SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 872

VARIJACIJSKO UČENJE NA ZAŠUMLJENIM OZNAKAMA

Dominik Jambrović

Zagreb, lipanj, 2025.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 3. ožujka 2025.

DIPLOMSKI ZADATAK br. 872

Pristupnik:	**Dominik Jambrović (0036534818)**
Studij:	Računarstvo
Profil:	Računarska znanost
Mentor:	prof. dr. sc. Siniša Šegvić

Zadatak: **Varijacijsko učenje na zašumljenim oznakama**

Opis zadatka:

Raspoznavanje slika važan je problem računalnog vida s mnogim zanimljivim primjenama. U posljednje vrijeme stanje tehnike postižu duboki modeli zasnovani na konvolucijama i slojevima pažnje. Međutim, standardni postupci teško se nose sa zašumljenim oznakama.. U okviru rada, potrebno je odabrati okvir za automatsku diferencijaciju te upoznati biblioteke za rukovanje tenzorima i slikama. Proučiti i ukratko opisati postojeće duboke arhitekture za raspoznavanje slika s posebnim naglaskom na prednaučene samonadzirane modele. Odabrati slobodno dostupne skupove slika te oblikovati podskupove za učenje, validaciju i testiranje. Formulirati optimizacijski cilj s latentnim predikcijama čistih razreda te predložiti rješenje utemeljeno na varijacijskoj aproksimaciji te maksimiziranju očekivanja. Komentirati učinkovitost učenja i zaključivanja. Predložiti pravce za budući rad. Radu priložiti izvorni i izvršni kod razvijenih postupaka, ispitne slijedove i rezultate, uz potrebna objašnjenja i dokumentaciju. Citirati korištenu literaturu i navesti dobivenu pomoć.

Rok za predaju rada: 4. srpnja 2025.

Zahvale!

Sadržaj

1. Uvod

Duboki modeli koriste se u brojnim aspektima naše svakodnevice. Pri razvoju i učenju modela, pažnju prije svega posvećujemo performansama na neviđenim podatcima - želimo naučiti modele koji dobro generaliziraju. Drugim riječima, želimo da modeli daju ispravna predviđanja za viđene, ali i za neviđene podatke. Ovime osiguravamo da naša rješenja imaju primjenu i van laboratorijskih uvjeta u kojima se uče.

U procesu razvoja modela za određeni zadatak strojnog učenja, osim odabira arhitekture, algoritma učenja i hiperparametara, veliku ulogu igraju podatci na kojima učimo. Općenito govoreći, prikupljanje i označavanje podataka jedan je od najskupljih dijelova procesa razvoja rješenja za nekih problem. Važno je da prikupljeni podatci što realističnije predstavljaju stvarne situacije s kojima će se naš model susretati tj. da distribucija podataka odgovara stvarnoj distribuciji situacija koje prikazuju. Dodatno, pokazuje se da duboki modeli uz dovoljan kapacitet mogu naučiti ispravno predviđati oznake čak i za nasumično označene podatke [1], tako da je veoma važno da su prikupljeni podatci što točnije označeni.

Područje računalnog vida [2] bavi se razvojem algoritama i modela za brojne zadatke raspoznavanja i razumijevanja slika. Najčešći zadatak je klasifikacija slika - model na ulazu dobiva sliku, a na izlazu treba predvidjeti razred koji odgovara ulaznom primjeru. Iako postoje brojni skupovi slikovnih podataka koji se mogu koristiti za učenje i evaluaciju modela, za konkretne zadatke u većini slučajeva trebamo prikupiti i označiti vlastite slike. Pritom postoji nekoliko čestih opasnosti: prisutnost zatrovanih podataka [3] ili zašumljenih oznaka [4].

Kada govorimo o trovanju podataka, maliciozni agent u skup podataka dodaje zatrovane podatke s ciljem manipulacije izlaza naučenog modela za određene ulaze. S druge strane, anotator podataka bez zlih namjera određenim podatcima može pridijeliti netočne oznake, time dodajući podatke sa zašumljenim oznakama u skup. Kroz vrijeme, razvili su se brojni algoritmi za obranu modela od zatrovanih podataka [5, 6, 7], kao i za učenje na zašumljenim oznakama [8, 9]. Ipak, većina radova se fokusira na samo jedan od ovih problema, a ne na razvoj algoritma koji se može nositi s oba problema.

Cilj ovog rada je reproducirati i poboljšati rezultate okvira za obranu od zatrovanih podataka imena VIBE (engl. *Variational inference for backdoor elimination*) [10]. Osim ovoga, cilj je i primijeniti VIBE na problem zašumljenih oznaka. Pritom VIBE evaluiramo na nekoliko čestih vrsta trovanja odnosno zašumljivanja oznaka kako bi se osigurala robusnost okvira. Dodatno, cilj je usporediti VIBE sa stanjem tehnike (engl. *state of the art - SotA*) za problem zašumljenih oznaka.

2. Problem zatrovanih podataka

Cilj dodavanja zatrovanih podataka [3] u skup je ugrađivanje stražnjih vrata (engl. *backdoor*) u naučeni model. Ako napad uspije, napadač može kontrolirati izlaz modela koristeći suptilne izmjene ulaznog primjera. Općenito govoreći, stvaranje zatrovanih podataka podrazumijeva dodavanje vizualnog okidača na ulazni primjer, kao i prikladnu izmjenu oznaka. Pritom napadač radi izmjenu određenog udjela podataka, dok preostali podatci ostaju neizmjenjeni. Hiperparametar koji opisuje udio zatrovanih podataka zvat ćemo stopom trovanja (engl. *poisoning rate*). Pojedine metode trovanja podataka razlikuju se po načinu dodavanja okidača tj. načinu izmjene ulaznih primjera, kao i po načinu izmjene oznaka.

Kada govorimo o načinu izmjene ulaznih primjera, možemo napraviti podjelu na lokalne i globalne izmjene primjera. Kod lokalnih izmjena, mijenja se samo određeno područje slike, najčešće dodavanjem zadanog okidača na to područje [11]. S druge strane, kod globalnih izmjena se mijenja cijela slika koristeći različite tehnike poput miješanja slike s okidačem [12] ili transformiranja slike na temelju zadanog deformacijskog polja [13]. Osim korištenja jednog okidača za sve zatrovane podatke, određeni napadi koriste okidače specifične za pojedini uzorak [14].

Većinu napada možemo svrstati u jedan od dva načina izmjena oznaka: *all-to-one* i *all-to-all* izmjena oznaka [15]. Kod *all-to-one* metode, primjeri dobivaju zatrovanu oznaku jednog proizvoljno odabranog razreda neovisno o originalnim oznakama pojedinih primjera. S druge strane, kod *all-to-all* metode, primjeri dobivaju zasebne zatrovane oznake ovisno o originalnim oznakama. Osim ova dva načina izmjena oznaka, određeni napadi uopće ne mijenjaju oznake zatrovanih primjera, već se oslanjaju isključivo na jače izmjene ulaznih primjera. Ovakve napade zovemo napadi s čistim oznakama (engl. *clean-label attacks*) [16].

Uspješnost pojedinog napada mjerimo metrikom imena stopa uspješnosti napada (engl. *attack success rate* - ASR). Ovu mjeru definiramo kao točnost modela mjerenu isključivo na zatrovanim primjerima. Cilj algoritama za obranu od zatrovanih podataka je naučiti čisti model na zatrovanom skupu. Drugim riječima, glavni cilj obrane je naučiti model koji ima izvrsne performanse na čistim podatcima, ali i što nižu stopu uspješnosti napada.

2.1. Primjeri metoda trovanja podataka

U ovome radu, fokusiramo se na tri metode trovanja podataka: napade BadNets [11], Blend [12] te WaNet [13].

2.1.1. Napad BadNets

Napad BadNets uobičajeno dodaje jedan zadani okidač na svaki odabrani ulazni primjer. Okidač možemo shvatiti kao uzorak piksela koji se dodaje na specifično mjesto na slici. Na primjer, okidač može biti bijeli pravokutnik pozicioniran u donjem lijevom kutu slike. Naravno, korišteni uzorak može biti proizvoljne kompleksnosti i veličine. Kod napada BadNets, izmjene oznaka su najčešće tipa *all-to-one*, ali česte su i izmjene tipa *all-to-all*.

Slika 2.1. Primjer primjene napada BadNets. Izvornoj slici (lijevo) dodaje se okidač kako bi nastala zatrovana slika (desno).

2.1.2. Napad Blend

Napad Blend provodi miješanje zadanog okidača sa svakim odabranim ulaznim primjerom. Pritom je jačina napada određena hiperparametrom α koji nazivamo jačina miješanja (engl. *blending strength*). Primjenu napada Blend možemo prikazati jednadžbom:

$$\tilde{\boldsymbol{x}} = (1 - \alpha) \cdot \boldsymbol{x} + \alpha \cdot \boldsymbol{t} \tag{2.1}$$

Pri čemu $\boldsymbol{x}$ označava ulazni primjer, $\boldsymbol{t}$ okidač, a $\tilde{\boldsymbol{x}}$ zatrovani primjer. Kod napada Blend, izmjene oznaka su uobičajeno tipa *all-to-one*.

Slika 2.2. Primjer primjene napada Blend. Izvorna slika (lijevo) miješa se s okidačem uz $\alpha = 0.2$ kako bi nastala zatrovana slika (desno).

2.1.3. Napad WaNet

Napad WaNet provodi geometrijsku transformaciju svakog odabranog ulaznog primjera koristeći nasumično generirano deformacijsko polje. Deformacijsko polje svakom pikselu odredišne slike dodjeljuje vektor pomaka prema pikselu izvorne slike. Pritom hiperparametar k određuje veličinu nasumično generiranog polja šuma na temelju kojeg se skaliranjem i interpolacijom dobiva konačno deformacijsko polje, a hiperparametar s određuje jačinu deformacije. Primjenu napada WaNet možemo definirati jednadžbom:

$$\tilde{\boldsymbol{x}} = \mathcal{W}(\boldsymbol{x}, \boldsymbol{M}(k, s)) \tag{2.2}$$

Pri čemu $\boldsymbol{x}$ označava ulazni primjer, $\boldsymbol{M}$ deformacijsko polje generirano uz hiperparametre k i s, $\mathcal{W}$ primjenu deformacijskog polja na ulazni primjer, a $\tilde{\boldsymbol{x}}$ zatrovani primjer. Kao i kod napada Blend, kod napada WaNet su izmjene oznaka uobičajeno tipa *all-to-one*.

Slika 2.3. Primjer primjene napada WaNet. Izvorna slika (lijevo) transformira se koristeći deformacijsko polje uz $k = 8$ i $s = 4$ kako bi nastala zatrovana slika (desno). Hiperparametri k i s su uvećani kako bi učinak trovanja bio uočljiviji.

2.2. Primjeri algoritama za obranu od zatrovanih podataka

U ovome radu, rezultate okvira VIBE uspoređujemo s rezultatima triju algoritama za obranu od zatrovanih podataka: *Anti-backdoor learning* (ABL) [5], *Decoupling based defense* (DBD) [6] te *Adaptively splitting dataset-based defense* (ASD) [7].

2.2.1. Obrana ABL

Algoritam *Anti-backdoor learning* (ABL) sastoji se od dva glavna koraka: izoliranje zatrovanih podataka (engl. *backdoor isolation*) i odučavanje trovanja (engl. *backdoor unlearning*). Osnovna ideja ove obrane je da se nakon određenog broja epoha učenja uz posebno definiran gubitak izolira određeni broj primjera za koje se smatra da su zatrovani. Nakon prvog koraka, ti se primjeri koriste za odučavanje trovanja, dok se preostali primjeri koriste za standardno učenje.

Konkretno, cilj prvog koraka je zadržati vrijednost gubitka svakog pojedinog primjera oko praga γ. Kako bi ovo postigli, autori predlažu korištenje gradijentnog uspona u slučaju da gubitak primjera padne ispod praga, dok se inače koristi gradijentni spust. Gubitak u prvom koraku možemo prikazati jednadžbom:

$$\mathcal{L}_1 = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\operatorname{sign}\left(\ell(f_\theta(\boldsymbol{x}),y)-\gamma\right)\cdot\ell(f_\theta(\boldsymbol{x}),y)\right] \tag{2.3}$$

Pritom $(\boldsymbol{x}, y) \sim \mathcal{D}$ označava primjer $\boldsymbol{x}$ s pripadnom oznakom y iz skupa podataka $\mathcal{D}$, $f_\theta(\boldsymbol{x})$ izlaz modela parametriziranog parametrima θ, $\ell(\hat{y}, y)$ gubitak za predviđenu oznaku $\hat{y}$ i stvarnu oznaku y, a sign operaciju signum.

Ideja je da će gubitak za zatrovane primjere veoma brzo pasti ispod praga te će se za njih često aktivirati gradijentni uspon, dok će gubitak čistih primjera sporije padati i stabilizirati se oko praga. Nakon zadanog broja epoha, izolira se udio p primjera s najnižim gubitkom i proglašava potencijalnim zatrovanim skupom. Važno je napomenuti da bismo prvi korak ABL-a mogli zamijeniti proizvoljnim algoritmom detekcije zatrovanih primjera.

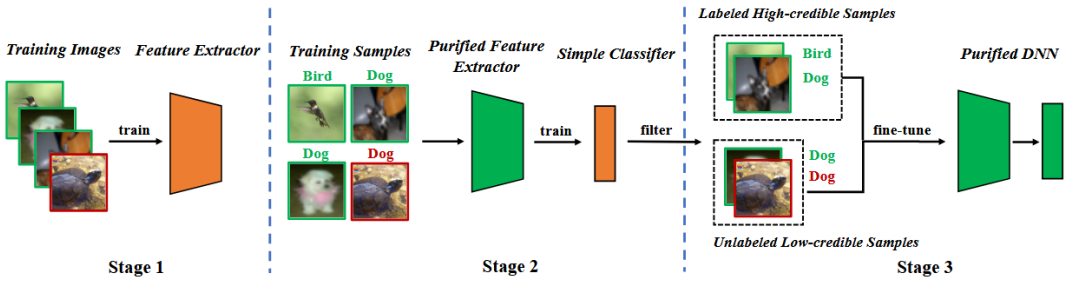
U drugom koraku, učenje se u svakoj epohi provodi zasebno za procijenjeni čisti odnosno zatrovani skup. Dok se učenje na čistom skupu provodi uz standardni gradijentni spust, učenje na zatrovanom skupu provodi se uz gradijentni uspon kako bi model odučili od trovanja. Ovo je moguće zato što je trovanje najčešće realizirano uz samo jedan ciljni razred tj. uz *all-to-one* način izmjene oznaka. Gubitak u drugom koraku možemo prikazati jednadžbom:

$$\mathcal{L}_2 = \mathbb{E}_{(\boldsymbol{x},y)\sim\hat{\mathcal{D}}_c}\left[\ell(f_\theta(\boldsymbol{x}),y)\right] - \mathbb{E}_{(\boldsymbol{x},y)\sim\hat{\mathcal{D}}_b}\left[\ell(f_\theta(\boldsymbol{x}),y)\right] \tag{2.4}$$

Pritom $\hat{\mathcal{D}}_c$ označava procijenjeni čisti skup, a $\hat{\mathcal{D}}_b$ procijenjeni zatrovani skup.

2.2.2. Obrana DBD

Algoritam *Decoupling based defense* (DBD) problem učenja modela na zatrovanim podatcima razdvaja na tri koraka. Pritom DBD model tretira kao dvije povezane cjeline: ekstraktor značajki (engl. *feature extractor*) te klasifikator (najčešće nekoliko potpuno-povezanih slojeva). Ekstraktor značajki za ulazni primjer na izlazu daje vektor značajki tj. ugrađivanje (engl. *embedding*) u latentnom metričkom prostoru. S druge strane, klasifikator za ugrađivanje na ulazu predviđa jednu od mogućih oznaka.



Slika 2.4. Prikaz glavnih koraka obrane DBD. Preuzeto iz[6]

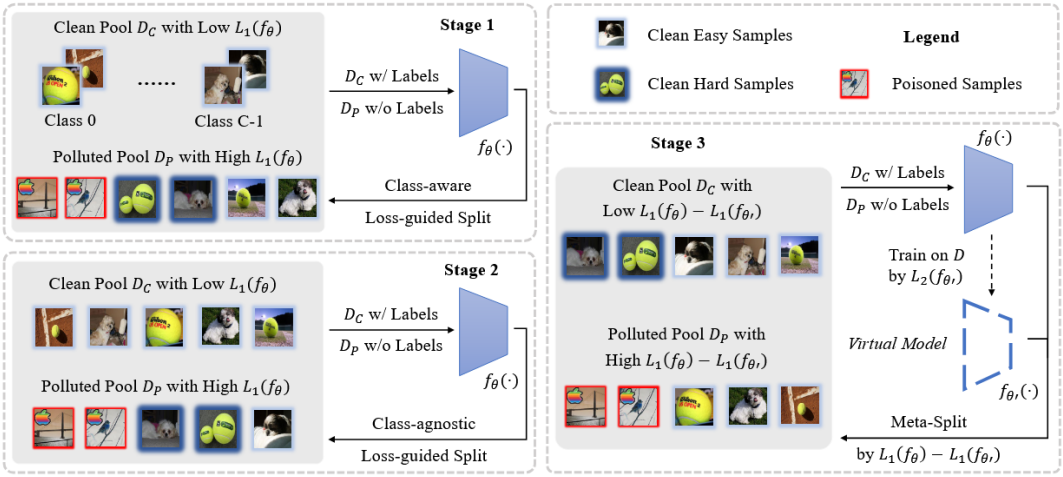
U prvom koraku, DBD uči ekstraktor značajki koristeći proizvoljan algoritam samonadziranog učenja [17] na slikama bez oznaka. Pošto algoritmi samonadziranog učenja uobičajeno uključuju korištenje jakih augmentacija ulaznih primjera, kvaliteta okidača kod zatrovanih primjera bit će narušena. Dodatno, pošto se model u ovoj fazi uči bez oznaka, u prvom koraku nije moguće zatrovati ekstraktor značajki. Drugim riječima, na kraju prvog koraka imat ćemo naučen čisti ekstraktor značajki.

U drugom koraku, parametri ekstraktora značajki su zamrznuti, a klasifikator učimo koristeći klasično nadzirano učenje na podatcima s oznakama. Pritom kao funkciju gubitka koristimo simetričnu unakrsnu entropiju - pokazano je da korištenje iste rezultira višim gubitkom kod zatrovanih primjera [18]. Ipak, ako bismo koristili samo ova dva koraka, dobiveni model ne bi imao performanse jednake stanju tehnike jer smo ekstraktor značajki učili bez oznaka. Zbog toga, važno je napraviti podjelu skupa podataka na čisti i zatrovani skup te nakon toga ugoditi (engl. *fine-tune*) cijeli model. Kod algoritma DBD, ova podjela se radi na temelju iznosa gubitka: udio α primjera s najnižim iznosom gubitka smatrat ćemo vjerodostojnim tj. čistim skupom, dok ćemo preostale primjere smatrati zatrovanima.

Treći korak koristi podjelu na čisti i zatrovani skup kako bi dodatno ugodio parametre cijelog modela. Konkretno, zatrovanom skupu uklanjamo oznake te potom model učimo koristeći proizvoljni algoritam polu-nadziranog učenja [19]. Na kraju ovog koraka, imat ćemo naučen cjelokupni model otporan na zatrovane primjere.

2.2.3. Obrana ASD

Algoritam *Adaptively splitting dataset-based defense* (ASD) problem učenja modela na zatrovanim podatcima razdvaja na tri koraka. Tijekom sva tri koraka, održavaju se dva skupa podataka: čisti i zagađeni skup. Pritom se u čistom skupu nalaze podatci za koje je velika vjerojatnost da su čisti, dok se u zagađenom skupu nalaze zatrovani podatci, kao i preostali čisti podatci. Kroz učenje, čisti skup se povećava dodavanjem primjera iz zagađenog skupa. Konačno, na kraju učenja bi zagađeni skup trebao sadržavati isključivo zatrovane podatke. Parametri modela uvijek se ažuriraju na temelju proizvoljnog polu-nadziranog gubitka, pri čemu se zagađeni skup koristi za učenje bez oznaka.



Slika 2.5. Prikaz glavnih koraka obrane ASD. D_C predstavlja čisti skup, a D_P zagađeni skup. Preuzeto iz[7]

U prvom koraku, čisti skup se inicijalizira na mali broj provjereno čistih primjera (na primjer, po 10 primjera za svaki razred), dok se zagađeni skup inicijalizira na cijeli skup podataka. Svakih t epoha učenja, u čisti skup se iz zagađenog skupa dodaje n primjera iz svakog pojedinog razreda koji imaju najniži gubitak $\mathcal{L}_1$. Pritom kao funkciju gubitka $\mathcal{L}_1$ koristimo simetričnu unakrsnu entropiju kako bi zatrovani primjeri imali što viši gubitak. Ovu podjelu zovemo razredno-svjesna podjela vođena gubitkom (engl. *class-aware loss-guided split*).

Tijekom drugog koraka, čisti skup značajno proširujemo dodavanjem udjela α zagađenog skupa. Pritom se dodaju primjeri iz cijelog skupa (neovisno o razredu) koji imaju najniži gubitak $\mathcal{L}_1$. Ovu podjelu zovemo razredno-nesvjesna podjela vođena gubitkom (engl. *class-agnostic loss-guided split*).

Nakon prva dva koraka ASD-a, u zagađenom skupu preostaju zatrovani primjeri, ali i neki teški primjeri specifični za model. Zbog toga što model nismo učili na teškim primjerima s oznakama, performanse modela trenutno su niže od stanja tehnike. Kako bismo dodali i teške primjere u čisti skup, svaku epohu trećeg koraka konstruiramo virtualni model. Virtualni model inicijaliziramo parametrima glavnog modela te potom provodimo jednu epohu nadziranog učenja na zagađenom skupu uz korištenje gubitka $\mathcal{L}_2$. Kao funkciju gubitka $\mathcal{L}_2$ koristimo standardnu unakrsnu entropiju. Nakon učenja virtualnog modela, mjerimo smanjenje gubitka definirano jednadžbom:

$$\Delta\mathcal{L} = \mathcal{L}_1(f_\theta) - \mathcal{L}_1(f_{\theta'}) \tag{2.5}$$

Pritom f_θ označava glavni model parametriziran parametrima θ, a $f_{\theta'}$ predstavlja virtualni model parametriziran parametrima θ' dobivenim nakon jedne epohe nadziranog učenja na zagađenom skupu. Konačno, udio γ primjera s najmanjim smanjenjem gubitka dodajemo u čisti skup. Intuicija iza ove podjele je da su zatrovani podatci lagani za naučiti, tako da već nakon jedne epohe nadziranog učenja isti imaju veoma nizak gubitak, dok teški čisti primjeri i dalje imaju visok gubitak. Važno je napomenuti da se virtualni model koristi isključivo za provođenje podjele na temelju smanjenja gubitka. Ovu podjelu zovemo meta-podjela (engl. *meta-split*) jer je pristup s učenjem virtualnog modela inspiriran područjem meta-učenja (engl. *meta-learning*) [20]. Na kraju trećeg koraka, u čistom skupu će se nalaziti gotovo svi čisti primjeri, a dobiveni model će imati izvrsne performanse uz otpornost na zatrovane primjere.

3. Problem zašumljenih oznaka

Proces stvaranja skupa vizualnih podataka otprilike možemo podijeliti u tri koraka. U prvom koraku, potrebno je definirati skup razreda tj. oznaka koje mogu biti dodijeljene svakom primjeru. Tijekom drugog koraka, cilj nam je prikupiti što veći skup slika, pritom pazeći na to da slike precizno prikazuju situacije s kojima će se naš model susretati. Dodatno, veoma je važno da je prikupljeni skup što raznovrsniji kako bi model mogao naučiti dobro generalizirati. Konačno, u trećem koraku anotatori označavaju slike na temelju definiranog skupa oznaka. Drugi i treći korak ovog procesa mogu se provoditi jedan za drugim, ali i paralelno - nakon što se prikupi određen broj slika, taj skup se šalje na označavanje, a istovremeno je moguće prikupiti još slika.

Iako anotatori podataka prolaze kroz brojne edukacije kako bi se što bolje upoznali s pravilima na temelju kojih će označavati podatke, određeni podatci su prirodom teži od drugih pa može doći do pogrešnog označavanja. Problem zašumljenih oznaka sličan je problemu zatrovanih podataka po tome što se kod oba problema model mora nositi s pogrešnim oznakama. Ipak, kod problema zašumljenih oznaka podrazumijevamo da su ulazne slike netaknute tj. da ne postoji nikakva izmjena ulaza. Dodatno, kod problema zašumljenih oznaka ne postoji konkretan cilj - podatci sa zašumljenim oznakama nastaju slučajno, bez ikakvih loših namjera.

Dok je kod zatrovanih podataka problem prelako učenje poveznice između prisutnosti okidača i određenog ciljnog razreda, kod podataka sa zašumljenim oznakama je problem činjenica da model uz dovoljan kapacitet može naučiti oznake čak i ako su one nasumično generirane [1]. Drugim riječima, ako nismo oprezni, lako je doći do prenaučenosti (engl. *overfitting*) modela. Naravno, ovakav model će loše generalizirati na ispravno označenim podatcima.

Kako bismo mogli usporediti performanse različitih algoritama za učenje na zašumljenim oznakama, najčešće određenom udjelu čistog skupa podataka dodajemo sintetičke zašumljene oznake. Hiperparametar koji opisuje udio podataka sa zašumljenim oznakama zvat ćemo stopa šuma (engl. *noise rate*). Nakon učenja na zašumljenim podatcima, performanse modela evaluiramo na čistom, ali i na u potpunosti zašumljenom skupu. Cilj algoritama za učenje na zašumljenim oznakama tada je zadržati performanse modela učenog na zašumljenim podatcima što bližima performansama modela učenog na čistim podatcima.

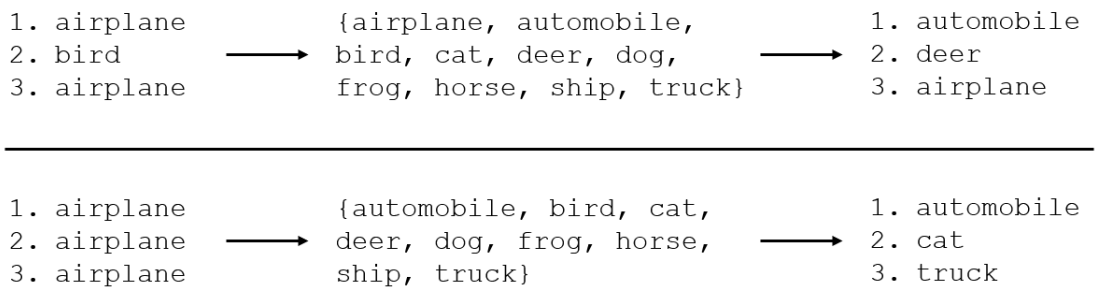
3.1. Primjeri metoda zašumljivanja oznaka

U ovome radu, fokusiramo se na dvije metode zašumljivanja oznaka: simetrično (engl. *symmetric*) i asimetrično (engl. *asymmetric*) zašumljivanje [21].

3.1.1. Simetrično zašumljivanje

Kod simetričnog zašumljivanja, nasumično biramo zadani udio primjera iz čistog skupa podataka te istima dodjeljujemo nasumično generirane nove oznake. Ovu metodu zašumljivanja također zovemo i uniformno zašumljivanje jer svaki razred ima jednaku vjerojatnost postati nova oznaka za neki zadani primjer.

Pritom razlikujemo dvije vrste simetričnog zašumljivanja: inkluzivnu (engl. *symm-inc*) i ekskluzivnu (engl. *symm-exc*) vrstu [21]. Ako govorimo o inkluzivnom simetričnom zašumljivanju, stvarna oznaka za neki zadani primjer može biti odabrana i kao zašumljena oznaka. S druge strane, kod ekskluzivnog simetričnog zašumljivanja ignoriramo stvarnu oznaku tj. zašumljena oznaka se uvijek razlikuje od stvarne oznake primjera. U našem radu, fokusiramo se na inkluzivno simetrično zašumljivanje.

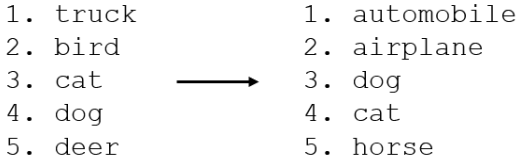


Slika 3.1. Prikaz inkluzivnog (gore) odnosno ekskluzivnog (dolje) simetričnog zašumljivanja za tri odabrana primjera iz skupa CIFAR10 [22]. Čiste oznake (lijevo) preslikavaju se na jednu od mogućih oznaka (sredina) i nastaju zašumljene oznake (desno). Vidimo da primjeri s istim čistim oznakama nemaju nužno i iste zašumljene oznake.

3.1.2. Asimetrično zašumljivanje

Kod asimetričnog zašumljivanja, prvo nasumično biramo zadani udio primjera zasebno za svaki razred. Drugim riječima, ako je u pitanju skup CIFAR10 [22], za svaki od 10 razreda nasumično ćemo odabrati zadani udio primjera kojima ćemo potencijalno dodijeliti zašumljene oznake.

Nakon početnog odabira, svakom primjeru dodjeljujemo novu oznaku na temelju predodređenog preslikavanja. Pritom je moguće da određeni razredi nemaju definirano preslikavanje u neki drugi razred, tako da odabranim primjerima iz tih razreda neće biti dodijeljene zašumljene oznake. Preslikavanje na temelju kojeg se pojedinim primjerima dodjeljuje nova oznaka unaprijed je definirano za pojedine skupove podataka poput skupova MNIST [23], CIFAR10 i CIFAR100 [22]. Kod skupova s hijerarhijskim odnosom razreda poput skupa CIFAR100, preslikavanja su definirana nasumično unutar pojedinih grupa razreda.



Slika 3.2. Prikaz preslikavanja razreda za skup CIFAR10. Odabranim primjerima se na temelju čistih oznaka (lijevo) dodjeljuju zašumljene oznake (desno). Pritom odabrani primjeri s istim čistim oznakama uvijek imaju i iste zašumljene oznake.

3.2. Primjeri algoritama za učenje na zašumljenim oznakama

U ovome radu, rezultate prilagođenog okvira VIBE uspoređujemo s rezultatima dvaju algoritama za učenje na podatcima sa zašumljenim oznakama: *Sparse over-parameterization* (SOP) [8] te *Imprecise label learning* (ILL) [9].

3.2.1. Algoritam SOP

3.2.2. Algoritam ILL

4. Samonadzirano učenje

5. Algoritam maksimizacije očekivanja

6. Transportni problem

7. VIBE

8. Skup podataka

9. Eksperimenti

10. Zaključak

Literatura

[1] C. Zhang, S. Bengio, M. Hardt, B. Recht, i O. Vinyals, "Understanding deep learning requires rethinking generalization", *arXiv preprint arXiv:1611.03530*, 2016.

[2] A. Voulodimos, N. Doulamis, A. Doulamis, i E. Protopapadakis, "Deep learning for computer vision: A brief review", *Computational intelligence and neuroscience*, sv. 2018, br. 1, str. 7068349, 2018.

[3] B. Biggio, B. Nelson, i P. Laskov, "Poisoning attacks against support vector machines", *arXiv preprint arXiv:1206.6389*, 2012.

[4] S. Gupta i A. Gupta, "Dealing with noise problem in machine learning data-sets: A systematic review", *Procedia Computer Science*, sv. 161, str. 466–474, 2019.

[5] Y. Li, X. Lyu, N. Koren, L. Lyu, B. Li, i X. Ma, "Anti-backdoor learning: Training clean models on poisoned data", *Advances in Neural Information Processing Systems*, sv. 34, str. 14 900–14 912, 2021.

[6] K. Huang, Y. Li, B. Wu, Z. Qin, i K. Ren, "Backdoor defense via decoupling the training process", *arXiv preprint arXiv:2202.03423*, 2022.

[7] K. Gao, Y. Bai, J. Gu, Y. Yang, i S.-T. Xia, "Backdoor defense via adaptively splitting poisoned dataset", u *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023., str. 4005–4014.

[8] S. Liu, Z. Zhu, Q. Qu, i C. You, "Robust training under label noise by over-parameterization", u *International Conference on Machine Learning*. PMLR, 2022., str. 14 153–14 172.

[9] H. Chen, A. Shah, J. Wang, R. Tao, Y. Wang, X. Li, X. Xie, M. Sugiyama, R. Singh, i B. Raj, "Imprecise label learning: A unified framework for learning with various imprecise label configurations", *Advances in Neural Information Processing Systems*, sv. 37, str. 59 621–59 654, 2024.

[10] I. Sabolić, M. Grcić, i S. Šegvić, "Seal your backdoor with variational defense", *arXiv preprint arXiv:2503.08829*, 2025.

[11] T. Gu, K. Liu, B. Dolan-Gavitt, i S. Garg, "Badnets: Evaluating backdooring attacks on deep neural networks", *IEEE Access*, sv. 7, str. 47 230–47 244, 2019.

[12] Y. Chen, X. Gong, Q. Wang, X. Di, i H. Huang, "Backdoor attacks and defenses for deep neural networks in outsourced cloud environments", *IEEE Network*, sv. 34, br. 5, str. 141–147, 2020.

[13] A. Nguyen i A. Tran, "Wanet–imperceptible warping-based backdoor attack", *arXiv preprint arXiv:2102.10369*, 2021.

[14] Y. Li, Y. Li, B. Wu, L. Li, R. He, i S. Lyu, "Invisible backdoor attack with sample-specific triggers", u *Proceedings of the IEEE/CVF international conference on computer vision*, 2021., str. 16 463–16 472.

[15] K. D. Doan, Y. Lao, i P. Li, "Marksman backdoor: Backdoor attacks with arbitrary target class", *Advances in Neural Information Processing Systems*, sv. 35, str. 38 260–38 273, 2022.

[16] M. Barni, K. Kallas, i B. Tondi, "A new backdoor attack in cnns by training set corruption without label poisoning", u *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019., str. 101–105.

[17] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, i F. Makedon, "A survey on contrastive self-supervised learning", *Technologies*, sv. 9, br. 1, str. 2, 2020.

[18] Y. Wang, X. Ma, Z. Chen, Y. Luo, J. Yi, i J. Bailey, "Symmetric cross entropy for robust learning with noisy labels", u *Proceedings of the IEEE/CVF international conference on computer vision*, 2019., str. 322–330.

[19] J. E. Van Engelen i H. H. Hoos, “A survey on semi-supervised learning”, *Machine learning*, sv. 109, br. 2, str. 373–440, 2020.

[20] R. Vilalta i Y. Drissi, “A perspective view and survey of meta-learning”, *Artificial intelligence review*, sv. 18, str. 77–95, 2002.

[21] F. R. Cordeiro i G. Carneiro, “A survey on deep learning with noisy labels: How to train your model when you cannot trust on the annotations?” u *2020 33rd SIBGRAPI conference on graphics, patterns and images (SIBGRAPI)*. IEEE, 2020., str. 9–16.

[22] A. Krizhevsky, G. Hinton *et al.*, “Learning multiple layers of features from tiny images”, 2009.

[23] L. Deng, “The mnist database of handwritten digit images for machine learning research [best of the web]”, *IEEE signal processing magazine*, sv. 29, br. 6, str. 141–142, 2012.

Sažetak

Varijacijsko učenje na zašumljenim oznakama

Dominik Jambrović

Sažetak...

Ključne riječi: prva ključna riječ; druga ključna riječ; treća ključna riječ