text{(c)}} \tag{4.5}$$

Dijelovi (a) i (b) odgovaraju funkciji cijene klasifikatora s dvostrukom pouzdanošću, dok dijelovi (b) i (c) odgovaraju funkciji cilja generativnog suparničkog para mreža. Dio (b) sadrži komponentu KL divergencije koja je zajednička klasifikatoru i generativnom suparničkom paru mreža pod zajedničkim režimom učenja, a kroz koju se komponente međusobno unaprjeđuju.

Algoritam 3 opisuje združenu shemu učenja, gdje se kroz optimizacijski proces naizmjenice ažuriraju parametri središnjeg klasifikatora $\{\theta_{cc}\}$ i generativnog suparničkog para mreža $\{\theta_g, \theta_d\}$. Lee et al. (2017) objavljuju najbolje dobivene rezultate upravo s navedenim režimom učenja.

U sklopu ovog rada razmotrena su dva dodatna režima učenja koje ovdje navodimo, s rezultatima eksperimenata objavljenim u nastavku.

Algoritam 4 predstavlja pristup kod kojeg se u prvom dijelu treniraju generativne suparničke mreže prema algoritmu 2. Ideja je iskoristiti nesavršenost generatora suparničkog para učenog nad p_{in} pri rekonstrukciji p_{in} za generiranje izvan-distribucijskih primjera pri učenju središnjeg klasifikatora. U drugom dijelu se trenira središnji klasifikator s dvostrukom pouzdanošću, gdje su izvan-distribucijski primjeri za KL komponentu funkcije cijene 4.3 uzorkovani uz pomoć naučenog nesavršenog generatora unutar-distribucijskih primjera iz prvog dijela.

Algoritam 5 predstavlja pristup kod kojeg se prvo trenira jednostavan klasifikator s dvostrukom pouzdanošću, gdje su izvan-distribucijski primjeri za KL komponentu funkcije cijene 4.3 uzorkovani iz standardne normalne distribucije. Zatim se treniraju generativne suparničke mreže, gdje jednostavni klasifikator iz prvog koraka zauzima mjesto naučenog detektora izvan-distribucijskih primjeraka u 4.4. Konačno, trenira se

Algoritam 3 Shema združenog učenja detektora i generatora izvan-distribucijskih primjeraka.

for broj iteracija združenog učenja **do**

/* Ažuriranje para suparničkih mreža */

Uzorkuj mini-grupu vektora šuma $\{\mathbf{z}^{(1)}, ..., \mathbf{z}^{(M)}\}$ iz distribucije p_z.

Pripremi mini-grupu primjera $\{(\mathbf{x}^{(1)}, y^{(1)}), ..., (\mathbf{x}^{(M)}, y^{(M)})\}$ iz distribucije p_{in}.

Ažuriraj parametre θ_d diskriminatora D pomicanjem njihovog stohastičkog gradijenta uzlazno:

$$\nabla_{\theta_d} \frac{1}{M} \sum_{i=1}^{M} \left[\log D(\mathbf{x}^{(i)}) + \log(1 - D(G(\mathbf{z}^{(i)}))) \right].$$

Uzorkuj mini-grupu vektora šuma $\{\mathbf{z}^{(1)}, ..., \mathbf{z}^{(M)}\}$ iz distribucije p_z.

Ažuriraj parametre θ_g generatora G pomicanjem njihovog stohastičkog gradijenta silazno:

$$\nabla_{\theta_g} \frac{1}{M} \sum_{i=1}^{M} [\, \log(1 - D(G(\mathbf{z}^{(i)}))) + \beta\, KL\, (\mathcal{U}(y) \,||\, P_{\theta_c}\, (y|G(\mathbf{z}^{(i)}))) \,].$$

/* Ažuriranje središnjeg klasifikatora */

Uzorkuj mini-grupu vektora šuma $\{\mathbf{z}^{(1)}, ..., \mathbf{z}^{(M)}\}$ iz distribucije p_z.

Pripremi mini-grupu primjera $\{(\mathbf{x}^{(1)}, y^{(1)}), ..., (\mathbf{x}^{(M)}, y^{(M)})\}$ iz distribucije p_{in}.

Ažuriraj parametre θ_c središnjeg klasifikatora pomicanjem njihovog stohastičkog gradijenta silazno:

$$\nabla_{\theta_c} \frac{1}{M} \sum_{i=1}^{M} \left[-\log P_{\theta_c}\, (y = y^{(i)}|\mathbf{x}^{(i)}) + \beta\, KL\, (\mathcal{U}(y) \,||\, P_{\theta_c}\, (y|G(\mathbf{z}^{(i)}))) \right].$$

Algoritam 4 Treniranje generatora izvan-distribucijskih primjeraka nakon čega se trenira središnji klasifikator.

for broj iteracija učenja generatora **do**

Uzorkuj mini-grupu vektora šuma $\{\mathbf{z}^{(1)}, ..., \mathbf{z}^{(M)}\}$ iz distribucije p_z.

Pripremi mini-grupu primjera $\{(\mathbf{x}^{(1)}, y^{(1)}), ..., (\mathbf{x}^{(M)}, y^{(M)})\}$ iz distribucije p_{in}.

Ažuriraj parametre θ_d diskriminatora D pomicanjem njihovog stohastičkog gradijenta uzlazno:

$$\nabla_{\theta_d} \frac{1}{M} \sum_{i=1}^{M} [\, \log D(\mathbf{x}^{(i)}) + \log(1 - D(G(\mathbf{z}^{(i)}))) \,].$$

Uzorkuj mini-grupu vektora šuma $\{\mathbf{z}^{(1)}, ..., \mathbf{z}^{(M)}\}$ iz distribucije p_z.

Ažuriraj parametre θ_g generatora G pomicanjem njihovog stohastičkog gradijenta uzlazno:

$$\nabla_{\theta_g} \frac{1}{M} \sum_{i=1}^{M} [\, \log(D(G(\mathbf{z}^{(i)})) \,].$$

for broj iteracija učenja klasifikatora **do**

Uzorkuj mini-grupu vektora šuma $\{\mathbf{z}^{(1)}, ..., \mathbf{z}^{(M)}\}$ iz distribucije p_z.

Uporabom naučenog generatora G i mini-grupe $\{\mathbf{z}^{(1)}, ..., \mathbf{z}^{(M)}\}$ generiraj mini-grupu izvan-distribucijskih primjera $\{\mathbf{u}^{(1)} = G(\mathbf{z}^{(1)}), ..., \mathbf{u}^{(M)} = G(\mathbf{z}^{(M)})\}$.

Pripremi mini-grupu primjera $\{(\mathbf{x}^{(1)}, y^{(1)}), ..., (\mathbf{x}^{(M)}, y^{(M)})\}$ iz distribucije p_{in}.

Ažuriraj parametre θ_c središnjeg klasifikatora pomicanjem njihovog stohastičkog gradijenta silazno:

$$\nabla_{\theta_c} \frac{1}{M} \sum_{i=1}^{M} [\, -\log P_{\theta_c}\,(y = y^{(i)}|\mathbf{x}^{(i)}) + \beta\, KL\,(\mathcal{U}(y)\;||\;P_{\theta_c}\,(y|\mathbf{u}^{(i)})) \,].$$

središnji klasifikator s dvostrukom pouzdanošću, gdje su izvan-distribucijski primjeri za KL komponentu funkcije cijene 4.3 uzorkovani uz pomoć naučenog generatora izvan-distribucijskih primjeraka iz drugog koraka.

Algoritam 5 Treniranje jednostavnog klasifikatora, generatora izvan-distribucijskih primjeraka te konačno središnjeg klasifikatora.

for broj iteracija učenja jednostavnog klasifikatora **do**

Uzorkuj mini-grupu vektora šuma $\{\mathbf{z}^{(1)}, ..., \mathbf{z}^{(M)}\}$ iz distribucije p_{pri}.

Pripremi mini-grupu primjera $\{(\mathbf{x}^{(1)}, y^{(1)}), ..., (\mathbf{x}^{(M)}, y^{(M)})\}$ iz distribucije p_{in}.

Ažuriraj parametre θ_{sc} jednostavnog klasifikatora pomicanjem njihovog stohastičkog gradijenta silazno:

$$\nabla_{\theta_{sc}} \frac{1}{M} \sum_{i=1}^{M} [\, -\log P_{\theta_{sc}}\,(y = y^{(i)}|\mathbf{x}^{(i)}) + \beta\, KL\,(\mathcal{U}(y)\;||\;P_{\theta_{sc}}\,(y|\mathbf{z}^{(i)}))\,].$$

for broj iteracija učenja generatora **do**

Uzorkuj mini-grupu vektora šuma $\{\mathbf{z}^{(1)}, ..., \mathbf{z}^{(M)}\}$ iz distribucije p_z.

Pripremi mini-grupu primjera $\{(\mathbf{x}^{(1)}, y^{(1)}), ..., (\mathbf{x}^{(M)}, y^{(M)})\}$ iz distribucije p_{in}.

Ažuriraj parametre θ_d diskriminatora D pomicanjem njihovog stohastičkog gradijenta uzlazno:

$$\nabla_{\theta_d} \frac{1}{M} \sum_{i=1}^{M} [\, \log D(\mathbf{x}^{(i)}) + \log(1 - D(G(\mathbf{z}^{(i)})))\,].$$

Uzorkuj mini-grupu vektora šuma $\{\mathbf{z}^{(1)}, ..., \mathbf{z}^{(M)}\}$ iz distribucije p_z.

Ažuriraj parametre θ_g generatora G pomicanjem njihovog stohastičkog gradijenta silazno:

$$\nabla_{\theta_g} \frac{1}{M} \sum_{i=1}^{M} [\, \log(1 - D(G(\mathbf{z}^{(i)}))) + \beta\, KL\,(\mathcal{U}(y)\;||\;P_{\theta_{sc}}\,(y|G(\mathbf{z}^{(i)})))\,].$$

for broj iteracija učenja središnjeg klasifikatora **do**

Uzorkuj mini-grupu vektora šuma $\{\mathbf{z}^{(1)}, ..., \mathbf{z}^{(M)}\}$ iz distribucije p_z.

Uporabom naučenog generatora G i mini-grupe $\{\mathbf{z}^{(1)}, ..., \mathbf{z}^{(M)}\}$ generiraj mini-grupu izvan-distribucijskih primjera $\{\mathbf{u}^{(1)} = G(\mathbf{z}^{(1)}), ..., \mathbf{u}^{(M)} = G(\mathbf{z}^{(M)})\}$.

Pripremi mini-grupu primjera $\{(\mathbf{x}^{(1)}, y^{(1)}), ..., (\mathbf{x}^{(M)}, y^{(M)})\}$ iz distribucije p_{in}.

Ažuriraj parametre θ_{cc} središnjeg klasifikatora pomicanjem njihovog stohastičkog gradijenta silazno:

$$\nabla_{\theta_{cc}} \frac{1}{M} \sum_{i=1}^{M} [\, -\log P_{\theta_{cc}}\,(y = y^{(i)}|\mathbf{x}^{(i)}) + \beta\, KL\,(\mathcal{U}(y)\;||\;P_{\theta_{cc}}\,(y|\mathbf{u}^{(i)}))\,].$$

5. Eksperimenti

Performanse središnjeg klasifikatora i generativnog suparničkog para mreža evaluirane su na više različitih skupova podataka različite težine. Sljedeći Hendrycks i Gimpel (2016), metrike performansi središnjeg klasifikatora za primjer na ulazu u obzir uzimaju najveću vrijednost prediktivne distribucije, odnosno najveću vrijednost na izlazu softmax sloja. Rezultate eksperimenata prethode detaljni opisi arhitekture korištenih mreža. Eksperimenti su provedeni nad svim algoritmima iz 4.2.3 s rezultatima predstavljenim kao srednja vrijednost povrh više sjednica učenja. Ovaj pristup koristimo zbog izražene varijabilnosti u performansama središnjeg klasifikatora na izvandistribucijskim skupovima podataka, koju pridjeljujemo suparničkom paru mreža čiji veliki latentni prostor za uzorkovanje primjeraka u različitim konfiguracijama usmjerava središnji klasifikator.

5.1. Podatkovni skupovi

MNIST

MNIST (LeCun et al., 2010) je skup slika rukom pisanih znamenki od 0 do 9, crnobijele boje s dimenzijama 28x28 piksela. Skup se sastoji od 70,000 primjera, od kojih se 60,000 koristi za učenje i 10,000 za testiranje. MNIST je polazni i sveprisutni skup slika za brzo ispitivanje znanstvenih pretpostavki u području računalnog vida. Točnost klasifikacije određenih modela nad skupom dostiže i do 99.79% (Wan et al., 2013), stoga se problem klasifikacije znamenki nad ovim skupom danas smatra riješenim. U ovom radu je skup korišten tijekom rane potrage za najboljim načinom učenja središnjeg klasifikatora i generativnog suparničkog para mreža. Slika 5.1 prikazuje neke od primjera iz skupa.

Slika 5.1: Slučajno odabrana 64 primjera iz podatkovnog skupa MNIST

Slika 5.2: Slučajno odabrana 64 primjera iz podatkovnog skupa Fashion-MNIST

Fashion-MNIST

Fashion-MNIST (Xiao et al., 2017) je skup slika odjevnih predmeta od kojih svaki pripada jednom od 10 razreda. Slike su crno-bijele boje dimenzija 28x28 piksela. Poput MNIST skupa, sastoji se od 70,000 primjera, od kojih se 60,000 koristi za učenje, i 10,000 za testiranje. Fashion-MNIST je zamišljen kao zahtjevnija zamjena MNIST skupa podataka koja slijedi njegov format, kako bi znanstvena zajednica imala kvalitetniji i relevantniji polazni skup podataka za prototipiziranje. Slike iz Fashion-MNIST skupa su korištene u kapacitetu izvan-distribucijskih primjeraka za evaluaciju performansi treniranih modela tijekom rane potrage za najboljim načinom učenja središnjeg klasifikatora i generativnog suparničkog para mreža. Slika 5.2 prikazuje neke od primjera iz skupa.

Slika 5.3: Slučajno odabrana 64 primjera iz podatkovnog skupa SVHN

SVHN

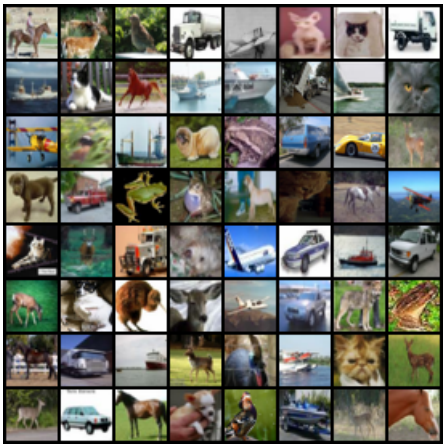
SVHN (Netzer et al., 2011) je skup slika brojeva u boji, uzetih iz stvarnog svijeta, namijenjen rješavanju problema prepoznavanja brojeva i znamenki u stvarnom okruženju. Korištena verzija SVHN skupa se sastoji od slika jedne znamenke s 73,257 primjera za učenje i 26,032 primjera za testiranje. Slike su dimenzija 3x32x32, gdje prva dimenzija predstavlja crveni, zeleni i plavi kanal slike, a zadnje dvije dimenzije definiraju njenu visinu i širinu. SVHN skup je korišten u kapacitetu unutar- i izvan-distribucijske domene pri različitim režimima učenja. Slika 5.3 prikazuje neke od primjera iz skupa.

CIFAR-10

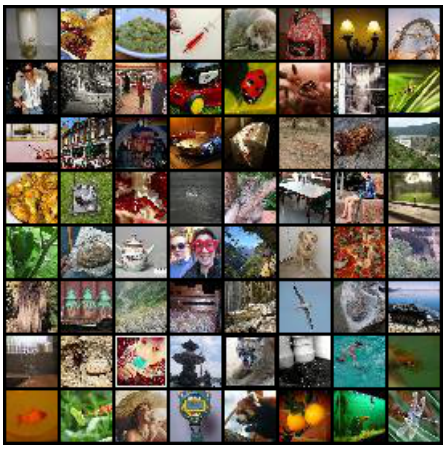
CIFAR-10 (Krizhevsky et al., 2014) je skup prirodnih slika u boji često korišten za učenje algoritama strojnog učenja, poglavito u području računalnog vida. Slike prikazuju objekte koji pripadaju jednom od 10 razreda, poput aviona, automobila i ptica između ostalih. Skup se sastoji od 60,000 slika dimenzija 3x32x32, od kojih se 50,000 koristi za učenje i 10,000 za testiranje. CIFAR-10 skup je korišten u kapacitetu unutar- i izvan-distribucijske domene pri različitim režimima učenja. Slika 5.4 prikazuje neke od primjera iz skupa.

ImageNet

ImageNet (Deng et al., 2009) je velika slikovna baza podataka namijenjena za korištenje pri različitim zadacima u području računalnog vida poput prepoznavanja i ocrtavanja objekata na slikama. ImageNet sadrži slike koje prikazuju više od 20,000 različitih



Slika 5.4: Slučajno odabrana 64 primjera iz podatkovnog skupa CIFAR-10



Slika 5.5: Slučajno odabrana 64 primjera iz podatkovnog skupa ImageNet

razreda objekata, gdje je svaki razred zastupljen s više stotina slika. Ovaj rad koristi podskup navedene baze slika koji sadrži 10,000 testnih slika preko 200 različitih razreda. Korištene slike su poduzorkovane inačice originalnih slika dimenzija 3x32x32. Slike iz ImageNet skupa su korištene u kapacitetu izvan-distribucijskih primjeraka za evaluaciju performansi treniranih modela, gdje su razredi objekata u korištenim slikama disjunktni razredima objekata u skupu za treniranje modela. Slika 5.5 prikazuje neke od primjera iz skupa.

LSUN

LSUN (Yu et al., 2015) je velika slikovna baza scena i objekata nastala procesom polu-automatiziranog obilježavanja. LSUN sadrži blizu 1 milijun slika za svaku od 10 zastupljenih kategorija scena te više od 1 milijun slika za svaku od 20 zastupljenih kategorija objekata. Ovaj rad koristi podskup navedene baze slika koji sadrži 10,000

Slika 5.6: Slučajno odabrana 64 primjera iz podatkovnog skupa LSUN

testnih slika preko 10 različitih scena. Korištene slike su poduzorkovane inačice originalnih slika dimenzija 3x32x32. Slike iz LSUN skupa su korištene u kapacitetu izvan-distribucijskih primjeraka za evaluaciju performansi treniranih modela. Slika 5.6 prikazuje neke od primjera iz skupa.

5.2. Metrike

Performanse naučenog središnjeg klasifikatora promatramo iz dva ugla. Kod podataka iz skupa za učenje nas zanima točnost klasifikacije. Kod podataka iz izvan-distribucijskih skupova nas zanima sposobnost klasifikatora da prepozna pripadnost primjerka distribuciji za koju nije specijaliziran. Dobar klasifikator treniran funkcijom cijene dvostruke pouzdanosti prvi kriterij zadovoljava najvišom vrijednošću prediktivne distribucije koja odgovara razredu kojem primjer pripada, dok drugi kriterij ostvaruje implicitno dajući prediktivnu distribuciju sličnu uniformnoj za izvan-distribucijski primjer na ulazu.

Sljedeći Hendrycks i Gimpel (2016) izlaz softmax aktivacijske funkcije promatramo kao prediktivnu distribuciju primjera na ulazu klasifikatora, kod koje bilježimo najveću vrijednost i pripadni razred. Razred kojem pripada najviša softmax vrijednost nam je od interesa za primjere koji pripadaju skupu za učenje kako bi izračunali performanse modela nad skupom za učenje, dok iznos najviše softmax vrijednosti bilježimo za primjere koji ne pripadaju skupu za učenje kako bi ispitali različite pragove detekcije te dobili uvid u odziv modela za nepoznate primjere.

U dijelu 4.1 jednadžbom (4.1) smo uveli jednostavan detektor izvan-distribucijskih primjera zasnovan na pragu i izgrađen nad klasifikatorom, gdje detektor $g(\mathbf{x}) : X \rightarrow$

$\{0, 1\}$ predikciji $\mathbf{x}$ pridružuje oznaku 1, ako je iznos povjerenja $g(\mathbf{x})$ u predikciju iznad nekog praga δ, i oznaku 0 inače. Prirodno iz definicije (4.1) proizlazi kako savršen detektor za sve unutar-distribucijske primjere ostvaruje iznos povjerenja $g(\mathbf{x}) \geq \delta$ pridružujući im pritom oznaku 1, dok svim izvan-distribucijskim primjerima pridružuje oznaku 0. Sljedeći gornju definiciju u duhu binarne klasifikacije u daljnjem razmatranju unutar-distribucijske primjere promatramo kao pozitivnu klasu, dok izvan-distribucijske primjere promatramo kao negativnu klasu.

Za realan detektor želimo odrediti vrijednost praga δ takvu da performanse modela nad pozitivnom klasom slijede one nepromijenjenog klasifikatora, odnosno klasifikatora koji nije učen funkcijom cijene dvostruke pouzdanosti. Takav nepromijenjeni klasifikator promatramo kao osnovicu usporedbe čije performanse detekcije izvan-distribucijskih primjeraka želimo nadjačati, uz prisutan kompromis u vidu postojanja pogrešno-pozitivno i pogrešno-negativno klasificiranih primjera. Robusnu evaluaciju performansi modela ostvarujemo korištenjem metrika koje ne koriste jedan strogo definiran prag detekcije, već analiziraju ponašanje modela za sve vrijednosti praga detekcije. Metriku udjela stvarnih negativa (engl. *true negative rate*) kada udio stvarnih pozitiva (engl. *true positive rate*) doseže iznos N koristimo za uvođenje strogo definiranog praga kako bi usporedili performanse istog modela preko više različitih izvan-distribucijskih skupova te različitih modela pri istom pragu.

Prepoznajemo postojanje problema određivanja optimalne vrijednosti praga povjerenja u točnost predikcije za naučeni klasifikator koji obnaša određenu funkciju, no to nije predmet analize ovog rada.

Površina ispod ROC-krivulje

Mjera površine ispod ROC-krivulje (engl. *AUROC, Area Under the Receiver Operating Characteristic curve*) računa udio površine ispod ROC-krivulje koji prekriva jedinični kvadrat. ROC-krivulja prikazuje međuovisnost udjela pogrešno-detektiranih pozitivnih vrijednosti $FPR = FP/(FP + TN)$ na X osi i točno-detektiranih pozitivnih vrijednosti $TPR = TP/(TP + FN)$ na Y osi za sve moguće vrijednosti praga detekcije. Nadalje, AUROC mjera može biti protumačena kao vjerojatnost da slučajno odabran pozitivan primjer ostvari veći/viši klasifikacijski rezultat od slučajno odabranog negativnog primjera. Klasifikator koji nasumično pridaje klasifikacijske oznake primjerima ostvaruje AUROC mjeru jednaku 50%, dok idealan klasifikator ostvaruje 100%.

Površina ispod PR-krivulje

Mjera površine ispod PR-krivulje (engl. *AUPR, Area Under the Precision-Recall curve*) računa udio površine ispod PR-krivulje koji prekriva jedinični kvadrat. PR-krivulja prikazuje međuovisnost odziva za pozitivan razred $RCL = TP/(TP + FN)$ na X osi, i preciznosti za pozitivan razred $PRC = TP/(TP + FP)$ na Y osi za sve moguće vrijednosti praga detekcije. AUPR mjera je informativnija od AUROC mjere za situacije gdje su populacije koje odgovaraju pozitivnom i negativnom razreda nesrazmjerne. Klasifikator koji nasumično pridaje klasifikacijske oznake primjerima ostvaruje AUPR mjeru približno jednaku preciznosti, dok idealan klasifikator ostvaruje 100%.

Točnost detekcije

Mjera točnosti detekcije (engl. *detection accuracy*) odgovara iznosu najveće vjerojatnosti točne detekcije izvan-distribucijskih primjeraka za neki prag δ:

$$1 \; - \; \min_{\delta} \{P_{in} \; (q(\mathbf{x}) \leq \delta) \; P(\mathbf{x} \text{ je iz } P_{in}) + P_{out} \; (q(\mathbf{x}) > \delta) \; P(\mathbf{x} \text{ je iz } P_{out})\}, \quad (5.1)$$

gdje je $g(\mathbf{x})$ iznos povjerenja detektora u predikciju (poput iznosa najviše vrijednosti prediktivne distribucije). Ovdje pretpostavljamo da se unutar- i izvan- distribucijski primjeri u test setu mogu pojaviti s istom vjerojatnošću, odnosno $P(\mathbf{x} \text{ je iz } P_{in}) = P(\mathbf{x} \text{ je iz } P_{out}) = 0.5$.

Udio stvarnih negativa pri udjelu stvarnih pozitiva jednakom N

Gdje prethodne metrike mjere uspješnost modela kod detekcije izvan-distribucijskih primjeraka za više pragova detekcije, TNRN (engl. *true negative rate at N*) metrika radi procjenu uspješnosti za jedan striktno-definiran prag. Promatrajući uspješnost modela na striktno-definiranom pragu možemo jasno usporediti njegove performanse na različitim izvan-distribucijskim skupovima, kao i njegove performanse s performansa drugih modela za taj isti prag. U kontekstu rada pritom odabiremo prag kod kojeg je udio točno-detektiranih pozitivnih vrijednosti $TPR = TP/(TP + FN)$ jednak 95% za kojeg zatim prijavljujemo udio točno-detektiranih negativnih vrijednosti $TNR = TN/(FP + TN)$. Ovo možemo promatrati kao vjerojatnost točne detekcije izvan-distribucijskog primjerka kod praga koji osigurava da je 95% unutar-distribucijskih primjeraka točno detektirano, što prirodno želimo maksimizirati.

5.3. Arhitekture mreža

Središnji klasifikator

Lee et al. (2017) koriste duboki konvolucijski model VGG-13 (engl. *visual geometry group-13*) (Simonyan i Zisserman, 2014) u kapacitetu središnjeg klasifikatora koji se uči funkcijom cijene dvostruke pouzdanosti. Porodicu arhitektura dubokih modela VGG-* karakterizira korištenje rastućeg broja konvolucijskih slojeva s malim konvolucijskim jezgrama dimenzija 3x3. Različite inačice ovakvih mreža postižu izvrsne rezultate na zadacima lokalizacije i prepoznavanja objekata na slikama iz stvarnog svijeta. Specifikacija mreže VGG-13 je dana tablicom 5.2.

Kod ispitivanja alternativnih pristupa učenju središnjeg klasifikatora opisanih algoritmima 4 i 5, korištena je jednostavnija arhitektura središnjeg klasifikatora za brže prototipiziranje. Specifikacija jednostavnije inačice središnjeg klasifikatora dana je tablicom 5.1, te slijedi ograničene upute o provedenim ispitivanjima alternativnih pristupa učenju pružene od Lee et al. (2017).

Konvolucijski generativni suparnički model

Konvolucijski model generativnih suparničkih mreža korišten u radu idejno slijedi model predložen od strane Radford et al. (2015). Model je prilagođen za generiranje i klasificiranje slika dimenzija Cx32x32, gdje C označava broj kanala slike, a preostale dvije dimenzije određuju njenu visinu i širinu. Specifikacije generatora i diskriminatora dane su tablicama 5.3, odnosno 5.4.

Kod ispitivanja alternativnih pristupa učenju središnjeg klasifikatora opisanih algoritmima 4 i 5, duboki konvolucijski generator odstupa od specifikacije 5.3 u posljednjem sloju, gdje je kao aktivacijska funkcija korišten tangens hiperbolni. Ova specifikacija generativne suparničke mreže slijedi ograničene upute o provedenim ispitivanjima alternativnih pristupa učenju pružene od Lee et al. (2017).

Jednostavni klasifikator

Dimenzije izlaza	Sloj
3 x 32 x 32	Ulazna slika
128 x 28 x 28	Konvolucija: MZ = 128, DJ = 5 x 5, K = 1, korišten pomak
128 x 14 x 14	Max Pool: DJ = 2 x 2, K = 2
256 x 10 x 10	Konvolucija: MZ = 256, DJ = 5 x 5, K = 1, korišten pomak
256 x 5 x 5	Max Pool: DJ = 2 x 2, K = 2
256	Potpuno povezani: NSS = 256
256	ReLU aktivacijska funkcija
128	Potpuno povezani: NSS = 128
128	ReLU aktivacijska funkcija
10	Potpuno povezani: NSS = 10

Tablica 5.1: Specifikacija jednostavnije inačice klasifikatora. MZ = broj izlaznih mapi značajki, DJ = dimenzije konvolucijske jezgre, K= korak, NSS = broj neurona skrivenog sloja.

VGG-13

Dimenzije izlaza	Sloj
3 x 32 x 32	Ulazna slika
64 x 32 x 32	Konvolucija: MZ = 64, DJ = 3 x 3, I = 1
64 x 32 x 32	Konvolucija: MZ = 64, DJ = 3 x 3, I = 1
64 x 16 x 16	Max Pool: DJ = 2 x 2, K = 2
128 x 16 x 16	Konvolucija: MZ = 128, DJ = 3 x 3, I = 1
128 x 16 x 16	Konvolucija: MZ = 128, DJ = 3 x 3, I = 1
128 x 8 x 8	Max Pool: DJ = 2 x 2, K = 2
256 x 8 x 8	Konvolucija: MZ = 256, DJ = 3 x 3, I = 1
256 x 8 x 8	Konvolucija: MZ = 256, DJ = 3 x 3, I = 1
256 x 4 x 4	Max Pool: DJ = 2 x 2, K = 2
512 x 4 x 4	Konvolucija: MZ = 512, DJ = 3 x 3, I = 1
512 x 4 x 4	Konvolucija: MZ = 512, DJ = 3 x 3, I = 1
512 x 2 x 2	Max Pool: DJ = 2 x 2, K = 2
512 x 2 x 2	Konvolucija: MZ = 512, DJ = 3 x 3, I = 1
512 x 2 x 2	Konvolucija: MZ = 512, DJ = 3 x 3, I = 1
512 x 1 x 1	Max Pool: DJ = 2 x 2, K = 2
512	Potpuno povezani: NSS = 512
512	ReLU aktivacijska funkcija
512	Dropout: $p = 0.5$
512	Potpuno povezani: NSS = 512
512	ReLU aktivacijska funkcija
512	Dropout: $p = 0.5$
10	Potpuno povezani: NSS = 10

Tablica 5.2: Specifikacija duboke konvolucijske mreže VGG-13 korištene u kapacitetu središnjeg klasifikatora. MZ = broj izlaznih mapi značajki, DJ = dimenzije konvolucijske jezgre, K = korak, I = ispunjenje, NSS = broj neurona skrivenog sloja.

Generator

Dimenzije izlaza	Sloj
100 x 1 x 1	Vektor šuma z uzorkovan iz $\mathcal{N}(0, 1)$
512 x 4 x 4	Dekonvolucija: MZ = 512, DJ = 4 x 4, K = 1
512 x 4 x 4	Normalizacija po grupama
512 x 4 x 4	ReLU aktivacijska funkcija
256 x 8 x 8	Dekonvolucija: MZ = 256, DJ = 4 x 4, K = 2, I = 1
256 x 8 x 8	Normalizacija po grupama
256 x 8 x 8	ReLU aktivacijska funkcija
128 x 16 x 16	Dekonvolucija: MZ = 128, DJ = 4 x 4, K = 2, I = 1
128 x 16 x 16	Normalizacija po grupama
128 x 16 x 16	ReLU aktivacijska funkcija
3 x 32 x 32	Dekonvolucija: MZ = 3, DJ = 4 x 4, K = 2, I = 1
3 x 32 x 32	Sigmoidalna aktivacijska funkcija

Tablica 5.3: Specifikacija dubokog konvolucijskog generatora generativnog suparničkog modela. MZ = broj izlaznih mapi značajki, DJ = dimenzije konvolucijske jezgre, K = korak, I = ispunjenje.

Diskriminator

Dimenzije izlaza	Sloj
3 x 32 x 32	Ulazna slika
128 x 16 x 16	Konovlucija: MZ = 128, DJ = 4 x 4, K = 2, I = 1
128 x 16 x 16	Leaky ReLU aktivacijska funkcija: $\alpha = 0.2$
256 x 8 x 8	Konovlucija: MZ = 256, DJ = 4 x 4, K = 2, I = 1
256 x 8 x 8	Normalizacija po grupama
256 x 8 x 8	Leaky ReLU aktivacijska funkcija: $\alpha = 0.2$
512 x 4 x 4	Konovlucija: MZ = 512, DJ = 4 x 4, K = 2, I = 1
512 x 4 x 4	Normalizacija po grupama
512 x 4 x 4	Leaky ReLU aktivacijska funkcija: $\alpha = 0.2$
1 x 1 x 1	Konovlucija: MZ = 1, DJ = 4 x 4, K = 1
1	Sigmoidalna aktivacijska funkcija

Tablica 5.4: Specifikacija dubokog konvolucijskog diskriminatora generativnog suparničkog modela. MZ = broj izlaznih mapi značajki, DJ = dimenzije konvolucijske jezgre, K = korak, I = ispunjenje.

5.4. Rezultati

U nastavku, sve vrijednosti rezultata eksperimenata objavljene u tabličnom obliku se odnose na postotke.

5.4.1. Združeno učenje središnjeg klasifikatora i generativnog suparničkog modela

CIFAR-10 kao skup podataka za učenje

Prvotni pokušaji reprodukcije rezultata iz Lee et al. (2017) korištenjem algoritma združenog učenja (3) nad CIFAR-10 skupom kao unutar-distribucijskim koristeći izvorni kod autora[1] su konzistentno rezultirali modelima koji su nakon evaluacije na izvan-distribucijskim skupovima performansama signifikantno odstupali od prijavljenih najboljih rezultata. Odstupanja rezultata nisu bila konzistentna, pri čemu bi ponovljeni postupci učenja ponekad rezultirali modelima sa signifikantno boljim, odnosno lošijim performansama. Iz ovog razloga pri objavi rezultata eksperimenata navodimo rezultate kroz sažete mjere performansi koje predstavljaju srednju vrijednost i standardnu devijaciju mjera performansi preko 5 zasebnih sjednica učenja. Arhitekture korištenih modela opisane su u 5.3, korištena veličina mini-grupe u svim eksperimentima je 64, modeli su trenirani 100 epoha, te je doprinos komponente Kullback-Leiblerove divergencije funkcije dvostruke pouzdanosti 0.1. Osnovica usporedbe (engl. *baseline*) je klasifikator treniran funkcijom cijene dvostruke pouzdanosti kojem su predstavljeni izvan-distribucijski primjeri uzorkovani iz standardne normalne distribucije, dok su ostali uvjeti učenja nepromijenjeni. Osnovica usporedbe našeg rada nadilazi osnovicu usporedbe Lee et al. (2017), gdje je razmatran klasifikator koji minimizira funkciju gubitka unakrsne entropije, bez korištenja izvan-distribucijskih primjeraka u fazi učenja. Smatramo kako izmijenjena osnovica usporedbe pruža bolji kontekst rezultatima u radu, s obzirom na to da opisuje jednostavnu ideju za detekciju izvan-distribucijskih primjeraka s kompetitivnim rezultatima. Performanse osnovice usporedbe navodimo u dodatku A.

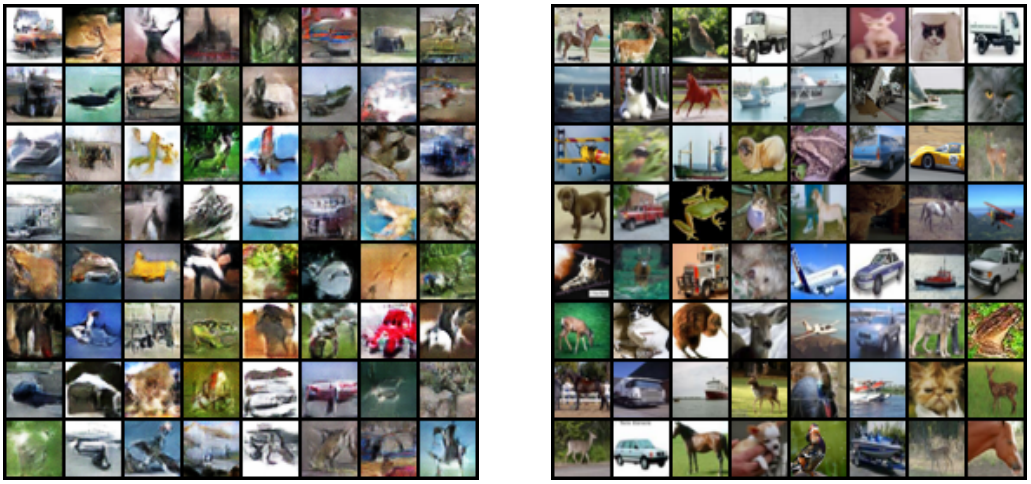
Unatoč prisutnoj varijabilnosti u performansama, naučeni klasifikatori su u većini sjednica učenja prema svim metrikama bolji od novo-predstavljene osnovice usporedbe. Najveći napredak u odnosu na osnovicu usporedbe se postiže prema metrici TNR@TPR95%, koja razmatra korištenje strogog klasifikatora, gdje su rezultati konzistentno bolji od osnovice za više od 100%.

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

Točnost središnjeg klasifikatora na test setu podataka za učenje (5.5) ne varira značajno kroz ponovljene eksperimente pod novom funkcijom cilja (4.3) te kroz epohe učenja blisko slijedi točnost koju ta ista arhitektura središnjeg klasifikatora ostvaruje pod nepromijenjenom funkcijom cilja - funkcijom gubitka unakrsne entropije. Ovo potvrđuje sposobnost središnjeg klasifikatora da se stabilno i uspješno specijalizira nad skupom podataka za učenje s novom funkcijom cilja, gdje se KL komponenta ponaša kao regularizacijski član. Nadalje, ovo potvrđuje zaključke prijavljene od Lee et al. (2017) kako združeno učenje klasifikatora i generativne suparničke mreže ne uzrokuje degradaciju performansi klasifikatora na skupu za učenje.

Značajnu varijabilnost performansi klasifikatora na izvan-distribucijskim skupovima kroz epohe učenja (5.6, 5.7, 5.8) povezujemo s KL komponentom funkcije cijene dvostruke pouzdanosti klasifikatora, odnosno s nedostatcima generativnih suparničkih mreža - nestabilnim postupkom učenja i netraktabilnim latentim prostorom primjera. Generirani primjeri se kroz ponovljene eksperimente značajno razlikuju, jer generator kroz ponovljene eksperimente s različitim uspjehom uspijeva opisati rubne primjere domene za učenje sukladno cilju 4.4 te u svakoj iteraciji učenja nasumično uzorkuje primjere koji, kako je opisano u algoritmu 3, usmjeravaju klasifikator.

Iz tablica 5.6, 5.7, 5.8 uočavamo trend spore degradacije performansi klasifikatora kroz epohe učenja na skupovima SVHN, Imagenet i LSUN respektivno. Degradacija performansi ukazuje na to da generator izvan-distribucijskih primjeraka najviše doprinosi generalizacijskog sposobnosti klasifikatora u ranoj fazi učenja. Slika 5.7 prikazuje primjere uzorkovane uporabom generatora za jedan vrlo uspješan klasifikator evaluiran u 20. epohi faze učenja u usporedbi sa slučajno uzorkovanim primjerima iz skupa CIFAR-10. Primjećujemo kako generator proizvodi visoko-degradirane, krnje primjere koji blago podsjećaju na motive iz skupa za učenje. Pretpostavljamo kako generator u ranim epohama s visoko-degradiranim primjerima najviše doprinosi generalizacijskoj sposobnosti klasifikatora. Dokumentirano je kako učenje suparničkog para zahtjeva mnogo epoha za uspješno usvajanje domene za učenje (Goodfellow, 2016) te pretpostavljamo da se kroz nastavak učenja kvaliteta generiranih primjera približava onoj slika iz skupa za učenje. Klasifikator predstavljen primjerima generatora iz kasnijih epoha kao izvan-distribucijskim polagano gubi dobre generalizacijske sposobnosti, jer one sve više nalikuju skupu za učenje. Ovu pretpostavku ne možemo analitički ispitati. Postoje pokušaji usporedbe kvalitete generiranih primjera kroz epohe učenja s primjerima iz domene za učenje (Barratt i Sharma, 2018), no u ovom slučaju takve usporedbe nisu moguće zbog izmijenjene funkcije cilja generativnog suparničkog para (4.4). Za generirane primjere uzorkovane generatorom koji podliježe ovakvoj funkciji



Slika 5.7: Lijevo: primjeri uzorkovani uporabom generatora za vrlo uspješan klasifikator evaluiran u 20. epohi faze učenja. Desno: slučajno uzorkovane slike iz CIFAR-10 skupa. Generator u ranim epohama procesa učenja generira primjere koji preuzimaju motive poput pozicije subjekta na slici, palete korištenih boja i kontrasta iz početnog skupa za učenje. Pretpostavljamo kako navedeni visoko-degradirani primjeri najviše doprinose generalizacijskoj sposobnosti središnjeg klasifikatora.

cilja ne postoji temeljna istina s kojom možemo izvršiti usporedbu u smislu kvalitete reprodukcije, odnosno ustanoviti da li je zahvaćen rijedak rubni prostor distribucije podataka za učenje. Pretpostavku ispitujemo vizualizacijom primjera koje generator proizvodi za fiksan vektor šuma z na ulazu generatora. Slika 5.9 prikazuje primjere generirane uporabom generatora iz nastavka rada gdje se skup SVHN koristi za učenje središnjeg klasifikatora. Primjeri su generirani uporabom fiksnog vektor šuma na kraju 10, 20, 30, 40, 50 i 60 epohe učenja. Primjećujemo kako generator sukladno pretpostavci u ranim epohama učenja proizvodi iznimno degradirane primjere koji snažno podsjećaju na motive iz skupa za učenje. Nadalje, kvaliteta generiranih primjera postepeno raste kroz epohe što vidimo kroz porast kontrasta i povećanje glatkoće rubova znamenki, iako primjeri često zadržavaju karakteristična izobličenja poput nejednake ispunjenosti pozadine.

Na slici 5.8 za klasifikator iz slike 5.7 prikazujemo raspodjelu najviših vrijednosti prediktivnih distribucija za sve primjere iz unutar- i izvan-distribucijskih skupova. Crvena isprekidana linija na grafovima aludira na klasifikator s idealnim detekcijskim sposobnostima koji za svaki primjer iz izvan-distribucijskog skupa na izlazu softmax sloja daje idealnu uniformnu distribuciju. Primjećujemo teže 'repove' grafova izvan-distribucijskih skupova koji govore o većoj nesigurnosti klasifikatora u predikciju na tim primjerima. Također, vidljivo je da klasifikator najviše griješi na SVHN skupu kod

kojeg za skoro polovinu primjera daje predikciju s vjerojatnošću ≥ 0.9.

Svi raniji rezultati su dobiveni postavljanjem udjela doprinosa $\beta = 0.1$ komponente funkcije cijene dvostruke pouzdanosti koja odgovara Kullback-Leiblerovoj divergenciji. U pokušaju stabilizacije postupka učenja ponovili smo postupak validacije parametra β za niže $\{0.05, 0.06, 0.07, 0.08, 0.09\}$, odnosno više vrijednosti $\{0.2, 0.3, 0.4, 0.5\}$. Ostale postavke učenja van parametra β su ostale nepromijenjene. Za svaku od vrijednosti hiperparametra učenje je ponovljeno 3 puta te su detaljni rezultati dostupni na online repozitoriju projekta[2].

Pri ispitivanju vrijednosti $\beta \leq 0.07$ klasifikator performansama degradira na test setu skupa za učenje, zbog čega sjednice s navedenim vrijednostima nismo dalje razmatrali. Ponašanje modela s ostalim razmatranim vrijednostima hiperparametra β na test setu skupa za učenje nije značajno odudaralo od onog zabilježenog s $\beta = 0.1$. U svim eksperimentima su se najkompetitivniji modeli javljali do 50. epohe učenja, nakon čega primjećujemo spor trend opadanja performansi na izvan-distribucijskim skupovima. Ovo slijedi trendove zabilježene pri $\beta = 0.1$, a koje povezujemo s kvalitetom generiranih izvan-distribucijskih primjera. Varijacije među performansama modela kod svih metrika su se značajno smanjile kako smo se približavali nižim ($\beta = 0.07$) odnosno višim ($\beta = 0.5$) vrijednostima hiperparametra.

U eksperimentima je vidljiv trend zabilježen u Lee et al. (2017), gdje klasifikator treniran nad CIFAR-10 skupom najviše problema s detekcijom izvan-distribucijskih primjeraka ima kod skupa SVHN, koji se slikama brojeva značajno razlikuje od ostalih skupova s prirodnim slikama. Za alternativne vrijednosti iznosa doprinosa β, naučeni klasifikatori performansama na izvan-distribucijskom skupu SVHN nisu konzistentno prelazili osnovicu usporedbe.

Performanse modela na skupovima Imagenet i LSUN za strogi klasifikator pretpostavljen mjerom TNR@TPR95% su lošije od polaznog modela s $\beta = 0.1$, dok su vrijednosti ostalih metrika iznad osnovice usporedbe i kompetitivne s navedenim modelom, s manjim devijacijama kroz ponovljene sjednice. Izdvajamo manje vrijednosti parametra $0.7 < \beta \leq 1$ koje su postizale rezultate usporedive s polaznim najboljim modelima za $\beta = 0.1$ na Imagenet i LSUN skupovima pri AUROC i Detection Accuracy mjerama.

Konačno, preporučujemo korištenje iznosa parametra $0.7 < \beta \leq 1$ popraćenog dodatnim validacijskim postupkom u budućim istraživanjima koja uključuju rad s prirodnim skupovima slika.

[2] `https://github.com/hrvojebusic`

Epoha	Test Set Accuracy
10	79.46 ± 0.87
20	80.15 ± 0.86
30	80.59 ± 0.42
40	80.81 ± 0.37
50	81.39 ± 0.56
60	81.54 ± 0.2
70	82.89 ± 0.33
80	82.52 ± 0.24
90	82.39 ± 0.33
100	82.07 ± 0.36

Tablica 5.5: Srednja vrijednost performansi središnjeg klasifikatora preko 5 sjednica učenja treniranog nad CIFAR-10 skupom.

Epoha	TNR at TPR 95%	AUROC	Det Acc	AUPR In
10	7.47 ± 5.03	74.73 ± 3.67	73.55 ± 1.09	81.47 ± 1.92
20	12.13 ± 5.5	76.73 ± 3.26	73.02 ± 1.62	81.25 ± 1.83
30	14.94 ± 2.09	79.7 ± 2.27	75.42 ± 2.01	83.67 ± 2.19
40	14.53 ± 7.13	79.29 ± 3.6	75.12 ± 2.56	82.04 ± 4.13
50	13.2 ± 4.57	78.37 ± 2.45	74.23 ± 1.63	80.38 ± 4.22
60	14.21 ± 3.16	80.13 ± 2.42	75.87 ± 2.1	81.09 ± 3.55
70	14.07 ± 1.75	82.66 ± 1.03	73.38 ± 2.84	70.48 ± 3.78
80	18.37 ± 3.62	84.55 ± 1.15	73.99 ± 3.5	71.48 ± 5.21
90	13.82 ± 1.95	74.71 ± 18.51	69.56 ± 4.53	65.36 ± 4.4
100	15.6 ± 5.45	84.1 ± 1.95	70.21 ± 7.1	66.72 ± 6.75

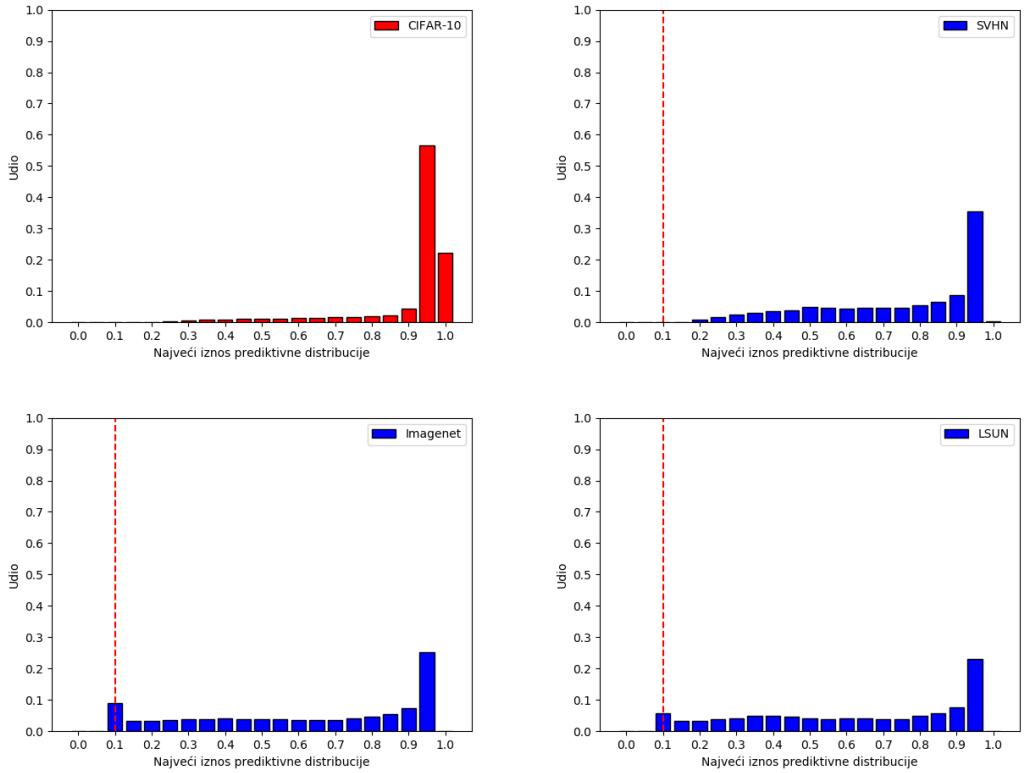
Tablica 5.6: Srednja vrijednost performansi središnjeg klasifikatora preko 5 sjednica učenja treniranog nad CIFAR-10 skupom i evaluiranog nad SVHN izvan-distribucijskim skupom.

Epoha	TNR at TPR 95%	AUROC	Det Acc	AUPR
10	48.59 ± 21.22	84.32 ± 6.66	76.85 ± 8.0	83.94 ± 6.86
20	32.43 ± 13.19	81.84 ± 5.25	74.1 ± 4.28	82.04 ± 5.95
30	23.38 ± 8.46	78.73 ± 3.55	72.69 ± 2.41	79.7 ± 3.72
40	33.22 ± 25.8	81.82 ± 7.37	75.83 ± 7.06	81.93 ± 7.42
50	22.15 ± 6.9	78.6 ± 3.1	72.02 ± 2.67	77.03 ± 4.74
60	18.71 ± 1.93	79.63 ± 1.19	72.77 ± 1.14	76.32 ± 2.31
70	19.0 ± 3.21	80.33 ± 5.93	72.51 ± 1.75	69.69 ± 2.32
80	23.6 ± 6.19	85.26 ± 1.62	73.49 ± 2.45	70.98 ± 4.16
90	26.53 ± 7.41	87.05 ± 1.1	74.65 ± 2.01	71.65 ± 4.13
100	30.92 ± 7.58	87.61 ± 0.82	75.79 ± 1.77	73.38 ± 3.71

Tablica 5.7: Srednja vrijednost performansi središnjeg klasifikatora preko 5 sjednica učenja treniranog nad CIFAR-10 skupom i evaluiranog nad Imagenet izvan-distribucijskim skupom.

Epoha	TNR at TPR 95%	AUROC	Det Acc	AUPR
10	51.81 ± 20.04	86.84 ± 5.51	78.88 ± 7.22	87.19 ± 5.28
20	31.39 ± 14.0	82.22 ± 5.97	74.89 ± 4.85	83.23 ± 6.58
30	26.68 ± 10.23	82.15 ± 3.1	75.41 ± 2.11	83.68 ± 2.94
40	34.48 ± 24.41	83.53 ± 6.58	77.07 ± 6.21	83.9 ± 6.62
50	25.08 ± 7.88	81.9 ± 2.82	74.94 ± 2.01	81.57 ± 2.75
60	20.22 ± 2.41	81.02 ± 1.31	74.42 ± 1.22	79.05 ± 2.08
70	18.95 ± 3.68	81.3 ± 4.92	74.11 ± 2.07	71.77 ± 2.94
80	23.42 ± 5.73	85.6 ± 1.66	74.74 ± 2.17	72.57 ± 4.01
90	26.25 ± 7.17	82.81 ± 8.1	75.61 ± 1.88	72.79 ± 3.86
100	31.42 ± 7.07	87.98 ± 0.95	76.99 ± 1.69	74.99 ± 3.54

Tablica 5.8: Srednja vrijednost performansi središnjeg klasifikatora preko 5 sjednica učenja treniranog nad CIFAR-10 skupom i evaluiranog nad LSUN izvan-distribucijskim skupom.



Slika 5.8: Raspodjela udjela najviših vrijednosti prediktivnih distribucija svih primjera iz unutar- i izvan-distribucijskih skupova za vrlo uspješan klasifikator evaluiran u 20. epohi učenja.

SVHN kao skup podataka za učenje

Također smo proveli eksperimente s ciljem reprodukcije prijavljenih rezultata klasifikatora treniranog nad SVHN skupom te smo ih podvrgli ispitivanju kroz ponovljene sjednice učenja kako bi identificirali moguća odstupanja u performansama između ponovljenih sjednica. Rezultate objavljujemo kroz sažete mjere performansi koje predstavljaju srednju vrijednost i standardnu devijaciju mjera performansi preko 5 sjednica učenja, te primjere slika uzorkovanih generatorom. Arhitekture korištenih modela slijede 5.3, korištena veličina mini-grupe u svim eksperimentima je 128, modeli su trenirani 100 epoha, te je doprinos komponente Kullback-Leiblerove divergencije funkcije dvostruke pouzdanosti 1. Osnovica usporedbe je klasifikator treniran funkcijom cijene dvostruke pouzdanosti kojem su predstavljeni izvan-distribucijski primjeri uzorkovani iz standardne normalne distribucije, dok su ostali uvjeti učenja nepromijenjeni. Performanse osnovice usporedbe navodimo u dodatku B. Naučeni klasifikatori su konzistentno i prema svim metrikama bolji od osnovice usporedbe.

Rezultati slijede one Lee et al. (2017) kod kojih naučeni klasifikator s dvostrukom pouzdanošću dostiže vrlo visoke performanse detekcije izvan-distribucijskih primjeraka u slučaju CIFAR-10 skupa (5.10), i gotovo savršene rezultate za Imagenet i LSUN skupove (5.7, 5.12). Nadalje, naučeni klasifikator zadržava visoku točnost klasifikacije na test setu unutar-distribucijskog skupa (5.9). Odlične detekcijske sposobnosti modela pridjeljujemo signifikantnoj razlici između SVHN skupa koji sadrži slike brojeva i skupova prirodnih slika koje koristimo u izvan-distribucijskom kapacitetu, odnosno pretpostavljamo da je model potpomognut prirodnom odijeljenošću razmatranih distribucija. Slika 5.9 prikazuje primjere generirane uporabom fiksnog vektora šuma z na kraju 10, 20, 30, 40, 50 i 60 epohe učenja. Primjećujemo napredak u kvaliteti slika kroz epohe učenja, koje unatoč izmjeni funkcije cilja generatora s vremenom snažno nalikuju onima iz SVHN skupa (5.3). Ovo ide u prilog ranijoj pretpostavci kako generator u ranim epohama učenja proizvodi visoko-degradirane, krnje primjere koji podsjećaju na motive iz skupa za učenje.

Epoha	Test Set Accuracy
10	93.2 ± 0.35
20	93.6 ± 0.09
30	93.61 ± 0.18
40	93.64 ± 0.4
50	93.6 ± 0.46
60	93.61 ± 0.15
70	94.25 ± 0.12
80	94.17 ± 0.16
90	94.24 ± 0.12
100	94.1 ± 0.24

Tablica 5.9: Srednja vrijednost performansi središnjeg klasifikatora preko 5 sjednica učenja treniranog nad SVHN skupom.

Epoha	TNR at TPR 95%	AUROC	Det Acc	AUPR
10	92.64 ± 3.91	98.89 ± 0.52	94.58 ± 1.15	98.97 ± 0.42
20	86.02 ± 5.05	97.99 ± 0.66	92.84 ± 0.83	98.2 ± 0.49
30	86.88 ± 4.16	98.1 ± 0.55	92.98 ± 0.8	98.24 ± 0.41
40	79.18 ± 6.22	97.06 ± 0.8	91.63 ± 1.04	97.32 ± 0.69
50	66.12 ± 3.37	95.21 ± 0.59	89.43 ± 1.02	95.86 ± 0.6
60	72.47 ± 7.37	96.25 ± 0.99	90.74 ± 1.2	96.59 ± 1.04
70	73.3 ± 9.5	96.49 ± 1.22	90.96 ± 1.69	95.28 ± 2.01
80	74.2 ± 10.69	96.16 ± 1.49	90.72 ± 2.38	92.97 ± 4.2
90	74.52 ± 7.5	94.45 ± 5.66	90.77 ± 1.99	91.31 ± 3.91
100	75.57 ± 6.65	95.09 ± 3.43	90.58 ± 1.57	91.66 ± 3.14

Tablica 5.10: Srednja vrijednost performansi središnjeg klasifikatora preko 5 sjednica učenja treniranog nad SVHN skupom i evaluiranog nad CIFAR-10 izvan-distribucijskim skupom.

Epoha	TNR at TPR 95%	AUROC	Det Acc	AUPR
10	99.78 ± 0.12	99.96 ± 0.02	99.38 ± 0.35	99.96 ± 0.02
20	99.78 ± 0.19	99.97 ± 0.03	99.46 ± 0.3	99.97 ± 0.03
30	99.88 ± 0.06	99.98 ± 0.01	99.67 ± 0.1	99.98 ± 0.01
40	99.77 ± 0.11	99.96 ± 0.02	99.51 ± 0.29	99.96 ± 0.02
50	99.15 ± 0.82	99.87 ± 0.13	98.65 ± 1.08	99.86 ± 0.13
60	99.46 ± 0.37	99.91 ± 0.06	98.98 ± 0.65	99.91 ± 0.06
70	99.49 ± 0.41	99.89 ± 0.11	98.99 ± 0.72	99.86 ± 0.09
80	99.57 ± 0.24	99.85 ± 0.12	99.07 ± 0.57	99.84 ± 0.11
90	99.63 ± 0.13	99.89 ± 0.03	99.16 ± 0.34	99.84 ± 0.08
100	99.62 ± 0.12	99.89 ± 0.05	99.06 ± 0.4	99.84 ± 0.08

Tablica 5.11: Srednja vrijednost performansi središnjeg klasifikatora preko 5 sjednica učenja treniranog nad SVHN skupom i evaluiranog nad Imagenet izvan-distribucijskim skupom.

Epoha	TNR at TPR 95%	AUROC	Det Acc	AUPR
10	99.93 ± 0.06	99.99 ± 0.01	99.73 ± 0.21	99.99 ± 0.01
20	99.94 ± 0.08	99.99 ± 0.01	99.81 ± 0.17	99.99 ± 0.01
30	99.98 ± 0.01	100.0 ± 0.0	99.93 ± 0.03	100.0 ± 0.0
40	99.97 ± 0.03	99.99 ± 0.0	99.83 ± 0.14	99.99 ± 0.0
50	99.51 ± 0.69	99.92 ± 0.11	99.13 ± 1.01	99.92 ± 0.11
60	99.78 ± 0.19	99.96 ± 0.04	99.37 ± 0.54	99.95 ± 0.05
70	99.8 ± 0.24	99.93 ± 0.09	99.39 ± 0.57	99.94 ± 0.06
80	99.8 ± 0.21	99.92 ± 0.07	99.44 ± 0.44	99.94 ± 0.06
90	99.88 ± 0.1	99.93 ± 0.04	99.57 ± 0.27	99.95 ± 0.05
100	99.89 ± 0.07	99.9 ± 0.05	99.47 ± 0.28	99.95 ± 0.03

Tablica 5.12: Srednja vrijednost performansi središnjeg klasifikatora preko 5 sjednica učenja treniranog nad SVHN skupom i evaluiranog nad LSUN izvan-distribucijskim skupom.

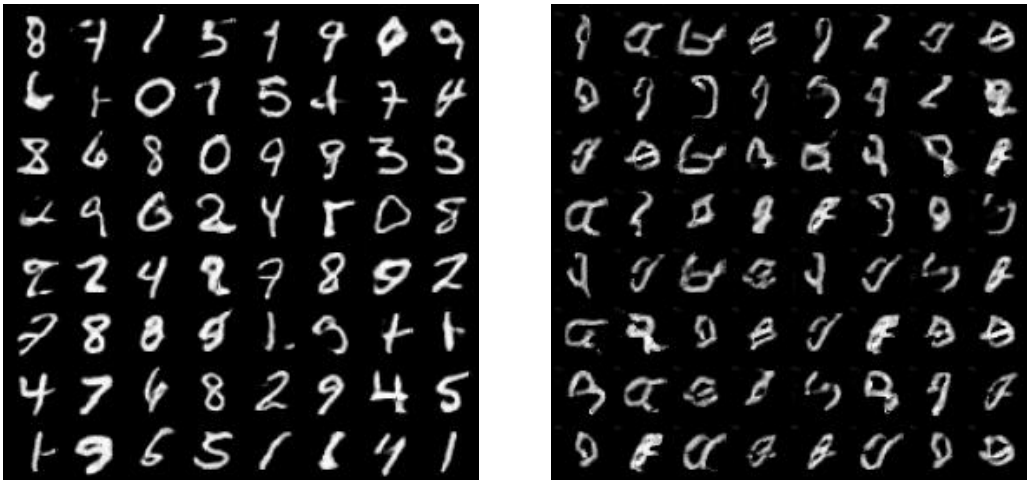
Slika 5.9: Primjeri generirani uporabom fiksnog vektora šuma z i generatora učenog funkcijom cilja dvostruke pouzdanosti nad SVHN skupom. S lijeva na desno, od vrha prema dnu, epohe u kojima su primjeri generirani: 10, 20, 30, 40, 50 i 60.

5.4.2. Alternativni pristupi treniranju

Lee et al. (2017) aludiraju na moguće alternativne pristupe učenju klasifikatora i suparničkog para mreža, koji su među ostalom korišteni pri validaciji ideje da generator suparničkog para proizvodi primjerke na granici unutar-distribucijskog skupa (5.10).

Algoritam 5 i 4 smo prvo ispitali korištenjem jednostavnog klasifikatora (5.1) i nepromijenjenog dubokog konvolucijskog suparničkog modela (5.3, 5.4) nad MNIST kao unutar-, i Fashion-MNIST kao izvan-distribucijskim skupom. Osnovicu usporedbe je činio jednostavni klasifikator predstavljen s primjerima iz standardne normalne distribucije kao izvan-distribucijskim. Na ovom jednostavnom zadatku je algoritam 5 postizao gotovo savršene rezultate prema svim metrikama, nadilazeći osnovicu usporedbe, dok je algoritam 4 bio lošiji od osnovice usporedbe. U nastavku smo stoga razmatrali samo algoritam 5.

Eksperimente smo nastavili trenirajući modele nad CIFAR-10 skupom kao unutar-distribucijskim. Središnji klasifikator dobiven u trećem koraku sheme 5 koji prati arhitekturu jednostavnog klasifikatora je performansama na izvan-distribucijskim skupovima Imagenet i LSUN bio lošiji od osnovice usporedbe, dok je na SVHN skupu prema svim metrikama bio bolji od osnovice usporedbe te u slučaju strogog klasifikatora pretpostavljenog metrikom TNR@TPR95% više nego dvostruko bolji. Ovaj trend je vrlo zanimljiv u kontekstu loših performansi koje je središnji klasifikator opisan VGG-13 arhitekturom učen združenom shemom učenja imao pri detekciji izvan-distribucijskih primjeraka iz SVHN skupa. Motivirani rezultatima smo ponovili eksperimente sukladno algoritmu 5 s arhitekturom klasifikatora koja slijedi 5.2 kako bi vidjeli hoće li zahtjevnija inačica klasifikatora rezultirati sličnim trendovima na izvan-distribucijskim skupovima. Ponovljeni eksperimenti su rezultirali središnjim klasifikatorom koji je na skupu SVHN lošiji od osnovice usporedbe, dok je na skupovima CIFAR-10 i LSUN bolji od osnovice usporedbe, no lošiji od najboljih modela dobivenih združenim algoritmom učenja.



Slika 5.10: Lijevo: primjeri uzorkovani generatorom suparničkog para mreža treniranog nad MNIST skupom. Desno: primjeri uzorkovani generatorom suparničkog para mreža pod izmijenjenim uvjetima učenja. Generator je sparen s naučenim jednostavnim klasifikatorom 5.1 s dvostrukom pouzdanošću treniranim nad šumom iz standardne normalne distribucije. Primjeri generatora trebaju imati prediktivnu distribuciju sličnu uniformnoj za klasifikator u pitanju. Preuzeto iz Lee et al. (2017).

6. Zaključak

Duboki konvolucijski diskriminativni modeli ostvaruje izvrsne rezultate na klasifikacijskim zadacima u području računalnog vida, no ti isti modeli ne pružaju uvid u pouzdanost predikcije koju nude. Ovo rezultira modelima koji su skloni neopravdanom optimizmu, što znači da primjerci izvan domene ekspertize modela često bivaju neispravno klasificirani s velikom pouzdanošću. Nemogućnost modela da naprave procjenu pouzdanosti vlastite predikcije predstavlja ozbiljan problem njihovom usvajanju i primjeni u stvarnom svijetu.

Razmotrili smo pristup detekciji izvan-distribucijskih primjeraka predložen od Lee et al. (2017) koji se fokusira na bolje odjeljivanje prediktivnih distribucija unutar- i izvan-distribucijskih primjeraka, s ciljem izgradnje izvan-distribucijskog detektora zasnovanog na pragu koji najviši iznos prediktivne distribucije interpretira kao pouzdanost modela u predikciju primjera. Funkcija cilja klasifikatora je proširena komponentom koja prediktivnu distribuciju izvan-distribucijskih primjeraka pokušava približiti uniformnoj. Izvan-distribucijski primjeri u fazi učenja potječu od generativnog suparničkog modela koji se uči u tandemu s klasifikatorom i služi generiranju primjera koji se nalaze na granici distribucije skupa za učenje te se interpretiraju kao najkorisniji izvan-distribucijski primjeri za regularizaciju središnjeg klasifikatora.

Eksperimenti su pokazali kako razmatran pristup nadilazi performanse osnovice usporedbe originalnog rada - klasifikator treniran funkcijom cijene unakrsne entropije, i kompetitivniju osnovicu uvedenu u ovom radu - klasifikator treniran novo-uvedenom funkcijom cijene dvostruke pouzdanosti s izvan-distribucijskim primjerima uzorkovanim iz standardne normalne distribucije. Za modele trenirane nad skupom prirodnih slika se performanse na izvan-distribucijskim skupovima značajno razlikuju kroz ponovljene sjednice učenja, te dok možemo prijaviti poboljšanje u usporedbi s osnovicom usporedbe, nivo rezultata prijavljen u originalnom radu ne možemo konzistentno reproducirati. Prethodni modeli najstabilniji napredak u odnosu na osnovicu usporedbe ostvaruju kada se njihove sposobnosti detekcije izvan-distribucijskih primjeraka evaluiraju nad drugim razmatranim skupovima prirodnih slika, dok najlošije performanse

ostvaruju pri evaluaciji na skupu slika brojeva iz stvarnog svijeta. Volatilnost rezultata pridjeljujemo dokumentiranoj nestabilnosti postupka učenja generativnih suparničkih modela i netraktabilnosti distribucije koju konačni generator opisuje.

U budućem radu predlažemo istraživanje stabilnijeg načina uzorkovanja izvan-distribucijskih primjeraka uparenog s predloženom funkcijom cijene dvostruke pouzdanosti i istraživanje performansi istog na izvan-distribucijskim skupovima koji se u svojim domenama signifikantno razlikuju. Prepoznajemo potencijal u mogućnosti uzorkovanja izvan-distribucijskih primjeraka najrelevantnijih za skup faze učenje, kojom se eliminira potreba za i ovisnost o kvaliteti eksplicitnog izvan-distribucijskog skupa.

LITERATURA

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, i Dan Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

Shane Barratt i Rishi Sharma. A note on the inception score. *arXiv preprint arXiv:1801.01973*, 2018.

Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, i Noemie Elhadad. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. U *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, stranice 1721–1730. ACM, 2015.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, i Li Fei-Fei. Imagenet: A large-scale hierarchical image database. U *2009 IEEE conference on computer vision and pattern recognition*, stranice 248–255. Ieee, 2009.

Ross Girshick. Fast r-cnn. U *Proceedings of the IEEE international conference on computer vision*, stranice 1440–1448, 2015.

Ian Goodfellow. Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*, 2016.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, i Yoshua Bengio. Generative adversarial nets. U *Advances in neural information processing systems*, stranice 2672–2680, 2014a.

Ian Goodfellow, Yoshua Bengio, i Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

Ian J Goodfellow, Jonathon Shlens, i Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014b.

Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*, 2014.

Dan Hendrycks i Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

Sergey Ioffe i Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Diederik P Kingma i Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Alex Krizhevsky, Vinod Nair, i Geoffrey Hinton. The cifar-10 dataset. *online: http://www. cs. toronto. edu/kriz/cifar. html*, 55, 2014.

Yann LeCun, Corinna Cortes, i CJ Burges. Mnist handwritten digit database. *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, 2:18, 2010.

Kimin Lee, Honglak Lee, Kibok Lee, i Jinwoo Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. *arXiv preprint arXiv:1711.09325*, 2017.

Vinod Nair i Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. U *Proceedings of the 27th international conference on machine learning (ICML-10)*, stranice 807–814, 2010.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, i Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

Alec Radford, Luke Metz, i Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Karen Simonyan i Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Vedran Vukotić. Raspoznavanje objekata dubokim neuronskim mrežama. Magistarski rad, Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva, 2014.

Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, i Rob Fergus. Regularization of neural networks using dropconnect. U *International conference on machine learning*, stranice 1058–1066, 2013.

Han Xiao, Kashif Rasul, i Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Bing Xu, Naiyan Wang, Tianqi Chen, i Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015.

Raymond A Yeh, Chen Chen, Teck Yian Lim, Alexander G Schwing, Mark Hasegawa-Johnson, i Minh N Do. Semantic image inpainting with deep generative models. U *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, stranice 5485–5493, 2017.

Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, i Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

Filip Zelić. Izlučivanje slikovnih reprezentacija generativnim suparničkim modelima. Magistarski rad, Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva, 2018.

Dodatak A
CIFAR-10 osnovica usporedbe

Epoha	Test Set Accuracy
10	76.96
20	78.85
30	77.65
40	79.81
50	79.29
60	78.49
70	80.61
80	79.98
90	79.83
100	80.95

Tablica A.1: Performanse klasifikatora treniranog funkcijom cijene dvostruke pouzdanosti nad skupom CIFAR-10 i primjerima iz standardne normalne distribucije na test setu.

Epoha	TNR at TPR 95%	AUROC	Det Acc	AUPR
10	21.51	79.85	73.07	83.47
20	16.75	80.3	75.87	85.1
30	10.73	76.59	73.72	81.45
40	12.66	78.19	74.02	79.2
50	10.84	78.1	74.23	81.12
60	15.7	80.15	74.42	78.64
70	15.64	80.22	74.14	78.99
80	8.15	73.59	67.58	66.41
90	12.53	79.32	74.87	79.87
100	12.71	80.21	75.65	77.2

Tablica A.2: Performanse klasifikatora treniranog funkcijom cijene dvostruke pouzdanosti nad skupom CIFAR-10 i primjerima iz standardne normalne distribucije na SVHN izvan-distribucijskom skupu.

Epoha	TNR at TPR 95%	AUROC	Det Acc	AUPR
10	17.29	72.52	66.44	74.0
20	17.94	77.46	71.34	77.58
30	14.81	75.69	69.26	73.59
40	13.91	76.61	69.52	71.11
50	11.55	73.95	66.76	68.14
60	10.6	73.02	63.58	62.11
70	15.95	77.7	70.17	71.86
80	16.72	79.04	71.67	73.66
90	16.6	77.93	70.32	72.23
100	14.71	78.84	69.54	68.43

Tablica A.3: Performanse klasifikatora treniranog funkcijom cijene dvostruke pouzdanosti nad skupom CIFAR-10 i primjerima iz standardne normalne distribucije na Imagenet izvan-distribucijskom skupu.

Epoha	TNR at TPR 95%	AUROC	Det Acc	AUPR
10	20.9	75.82	68.91	78.35
20	20.09	78.91	72.46	80.36
30	16.31	73.79	70.79	75.8
40	16.6	78.37	71.5	74.8
50	12.7	75.37	69.12	71.81
60	12.24	74.58	65.95	65.1
70	15.64	78.1	71.18	73.61
80	21.24	81.84	74.92	79.24
90	17.86	77.99	69.75	71.53
100	16.82	80.26	72.24	72.35

Tablica A.4: Performanse klasifikatora treniranog funkcijom cijene dvostruke pouzdanosti nad skupom CIFAR-10 i primjerima iz standardne normalne distribucije na LSUN izvan-distribucijskom skupu.

Dodatak B
SVHN osnovica usporedbe

Epoha	Test Set Accuracy
10	93.19
20	93.40
30	92.91
40	93.75
50	94.12
60	93.85
70	93.21
80	93.68
90	93.83
100	94.17

Tablica B.1: Performanse klasifikatora treniranog funkcijom cijene dvostruke pouzdanosti nad skupom SVHN i primjerima iz standardne normalne distribucije na test setu.

Epoha	TNR at TPR 95%	AUROC	Det Acc	AUPR
10	54.45	93.76	88.39	95.13
20	51.4	93.39	87.62	94.11
30	49.13	93.3	87.77	93.71
40	52.07	93.79	88.43	93.77
50	50.11	90.59	87.91	91.72
60	46.76	93.64	86.96	86.54
70	40.05	91.42	83.75	85.12
80	63.03	95.37	90.14	93.96
90	46.85	93.44	86.65	86.56
100	57.88	88.91	89.72	89.97

Tablica B.2: Performanse klasifikatora treniranog funkcijom cijene dvostruke pouzdanosti nad skupom SVHN i primjerima iz standardne normalne distribucije na CIFAR-10 izvan-distribucijskom skupu.

Epoha	TNR at TPR 95%	AUROC	Det Acc	AUPR
10	56.7	94.24	88.99	95.6
20	57.78	94.37	88.81	95.14
30	52.62	93.82	88.45	94.22
40	55.64	94.25	89.0	94.42
50	52.85	94.06	88.52	92.82
60	49.19	93.91	87.44	87.49
70	42.96	91.92	84.37	86.33
80	65.82	95.71	90.41	94.49
90	50.18	93.85	87.43	88.07
100	59.31	95.31	90.01	90.54

Tablica B.3: Performanse klasifikatora treniranog funkcijom cijene dvostruke pouzdanosti nad skupom SVHN i primjerima iz standardne normalne distribucije na Imagenet izvan-distribucijskom skupu.

Epoha	TNR at TPR 95%	AUROC	Det Acc	AUPR
10	52.51	93.36	87.83	94.81
20	55.97	94.1	88.56	94.92
30	53.19	93.94	88.55	94.42
40	55.34	94.23	88.97	94.7
50	51.57	93.79	87.9	91.8
60	46.73	93.65	87.13	86.83
70	44.58	92.34	85.38	87.95
80	65.33	95.67	90.62	94.83
90	48.3	93.15	86.93	90.59
100	58.19	95.21	89.61	89.81

Tablica B.4: Performanse klasifikatora treniranog funkcijom cijene dvostruke pouzdanosti nad skupom SVHN i primjerima iz standardne normalne distribucije na LSUN izvan-distribucijskom skupu.

Detekcija izvan-distribucijskih primjeraka suparničkim učenjem

Sažetak

Klasifikacija slika je važan zadatak računalnog vida koji u različitim primjenama postiže vrhunske rezultate, no korišteni diskriminativni konvolucijski modeli su skloni neopravdanom optimizmu što ih čini nespremnima za šire praktično prihvaćanje. Razmotren je pristup učenju klasifikatora usmjeren ka razdvajanju prediktivnih distribucija unutar- i izvan-distribucijskih primjeraka za lakšu izgradnju detektora izvan-distribucijskih primjeraka zasnovanog na pragu, gdje se generativni suparnički model koristi za uzorkovanje najkorisnijih izvan-distribucijskih primjeraka u fazi učenja. Performanse dobivenih klasifikatora su ispitane uporabom standardiziranog skupa metrika za problem detekcije izvan-distribucijskih primjeraka preko više različitih izvan-distribucijskih skupova različite težine.

Ključne riječi: računalni vid, detekcija izvan-distribucijskih primjeraka, nadzirano učenje, suparničko učenje

Out-of-distribution detection by adversarial learning

Abstract

Image classification is an important task of computer vision that achieves state-of-the-art results with various applications, but the discriminative convolutional models used are prone to unjustified optimism, making them unsuitable for wider practical adoption. A novel training method is considered which yields classifiers that more effectively separate predictive distributions of in- and out-of-distribution samples, enabling the construction of better threshold-based out-of-distribution detectors. The robust approach harnesses generative adversarial networks during training to sample the most useful out-of-distribution samples to present to the classifier. Obtained classifiers were evaluated using a standard set of metrics for measuring out-of-distribution detection performance on several unseen datasets of different complexity.

Keywords: computer vision, out-of-distribution detection, supervised learning, adversarial learning