УНИВЕРЗИТЕТ У БЕОГРАДУ
МАТЕМАТИЧКИ ФАКУЛТЕТ

Растко Ђорђевић

ПРИЛАГОЂАВАЊЕ МОДЕЛА МАШИНСКОГ УЧЕЊА ПРОМЕНИ ДОМЕНА СУПАРНИЧКИМ ТРЕНИНГОМ

мастер рад

Београд, 2022.

Ментор:

др Младен Николић, ванредни професор
Универзитет у Београду, Математички факултет

Чланови комисије:

др Јована Ковачевић, доцент
Универзитет у Београду, Математички факултет

др Александар Картељ, доцент
Универзитет у Београду, Математички факултет

Датум одбране: 28.09.2022.

Велику захвалност дугујем ментору др Младену Николићу за сву помоћ током израде рада.

Наслов мастер рада: Прилагођавање модела машинског учења промени домена супарничким тренингом

Резиме:

Приликом коришћења техника машинског учења у реалном свету услед променљивих решења и пропуста приликом прикупљања податка често је неопходно прилагодити решења новим доменима. Ово је неопходно да би решења континуирано давала добре резултате. Циљ овог рада је приказ приступа за решавање овог проблема који се заснива на супарничком учењу и део је подобласти прилагођавања модела променама у домену која тренутно даје најбоље резултате. Биће приказана употреба технике обртања градијента која на иновативан начин омогућава прилагођавање модела променама домена. Систем направљен током рада показао је одличне резултате за проблем класификације цифара у различитим доменима и додатно се може користити за произвољне проблеме и домене који су довољно слични.

Кључне речи: машинско учење, вештачка интелигенција, рачунарство, супарничко учење, прилагођавање, домен

Садржај

Глава 1

Увод

Приликом примене машинског учења над проблемима у реалном свету често се захтева да решења раде исправно за разна окружења и током дужег временског периода. Ови захтеви, иако звуче једноставно, представљају озбиљан проблем због статистичке природе решења која се могу направити коришћењем техника машинског учења. Управо та природа их тера да се ослањају на законитости које се могу пронаћи унутар података. Ове законитости у подацима имају незгодну особину да се разликују на разне начине између окружења над којим се решење обучава и окружења над којим се касније примењује.

Промена у законитостима представља све разлике између домена скупа података над којим се тренира и података над којим ће се касније решење примењивати, што је заправо промена домена. Разлог за различитости у доменима може лежати у начину на који су подаци прикупљени или у начину на које се окружење мења кроз време.

Прикупљање и припрема података је изузетно тежак проблем који иако звучи једноставно, у себи крије мноштво малих корака, од којих је сваки неопходан да би се омогућило успешно креирање решења. Веома је лако прикупити податке који су пристрасни и различити од података над којим ће се решење касније примењивати. Понекада је чак и немогуће прикупити податке из реалног окружења па се мора прибећи неким другим техникама као што је коришћење података из другог сличног домена или симулирање вештачког окружења.

Да би се прикупљени подаци могли користити морају се припремити и означити. Приликом припреме података треба пазити да се не одбаце подаци

који су корисни и у исто време да се уоче све грешке које су се десиле током прикупљања. Означавање података такође може унети пристрасност у податке на разне начине. Често означавање раде људи који својим грешкама и субјективношћу доприносе пристрасности. Лошим избором ознака се може изгубити доста информација па тиме додати пристрасност.

Чак и у случају да је прикупљање и чишћење прошло идеално и да домен података савршено осликава домен над којим ће решење бити примењено постоји још један проблем – мењање окружења током времена. У реалном свету окружења су скоро увек комплексна и динамична, што онемогућава предвиђање каква ће бити у будућности. Неке промене се могу превазићи робусним решењима, али за довољно велике промене је неопходно прилагодити решење.

Ако се примети да постоји разлика у доменима, први корак је установити који је њен узрок. Након тога треба прикупити нове податке који боље представљају окружење над којим ће се решење примењивати. Коначни корак је прилагодити решење новим подацима.

Цео процес прилагођавања решења новим подацима у индустрији доста компликује израду, одржавање и унапређивање система који почивају на техникама машинског учења, јер је неопходно континуално прилагођавати решења током времена. Коришћењем метода прилагођавања модела променама у домену се стога могу олакшати израда, унапређивање и одржавање система у реалним индустријским применама.

Циљ овог рада је да прикаже примену једног од приступа прилагођавања модела променама у домену описаног у раду [5] на више различитих скупова података и тиме покаже корисност ових техника. Приступ који се користи почива на учењу репрезентација података које поред информација неопходних за погађање циљне променљиве у себи треба да се обучи да не садржи информације којим се могу разликовати домени. Избацивање информације о доменима постиже се супарничким учењем чиме се повећава способност модела да генерализује.

У оквиру рада коришћени су скупови података *MNIST*, *MNIST-M* који у себи садрже руком писане цифре као и скуп података *SVHN* који се састоји од слика цифара са знакова уличних бројева. Ови скупови су често коришћени у области прилагођавања домену што омогућава брзо поређење приступа описаног у раду са осталим приступима у области.

Над описаним скуповима података показано је да приступ који се користи

даје знатна побољшања у поређењу са решењима која су обучавана само на изворном домену. Иако се не постижу резултати који су упоредиви са резултатима решења која су обучавана на циљном домену, ово се не може ни очекивати у случајевима у којима нису доступни сами подаци из циљног домена.

Систем [17] који је направљен у сврхе овог рада се може користити и за друге скупове података чији домени су довољно слични. Овај систем би се могао користити као део континуалног праћења модела машинског учења у реалности и његовог аутоматског прилагођавања модела променама у домену.

У другом поглављу рада биће дат опис релевантних појмова из области машинског учења који су неопходни за разумевање остатка рада. Наредно поглавље даје детаљнији увод у област прилагођавања модела променама у домену унутар ког су објашњени мотивација и концепти који ће бити коришћени током експеримената. Четврто поглавље у себи садржи опис целог експеримента који је извршен заједно са избором и припремом података, описом приступа који је коришћен, евалуацијом и резултатима експеримента. У последњем поглављу биће сумирани закључци и додатно ће бити продискутовани правци за даљи рад и истраживање.

Глава 2

Увод у основне појмове машинског учења

Машинско учење је област вештачке интелигенције која се бави изучавањем индуктивног закључивања и конструкцијом алгоритама који могу да генерализују [18]. Алгоритми машинског учења на улазу примају скуп података а као свој резултат дају моделе машинског учења, који у себи чувају законитости закључене из иницијалних података. Процес извршавања алгоритама машинског учења се колоквијално назива *обучавање модела*. Модел машинског учења на улазу добија податке, на основу којих на излазу даје циљну променљиву.

У зависности од типа података које имамо машинско учење се може поделити у три области:

1. надгледано учење (енг. *supervised learning*);
2. ненадгледано учење (енг. *unsupervised learning*);
3. учење поткрепљивањем (енг. *reinforcement learning*).

Код алгоритама надгледаног учења као улаз поред улазних података добијамо и циљну променљиву (енг. *target variable*) која представља излаз модела за одређене улазне податке. Циљна променљива се током обучавања модела користи како би се усмерио и убрзао процес учења. Назив „надгледано учење" треба да сугерише да је током целог процеса учења познато какве одлуке модел треба да донесе, тако да је могуће исправити грешке у одлучивању и

настојати да се оне минимизују. Надгледано учење је најпримењивији облик учења, јер даје моделе који су употребљиви у многим реалним применама.

Ненадгледано учење је учење код којег немамо циљну променљиву већ само улазне податке, што знатно отежава процес учења. Тиме што није познато шта треба научити ограничавају се могућности тога шта може бити научено. Алгоритми ненадгледаног учења стога најчешће уче врло специфичне повезаности и структуре међу улазним подацима у зависности од тога за шта су дизајниране.

Учење поткрепљивањем [15] се бави алгоритмима код којих имамо нумеричку оцену која одражава квалитет низа улазних података. Ова оцена представља награду која би требало да се пропагира кроз акције низа улазних података како би модел, који се назива агент, бирао све боље и боље акције. Има примене у роботици као и областима где постоје симулирана окружења у којима делају агенти, као што су на пример роботи унутар игрица.

У наставку рада биће коришћене методе из области надгледаног и ненадгледаног учења тако да ће они бити детаљније објашњени у наредним секцијама.

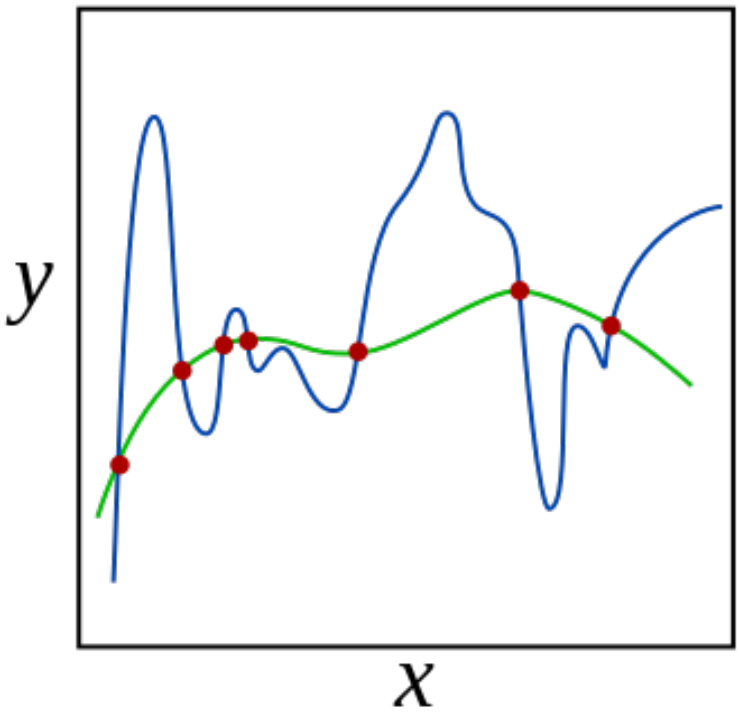
2.1 Битни концепти

У овој секцији биће објашњени основни појмови који су неопходни за разумевање комплекснијих концепата машинског учења. Ови концепти се примењују на различите начине у свим областима машинског учења.

Регуларизација

Током обучавања модела циљ је прилагодити га подацима и омогућити му да генерализује на податке који нису виђени. Једноставни модели нису у стању да се прилагоде комплекснијим законитостима унутар података и овај проблем се назива потприлагођавање (енг. *underfitting*). Са повећавањем комплексности модела расте његова моћ уочавања законитости и могућност преприлагођавања. Проблем са тим је што превише комплексни модели могу пронаћи законитости и тамо где их нема. Овај феномен се назива преприлагођавање (енг. *overfitting*). Регуларизација представља начин борбе против

преприлагођавања тиме што се прави погодан компромис између комплексности модела и његове моћи генерализације.



Слика 2.1: Пример преприлагођавања

На слици 2.1 се може видети пример преприлагођавања. Црвене тачке представљају улазне податке док је зелена линија законитост међу подацима коју треба пронаћи. Ако се изабере превише комплексан модел, као што је случај на слици са полиномом превеликог степена, у потпуности се губи могућност генерализације за остале тачке на зеленој линији. У овом случају се може приметити да је потребно на неки начин ограничити полином тако да ни у једној тачки не може да има превише велики градијент. То ограничење се управо постиже техникама регуларизације.

Методе регуларизације које се често користе код неуронских мрежа су ℓ_1 регуларизација, ℓ_2 регуларизација, унутрашња стандардизација (енг. *batch normalization*) [8], изостављање (енг. *dropout*) [14].

Мере квалитета модела

Приликом избора модела и евалуације најбољег модела неопходно је имати мере којим се може измерити њихов квалитет. Ове мере се разликују у зависности од типа проблема. Добар избор релевантних мера квалитета модела је неопходан за успешно решавање проблема. Будући да ће се у раду разматрати само класификациони проблеми, у наставку ће бити објашњене мере квалитета које су релевантне за њих.

Резултати бинарне класификације се визуелно могу представити матрицом конфузије (енг. *confusion matrix*) као што се може видети на слици 2.2. Истински позитивне инстанце (енг. *TP – true positive*) су инстанце које су позитивне и исправно класификоване као позитивне. Истински негативне инстанце (енг. *TN – true negative*) су инстанце које су негативне и исправно класификоване као негативне. Лажно позитивне инстанце (енг. *FP – false positive*) су негативне и лоше класификоване као позитивне. Лажно негативне инстанце (енг. *FN – false negative*) су позитивне и лоше класификоване као негативне.

За проблем класификовања болести одређеним тестом, истински позитивне инстанце представљају све инстанце теста који су успешно одредили да је пацијент болестан док су истински негативне инстанце ситуације где је тест успешно препознао да је пацијент здрав. Лажно позитивне инстанце би онда биле све инстанце теста који су здравим људима приказали да су болесни. Лажно негативне инстанце су у овом случају најпогубније пошто би оне болесним људима дали резултат да су здрави. Овај пример је илустрован на слици 2.2.

Тачност (енг. *accuracy*) се рачуна као удео исправно класификованих инстанци у односу на укупан број инстанци. Ова мера је интуитивна, али се понаша лоше када су класе небалансиране, то јест када се једна класа појављује пуно више од друге класе.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

Прецизност (енг. *precision*) је мера која представља удео истински позитивних инстанци у односу на све инстанце које су класификоване као позитивне. На овај начин се мери са коликом сигурношћу могу да се прихвате позитивна предвиђања модела.

Слика 2.2: Матрица конфузије

$$Precision = \frac{TP}{TP + FP}$$

Одзив (енг. *recall*) је мера која представља удео истински позитивних инстанци у односу на све инстанце које су позитивне укључујући и оне које су предвиђене као лажно негативне. На овај начин се мери вероватноћа да позитивна инстанца буде исправно класификована од стране модела.

$$Recall = \frac{TP}{TP + FN}$$

Оптимизација

Математичким моделовањем проблем се може представити као функција чије оптимално решење представља решење полазног проблема. У општем случају проблем оптимизације се представља на следећи начин:

$$\min_{x \in D} f(x)$$

где променљива x у машинском учењу представља улазне податке, D је домен улазних података а функција f представља фукнцију губитка коју треба минимизовати. Функција губитка представља укупну грешку коју модел прави за све улазне податке. Минимизовање ове функције може се постићи избором функционалне форме модела и њеним прилагођавањем. Сва ограничења која могу постојати се у машинском учењу најчешће се убацују у функцију f како би се поједноставио процес оптимизације.

Градијентни спуст је једна од најједноставнијих техника за оптимизацију функције. У сваком кораку се израчуна градијент у тренутној тачки и потом се бирају нови аргументи функције и то у правцу и смеру који вредност функције помера у супротном смеру од градијента. На овај начин се текуће решење помера ка локалном минимуму. Може се приметити да се градијентни спуст понаша много боље за добро условљене функције.

Постоје разни начини да се спречи заглављивање у локалном минимуму као и да се поспеши бржа оптимизација. За први проблем се поставља и питање колико често се уопште јавља за довољно комплексне моделе, јер се у практичним проблемима се показало да су локални минимуми веома ретки у случају великог броја параметара због природе високодимензионих простора.

Убрзавање оптимизације се најчешће изводи на 2 начина: паметним избором и мењањем величине корака током оптимизације и узимањем различите количине улазних података током једног корака. Неки од најраспрострањенијих оптимизатора у машинском учењу су Адам и стохастички градијентни спуст (енг. *SGD*).

2.2 Неуронске мреже

Неуронске мреже [6] представљају скуп метода машинског учења које су у последњих 10 година доживеле огроман скок у популарности. Ове методе су познате већ дуже време, али тек недавно је напредак у технологији омогућио њихово ефикасно извршавање. Дају одличне резултате за сирове податке као што су слике, снимци, звук. Решавају тип проблема који је људима веома лак, али је до скоро био немогућ за машине.

Неуронске мреже се могу представити као усмерен ациклични граф који образује једну сложену функцију. Сваки чвор у графу се рачуна као линеарна операција где су операнди чворови чије гране улазе у њега. На излазу сваког

чвора се додаје нелинеарност и то се постиже активационом функцијом (енг. *activation function*). На овај начин чворови који су дубље у мрежи имају већу моћ за учењем комплекснијих законитости у подацима.

Да би било могуће ефикасно извршити оптимизацију неуронских мрежа оне морају бити диференцијабилне. Ово је главни разлог зашто се користе линеарне функције за агрегацију информација из претходних чворова као и активационе функције које су диференцијабилне.

За оптимизацију неуронских мрежа се најчешће користе неке варијације градијентног спуста. Ефикасно извршавање оптимизације алгоритмом за пропагацију уназад је омогућило коришћење неуронских мрежа за најразноврсније проблеме.

Најпознатије архитектуре неуронских мрежа су:

- потпуно повезане мреже (енг. *fully-connected networks*)
- конволутивне мреже (енг. *fonvolution networks*)
- рекурентне мреже (енг. *recurent networks*)
- графовске мреже (енг. *graph neural networks*)

У раду су коришћене потпуно повезане мреже и конволутивне мреже тако да ће оне бити детаљније описане у наставку.

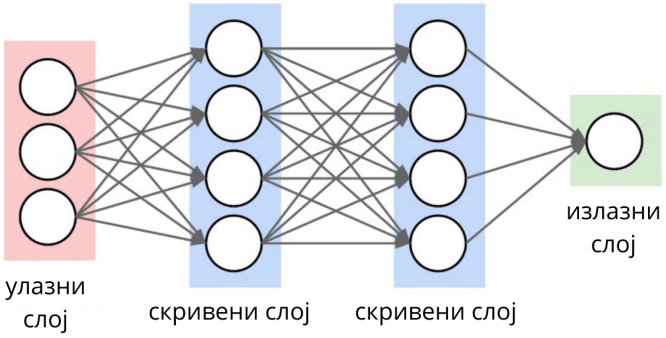
Потпуно повезане мреже

Потпуно повезане мреже се састоје од низа слојева (енг. *layers*) сваки од којих је повезан на улазу са свим неуронима из претходног слоја и чији излаз је повезан са сваким неуроном у наредном слоју, као што се може видети на слици 2.3. На овај начин дубљи слојеви у себи садрже неуроне који имају моћ да опишу све комплексније и апстрактније законитости у подацима.

Први слој се назива улазни слој (енг. *input layer*) и он представља податке које мрежа добија као улаз. Последњи слој је излазни слој (енг. *output layer*) који представља предвиђање мреже. Скривени слојеви (енг. *hidden layer*) су сви остали слојеви. Са повећавањем броја слојева и броја неурона у слојевима расте и комплексност мреже.

Један неурон се може представити као наредна функција:

$$f_w(x) = g(wx)$$



Слика 2.3: Потпуно повезана неуронска мрежа

где x представља вектор који се састоји од излазних вредности претходног слоја, w вектор тежина који се мења током оптимизационог процеса и g активациона функција која додаје нелинеарност у мрежу.

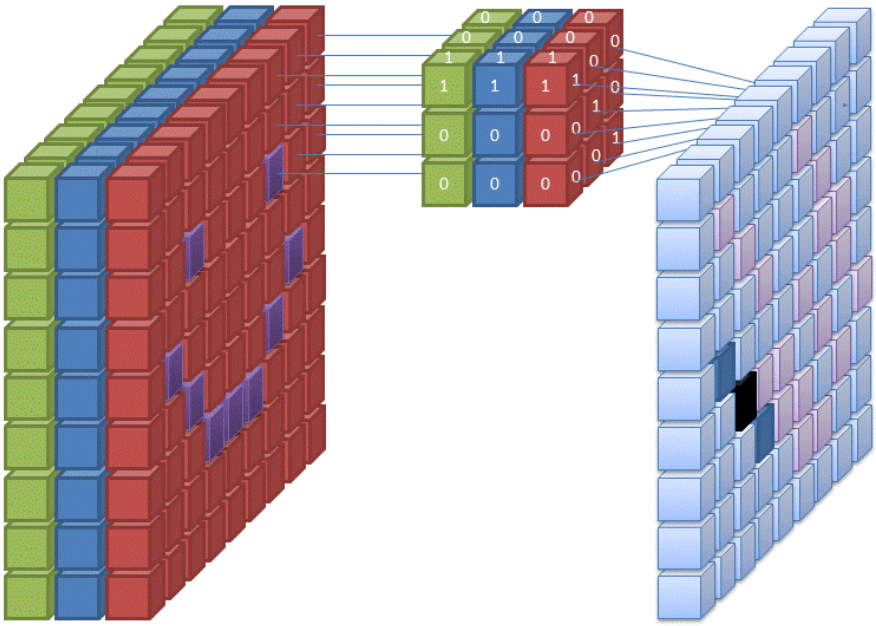
Потпуно повезаним мрежама се такође могу додавати слојеви који се користе за регуларизацију и/или бољу условљеност, као што су слој изостављања (енг. *dropout*) и унутрашња стандардизација (енг. *batch normalization*).

Увођењем слоја изостављања се приликом сваког оптимизационог корака бришу везе између предефинисаног броја псеудо-случајно изабраних парова неурона. На овај начин неуронска мрежа се може гледати као ансамбл више независних неуронских мрежа мање комплексности.

Слој унутрашње стандардизације решава проблем брзог мењања расподеле излаза неурона током оптимизационог процеса. Овај проблем се јавља јер неурони који се брзо мењају представљају улаз у следећи слој, а градијентни спуст је најефикаснији ако су улазни подаци стандардизовани и декорелисани. Унутрашња стандардизација састоји се од рачунања стандардизације сваког излаза неурона пре сваког корака оптимизације.

Конволутивне мреже

Конволутивне мреже се састоје од слојева конволуције и слојева агрегације, заједно са слојевима за регуларизацију као што су изостављање и унутрашња стандардизација. Ове мреже се најчешће користе над сликама, видео снимцима и звуком, али се могу применити над било којим сигналом.



Слика 2.4: Конволутивни слој

Конволутивни слојеви уче филтере који се конволуцијом примењују над улазним сигналом и тиме имају способност да уоче његове битне особине. Из слоја у слој се уче све комплексније и комплексније просторне особине сигнала као што су ивице, па контуре објеката, све до саме класификације објеката.

На слици 2.4 може се видети пример филтера који је $3 \times 3 \times 3$ коцка са јединицама на дијагонали која служи за уочавање ивица из улазне слике са 3 канала и као излаз даје мапу атрибута који представљају локације на слици где се налази ивица. Вредности фитера су параметри које се мењају током оптимизације неуронске мреже.

Слојеви агрегације служе за смањивање димензије улазног сигнала што има неколико ефеката. Један је да чини проблем оптимизације једноставнијим због мањег броја параметара које треба мењати. Такође након агрегације се шири количина информација који накнадни конволутивни филтери могу да користе. Максимум је најпопуларнији избор за агрегацију, али се могу користити и друге функције.

Постоје разне архитектуре конволутивних мрежа и ово је веома популарна област у којој се објављује велики број радова сваке године. Конволутивне мреже се најчешће користе као део комплексније мреже унутар које је улога конволутивне да пронађе особине сигнала који се даље могу обрађивати мрежама као што су потпуно повезане мреже.

2.3 Увод у супарничко учење

Супарничко учење је вид учења популаризован генеративним супарничким мрежама (eng. *generative adversarial networks*) [7] чије обучавање почива на истовременом обучавању генеративног модела и дискриминативног модела. Ова два модела имају међусобно супротстављене циљеве. Генеративни модел генерише слике које дискриминатор не би требало да разликује од правих док дискриминативни модел треба да што боље разликује праве слике од лажних.

Обучавање ова два модела се најчешће врши наизменично. Иако постоје најбоље праксе за обучавање ових модела оно је често нестабилно. Један од разлога нестабилности су нестајући градијенти узроковани природом проблема где је дискриминатор на почетку обучавања веома лако довести до савршених перформанси. Ово се у пракси може решити пажљивим избором броја корака обучавања пре него што се пређе на обучавање другог модела и обрнуто. Други начин да се овај проблем сузбије је мењање функције губитка са намером да оптимизација буде стабилнија.

Генеративне супарничке мреже постижу одличне резултате за проблем генерисања реалистичних слика. Пример генерисаних слика људских лица може се видети на слици 2.5

У општем случају, за супарничко учење је потребно само да постоје 2 дела мреже која имају супротстављене циљеве и који заједно утичу на функцију губитка. У области прилагођавања модела променама у домену овај принцип

(a) генерисана слика 1

(b) генерисана слика 2

Слика 2.5: Пример две слике генерисане генеративним супарничким мрежама

се користи за тренирање дискриминатора који не могу да разликују домене и исто се понашају за изворни и цильни домен.

Различити приступи праве одређене одлуке приликом дизајна као што су избор коришћења генератора или не, избор функције губитка, избор дељења тежина међу слојевима мрежа или не. У раду [16] дат је формални оквир за опис разних супарничких приступа у области прилагођавања модела променама у домену.

2.4 Библиотеке за машинско учење

Током израде рада коришћен је програмски језик *Python* због велике подршке за технике из области машинског учења као и за могућност брзе и ефикасне израде решења у њему. У овој секцији биће дат преглед битних библиотека које су коришћене током рада, а то су:

- python scientific stack (numpy, pandas, sklearn, matplotlib)
- pytorch [11]
- tensorboard [1]

Библиотеке за математичке операције у језику *Python* су неопходне за било какав рад у овом језику. Библиотека *Numpy* омогућава брзу и лаку операцију

над тензорима. Библиотека *Pandas* даје леп интерфејс за манипулисање табеларним подацима. Библиотека *Sklearn* у себи садржи велики број функција које олакшавају припрему података као и неке методе машинског учења. На крају *matplotlib* је библиотека која служи за визуализацију података која се користи као основ за остале библиотеке са сличним наменама.

Pytorch је библиотека за дубоко учење. Омогућава лако креирање неуронских мрежа коришћењем предефинисаних функција за креирање слојева и прављење модела. Срж ове библиотеке је погон за аутоматску диференцијацију који надограђује постојећу библиотеку за рад са тензорима *Numpy*.

Torchvision је библиотека која је блиско интегрисана са *Pytorch* библиотеком. Она у себи садржи неке класичне скупове података, као и велики број модела различитих архитектура који су претходно обучени на великим скуповима података. Ова библиотека стога даје могућност за брзе прототипе и знатно олакшава развој нових решења.

Tensorboard је библиотека за праћење мера квалитета модела и њихову визуализацију током процеса обучавања.

Све поменуте библиотеке су отвореног кода и бесплатне за коришћење. Ово омогућава заједници машинског учења да дели код, акумулира знање и брзо долази до напретка у области.

Глава 3

Прилагођавање модела променама у домену

Као што је описано у уводу током примене машинског учења домен над којем се модели обучавају често не одговара у потпуности домену над којим ће се примењивати. Овај вид генерализације на различите домене је веома тежак за постићи. Из тог разлога се често морају користити неке технике прилагођавања модела променама у домену.

У наставку поглавља ће бити описане неке од разлика између домена које постоје, која је њихова природа и које су методе за њихово решавање. Након тога биће дат приказ метода и приступа за прилагођавање модела променама у домену. На крају поглавља описана је примена ових метода као и резултати које постижу.

3.1 Померај домена

Унутар ове секције ће прво бити описани неки од честих узрока помераја домена у реалним применама машинског учења. Такође ће бити представљен и формалан опис типова помераја домена, када се они појављују и колико их је тешко решити.

Узроци помераја домена

Један од значајних узрока помераја домена је начин на који се подаци за обучавање модела прикупљају. Веома лако је на неки начин унети пристра-

сност и изабрати податке који не одговарају окружењу у којем ће се модел извршавати.

Ако подаци из реалног окружења нису доступни једина могућност је коришћење података из сличног домена или генерисање вештачких података који никада не могу да буду исти као и подаци над којем ће се модел примењивати. У оба случаја се природно јавља потреба за прилагођавањем модела обученог на таквим подацима на новом реалном домену.

Чак и ако јесте могуће добити добре податке из релевантног окружења, постоји могућност да је то једноставно превише скупо. То је још један од разлога из ког се често иницијално морају користити подаци мањег квалитета што може знатно утицати на квалитет коначног модела.

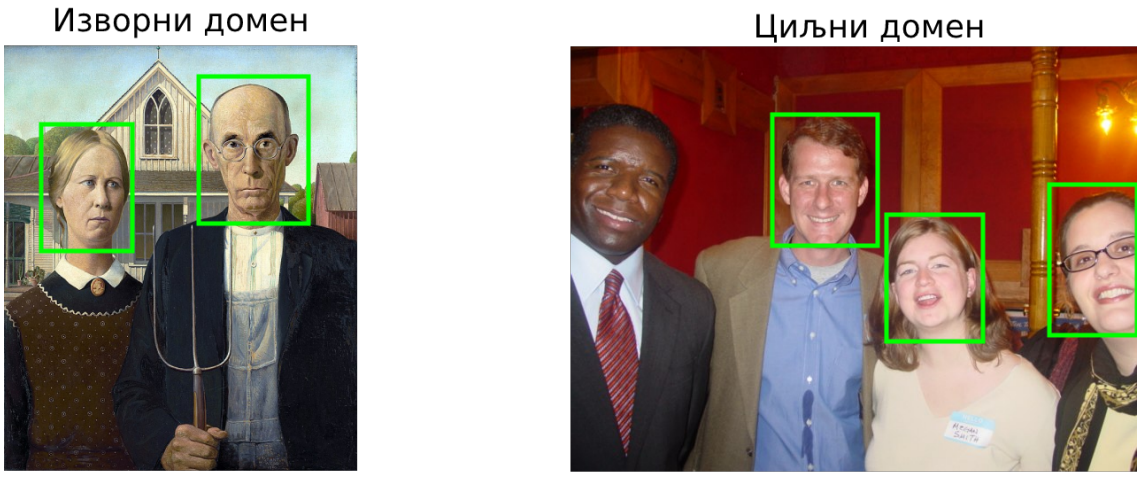
Велики проблем представља и избор репрезентативног скупа података. Током избора података за обучавање доноси се много одлука, од којих свака може у себи садржати неку лошу претпоставку која ће унети пристрасност у податке. Ако се узимају подаци са камере из неког подскупа свих објеката где ће се касније модел примењивати, постоји шанса да се у неким од објеката са чијих камера нису коришћене слике налази други тип камере.

Приликом прикупљања података за лечење болести мора се пазити да подаци буду бирани из популације која добро представља популацију људи који ће користити тај лек. Ако се изаберу само старији људи, то може унети пристрасност у скуп података. Старији људи на пример имају виши притисак што би из овако изабраног скупа изгледало као нормално, па би модел доносио лоше одлуке над млађим људима.

У случају прикупљања визуелних података из једног географског подручја у којем се већински налазе припадници једне расе, модел може имати проблема са препознавањем лица људи других раса ако би се користио у остатку света. Ово се може видети на слици 3.1

Поред увођења пристрасности током избора иницијалног скупа података, чак и за идеално изабран скуп се јавља проблем помераја домена током времена. Овај проблем се скоро увек јавља у реалним применама машинског учења због динамичне природе окружења у којима се извршава.

Ако је циљ класификовати постојање одређене болести, чак и ако се направи савршен модел за ту болест она може мутирати на разне начине и у потпуности променити природу проблема и резултате модела. У случају препознавања људи унутар одређених објеката одличан модел може постати



Слика 3.1: Пример лошег избора изворног домена где су прикупљени подаци само одређене расе људи као што се види на левој слици, због чега научени модел над циљним доменом не препознаје лица свих људи на слици.

неупотребљив ако сви људи крену да носе маске у случају неке пандемије.

Типови помераја домена

Скуп података који је прикупљен и који се користи за обучавање модела назива се изворни скуп података и долази из репрезентације реалног окружења које се назива изворни домен. Након што се изврши обучавање и избор најбољег модела над скупом из изворног домена он се извршава у реалном окружењу које се назива циљни домен. Претпоставља се да су доступни означени подаци из изворног домена и неозначени подаци из циљног домена.

Оно што нас формално занима у овом проблему је минимизација циљног ризика. За функцију губитка ℓ и парове (x, y) из одређене расподеле, ризик се дефинише као очекивани губитак за дати класификатор h:

$$R(h) = E_{x,y}[\ell(h(x), y)]$$

Као што је показано у раду [9] у ризик на циљном домену можемо убацити информацију о изворном домену на следећи начин:

$$\begin{aligned} R_T(h) &= \sum_{y \in Y} \int_X \ell(h(x), y) p_T(x, y) dx \\ &= \sum_{y \in Y} \int_X \ell(h(x), y) p_T(x, y) \frac{p_S(x, y)}{p_S(x, y)} dx \\ &= \sum_{y \in Y} \int_X \ell(h(x), y) p_S(x, y) \frac{p_T(x, y)}{p_S(x, y)} dx \end{aligned} \tag{3.1}$$

где p_S представља расподелу атрибута и циљне променљиве над изворним подацима а p_T над циљним подацима. Овако представљен ризик се може апроксимирати просеком:

$$R_T(h) = \frac{1}{n} \sum_{i=1}^{n} \ell(h(x_i), y_i) \frac{p_T(x_i, y_i)}{p_S(x_i, y_i)} dx$$

Битно је приметити да се узорци узимају из изворне расподеле а не циљне. Овим се постиже реална ситуација у којој се јавља потреба за прилагођавањем модела где се модел обучава над изворним доменом а грешка евалуира на циљном домену.

Разлике између изворног и циљног домена могу се поделити у 3 типа:

- класни померај (eng. *prior/class shift*);
- коваријатни померај (eng. *covariate shift*);
- концептни померај (eng. *concept shift*).

Заједничку расподелу можемо представити на два начина: $p(x, y) = p(x|y)p(y)$ и $p(x, y) = p(y|x)p(x)$. Померај домена се дели по типовима у зависности од тога која компонента расподеле се разликује између домена.

Класни померај подразумева ситуацију где се за $p(x, y) = p(x|y)p(y)$ условна вероватноћа која представља везу између x и y не мења али се зато пропорција класа између изворног и циљног домена мења.

За проблем класификације одређене болести, ако се након неког времена болест сузбије или мутира тако да се ређе јавља, циљна популација ће имати много мање случајева болесних људи него што је то био случај током обучавања модела над изворним доменом.

Овај проблем се може решити ако се зна на који начин су се промениле тежине класа, пошто се онда може поново обучити модел са адекватно отежаним класама. Уз означене податке из циљног домена се овај проблем стога може релативно једноставно решити.

Коваријатни померај подразумева ситуацију где се за $p(x, y) = p(y|x)p(x)$ условна вероватноћа која представља везу између x и y не мења исто као и код класног помераја, али се у овом случају пропорција између улазних атрибута изворног и циљног домена мења.

Овај тип промене у домену је предмет највећег броја истраживања и у остатку овог рада ће бити главна тема. Најчешћи узрок коваријатног помераја је лош избор података. На пример, ако је циљ проценити колико су људи у одређеном граду задовољни интернет сервисом а да би се то проценило упитник буде извршен само над људима у центру града који имају много боље услуге и не чине репрезентативан узорак за целу популацију. Из овог разлога је неопходно пажљиво прикупљати податке.

Коваријатни померај је због високе димензионалности простора у ком се налазе улазни атрибути тежак проблем, али је решив чак и са подацима из циљног домена који нису означени под условом да их има довољно.

Концептни померај се јавља у ситуацијама где се за $p(x, y) = p(y|x)p(x)$ условна вероватноћа која представља везу између x и y, за разлику од класног и коваријатног помераја, мења док пропорција између улазних атрибута изворног и циљног домена остаје иста.

На примеру класификације одређене болести, концептни померај би представљао мутацију болести тако да има другачије симптоме. У овом случају се мењају законитости између улазних података који су симптоми пацијента и циљне променљиве која је информација да ли је неко болестан или не.

Концептни померај је могуће решити, али су за то неопходни означени подаци из циљног скупа.

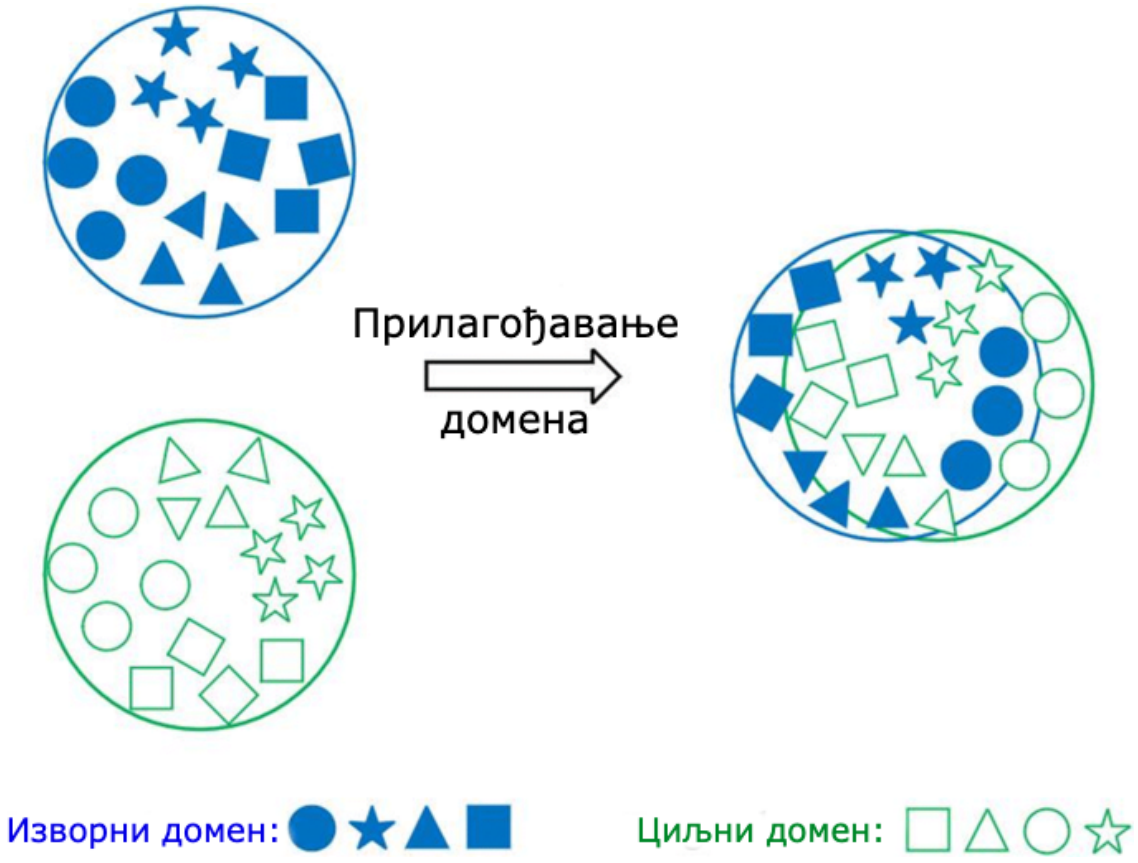
Такође је битно напоменути да захтев за означене податке из циљног домена који се јавља код концептног и класног помераја, иако наизглед звучи једноставно, може представљати велики проблем пошто често захтева људски рад и може да траје доста времена у зависности од проблема.

У реалним окружењима могу се јавити и генерални помераји домена, које представљају комбинацију једног или више типова помераја. Ово може бити скоро немогућ проблем за решити јер је за прилагођавање модела проме-

нама у домену неопходно постојање неке повезаности домена која се може искористити. У случају где су домени толико различити да скоро и немају сличности технике прилагођавања модела променама у домену се не могу успешно користити.

3.2 Приступи прилагођавању модела променама у домену

Прилагођавање модела променама у домену претпоставља да су доступне велике количине означених података из изворног домена и ограничене количине података из циљног домена. Илустрација прилагођавања модела променама домена се може видети на слици 3.2



Слика 3.2: Прилагођавање модела променама у домену

У случају коваријатног помераја, који је главна тема овог рада, подаци из циљног домена могу бити чак и неозначени. Ово доста олакшава аутоматиза-

цију процеса прилагођавања модела у реалним ситуацијама пошто не захтева процес означавања података који често захтева људски напор и траје дуго.

Неки од приступа за прилагођавање модела коваријатној промени домена су:

- приступи базирани на отежавању (eng. *weighting based*);
- пресликавање потпростора (eng. *subspace mappings*);
- прилагођавање модела промени домена супарничким тренингом (eng. *adversarial domain adaptation*).

Код приступа базираних на отежавању поставља се питање како апроксимирати тежине $w_{x_i} = \frac{P_T(x_i)}{P_S(x_i)}$ на адекватан начин. Обе расподеле у једначини се могу апроксимирати као гаусовске расподеле [12]. Алтернатива је да се користи оцена густине расподеле заснована на кернелима [2].

Ако се домени налазе у различитим потпросторима онда постоји пресликавање са циљног домена на изворни домен, које би омогућило да се модел обучен на изворном домену користи и над циљном домену. Приступи базирани на пресликавање потпростора почивају на овој претпоставци. Пресликавање потпростора се може извести тако што се изворни и циљни домен представе потпросторима који почивају на њиховим сопственим векторима. У раду [4] показано је како се пресликавање над тако описаним потпросторима може извршити ефикасно.

Прилагођавање модела промени домена супарничким тренингом се може извршити дискриминативним или генеративним моделима. Један од дискриминативних приступа [5] ће детаљније бити описан у остатку овог рада. У том приступу постојећој дубокој мрежи која решава одређени проблем се додаје још један део који служи да израчунате атрибуте учини инваријантним у односу на одређене промене домена. Такође постоје и генеративни приступи који користе варијације генеративних супарничких мрежа, описаних у претходном поглављу, да пресликају слике из циљног домена у изворни домен [13].

Глава 4

Прилагођавање домену супарничким учењем

У овом поглављу ће детаљно бити описан цео експеримент. Прво ће бити приказани скупови података који су коришћени као и начин на који су припремљени. Након тога биће описан приступ прилагођавања модела промени домена супарничким учењем који је коришћен током експеримената. Коначно, биће описан процес обучавања и приказани резултати и евалуација експеримента.

4.1 Припрема скупова података

У наредним експериментима биће коришћени скупови података који садрже руком писане цифре. Ови скупови се често користе у области рачунарског вида зато што је проблем релативно једноставан и брзо се могу добити резултати па тиме и служе као добра провера да предложени приступ има смисла.

Биће разматрана 3 скупа података:

- *MNIST*
- *MNIST-M*
- *SVHN*

MNIST

Скуп података *MNIST* [3] се састоји из 70000 црно-белих слика димензија 28×28 слика ручно писаних цифара. Подељен је на тренинг скуп од 60000 слика и скуп за тестирање који се састоји од 10000 слика. На слици 4.1 приказано је неколико слика из скупа.

Слика 4.1: скуп података *MNIST*

Овај скуп је један од најпознатијих и највише коришћених скупова података који се користи у области рачунарског вида. На њему се постижу одлични резултати од преко 99% тачности због чега важи за „решен" проблем. Због опште прихваћености у области често се користи као један од иницијалних скупова за почетну валидацију нових приступа.

Такође, пошто је добро дефинисан и једноставан проблем, релативно брзо се над њиме могу обучавати мреже што омогућава брзе итерације и чини велику предност приликом израде новог приступа. Због једноставности се не може користити за поређење све комплекснијих модела који настају у обла-

сти рачунарског вида, али је и даље користан за проверавање коректности имплементације.

Скуп се налази припремљен и спреман за коришћење унутар библиотеке *torchmetrics* која је коришћена у овом раду.

MNIST-M

Скуп података *MNIST-M* [5] је први пут коришћен у радовима везаним за прилагођавање промени домена [5]. Састоји се од 59001 слика у тренинг скупу и 90001 слика у скупу за тестирање.

Слика 4.2: скуп података *MNIST-M*

Добијен је комбиновањем слика из скупа података *MNIST* са насумично изабраним деловима обојених слика из скупа података *BSDS500* који се користи за проблеме сегментације. Комбиновање је постигнуто тако што се иницијално узме исечени део слике из скупа *BSDS500* а потом се инвертују боје пиксела који на одговарајућој слици из скупа *MNIST* одређују цифру. На овај начин се добијају слике цифара које су подједнако лаке за препознавање човеку као што се може видети на слици 4.2, али се домен довољно разликује од скупа *MNIST*.

Скуп је преузет од оригиналних аутора рада [5] и прилагођен за коришћење унутар библиотеке *pytorch* за потребе овог рада.

SVHN

Скуп података *SVHN* [10] садржи слике уличних табли са бројевима. Он се даље може процесирати у исечене слике цифара на табли димензија 32×32.

Слика 4.3: скуп података *SVHN*

Подељен је на тренинг скуп који се састоји од 73257 слика и на скуп за тестирање који броји 26032 слика.

Као што се може видети на слици 4.1 ово је најкомплекснији од скупова података који су коришћени у раду. Слике су из реалног света и у великом броју случајева се налазе делови других цифара при ивицама слике, које отежавају класификацију.

Скуп се налази припремљен и спреман за коришћење унутар библиотеке *torchmetrics* која је коришћена током овог рада.

4.2 Опис приступа

Архитектура која је коришћена преузета је из рада [5] и састоји се из 3 дела:

- део за екстракцију атрибута;
- класификатор циљне променљиве;
- класификатор домена.

Екстракција атрибута

Први део мреже је конволутивна мрежа која на излазу даје израчунате атрибуте који се даље могу користити за класификацију. Излаз првог дела мреже представља улаз у друга два класификатора. За овај део се могу користити различите архитектуре мрежа у зависности од комплексности проблема који се решава. За проблем препознавања цифара није потребно користити модерне архитектуре са великим бројем слојева већ су сасвим довољне и једноставније конволутивне мреже са неколико узастопних слојева конволуције и агрегације уз примену слојева регуларизације.

Класификатор циљне променљиве

Класификатор циљне променљиве је потпуно повезана мрежа која служи за решавање оригиналног проблема што је у случају овог рада препознавање цифара. Комплексност овог дела мреже такође зависи од комплексности проблема. На крају поглавља биће детаљније описане архитектуре које су коришћене током експеримената.

Класификатор домена

Класификатор домена је такође потпуно повезани део мреже који омогућава мрежи да прилагоди екстрактор атрибута померају домена. Ово се постиже слојем за обртање градијента којим су израчунати атрибути повезани за класификатор домена.

Јачина утицаја информација из класификатора домена одређена је параметром λ који директно утиче на то колико ће градијент ове главе мреже утицати на измену параметара дела мреже за екстракцију атрибута. Овај параметар се на почетку поставља на 0 и потом постепено повећава током тренинга. Детаљнији опис како утиче на остатак мреже и како се мења биће дат у наставку рада.

Слој обртања градијента

Слој за обртање градијента се приликом пропагације података унапред кроз мрежу понаша као функција идентитета. Са друге стране, приликом пропагације уназад унутар овог слоја се градијент множи са негативном кон-

стантом. На овај начин се израчунати атрибути мењају тако да на основу њих класификатору домена буде што теже да научи разлике између домена. Претпоставка је да ће се због овог услова учити атрибути који су неосетљиви на домен и који у себи садрже само законитости релевантне у свим доменима.

Формално овај слој се може гледати као псеудо функција $R_\lambda(x)$ која је дефинисана помоћу 2 једначине за пропагацију унапред и пропагацију уназад:

$$R_\lambda(x) = x$$
$$\frac{\partial R_\lambda}{\partial x} = -\lambda I$$

Ова псеудо функција се лако може оптимизовати техникама градијентног спуста које су описане у поглављу 2.

Детаљан опис архитектура

У раду су коришћене 2 архитектуре које прате исти приступ, једноставнија за скуп података *MNIST* и комплекснија за скупове података *MNIST-M* и *SVHN*.

На слици 4.4 под (а) описана је једноставнија мрежа. Део за екстракцију атрибута се састоји од 2 блока, сваки од којих садржи: слој изостављања, слој конволуције, активациони ReLU слој, слој унутрашње стандардизације, слој агрегације. Делови мреже за класификаторе циљне променљиве и домена поред слојева на слици такође садрже и слојеве изостављања и унутрашње стандардизације.

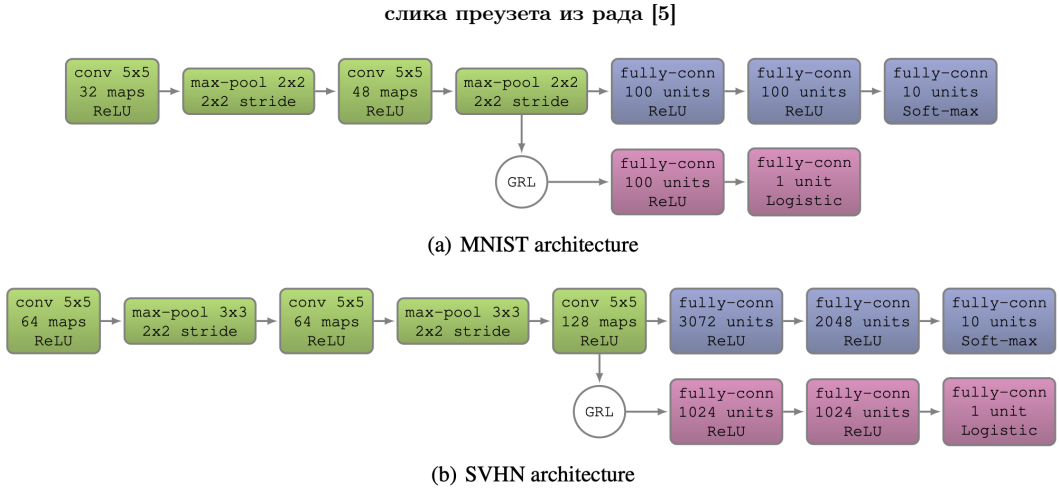
Под (б) је описана комплекснија мрежа која у делу за екстракцију атрибута има један блок више. Осим тога, комплекснија је и по броју различитих канала у делу за екстракцију атрибута као и по величини потпуно повезаних слојева унутар класификатора.

Количина слојева изостављања и њихови хиперпараметри изабрани су по узору на оригиналан рад у којем је слој изостављања приказан.

Оно што је битно напоменути је да је у обе архитектуре први слој класификатора домена слој обртања градијента описан у претходној секцији.

4.3 Експерименти и евалуација

У овој секцији детаљно ће бити описан цео експеримент укључујући начин на који су припремљени подаци, како је текао процес обучавања и евалуација



Слика 4.4: Архитектура мрежа. Под (а) описана је једноставнија MNIST архитектура док је под (б) описана комплекснија SVHN архитектура. Различитим бојама су обојени различити делови мреже: зеленом бојом – екстракција атрибута, љубичастом бојом – класификатор циљне променљиве, црвеном бојом – класификатор домена. Слој обртања градијента означен је са *GRL* и обојен белом бојом.

приступа над различитим скуповима података.

Припрема података

Експеримент је подељен у неколико делова, по паровима скупова података над којима се врши прилагођавање модела променама у домену. Одабрани парови који су коришћени су:

- MNIST → MNIST-M
- SVHN → MNIST

За сваки од изабраних парова скупова података изабрани су скупови за тренинг, валидацију и тест. Скупови за тренинг и валидацију се састоје од једнаког броја слика из изворног скупа и из циљног скупа података. Ова одлука је донета због природе обучавања модела где се током сваког корака оптимизације узима исти број из изворног и циљног скупа како би учење било балансирано. Постоје разне технике за решавање проблема небалансираних скупова и ово је само једно од једноставнијих техника која је дала добре резултате.

Скупови за евалуацију модела су независни и исти као и скупови за евалуацију изворног и циљног скупа података. Ови скупови су коришћени за добијање коначних резултата који ће бити описани у раду.

Сви скупови података су претпроцесирани стандардизацијом тако што су средње вредности и стандардна девијација израчунате над скуповима за тренинг одговарајућег скупа података. Тако израчунате вредности са скупа за тренинг су коришћене за стандардизацију података над сва 3 скупа како би праћење перформанса модела над валидационом скупом и коначна евалуација над скупом за тест биле непристрасне. Нису додаване аугментације током припреме података.

Скупови података *MNIST* і *SVHN* су доступни у библиотеци *torchvision* што је олакшало њихово коришћење. Скуп података *MNIST-M* је преузет од оригиналног аутора и прилагођен за коришћење унутар библиотеке *torchvision*.

Посебно је обраћена пажња да подаци из циљног скупа података не буду коришћени током тренирања над изворним скупом података.

Процес обучавања

Обучавање модела се извршавало у корацима од по 128 инстанци од којих је пола било из изворног домена а пола из циљног домена. За половину из изворног домена су коришћене и ознаке док је за половину из циљног домена коришћена само информација о домену.

Класификатор домена на почетку обучавања не утиче у потпуности на процес учења већ се постепено повећава његов значај коришћењем адаптационог параметра λ. Овај параметар утиче директно на слој обртања градијента и мења се на следећи начин:

$$\lambda_p = \frac{2}{1 + exp(-\gamma \cdot p)} - 1,$$

где је параметар γ постављен на 10 у свим експериментима као што је то случај и у оригиналном раду.

Током процеса обучавања, перформансе модела се могу проценити на основу грешке класификатора циљне променљиве на изворном домену као и грешке класификатора домена на оба домена. Након обучавања се приликом

евалуације модела додатно може проверити и грешка класификатора циљне променљиве на циљном домену пошто је то циљ прилагођавања модела.

Опис експеримената

Циљ експеримента је проверити колико прилагођавање модела циљном домену доприноси перформансама модела над циљним доменом. Да би се ово постигло обучавају се 3 модела. Изворни модел служи да постави доњу границу перформанси, циљни модел је за горњу границу модела, док се модел из изабраног приступа евалуира поређењем са изворним и циљним моделима.

Први модел се обучава само на изворном скупу и евалуира се на циљном домену. Овај модел биће назван изворни модел и он се користи као доња граница за приступ.

Други модел служи да постави теоријске границе перформанси које се могу постићи на циљном скупу података. Он се обучава искључиво над циљним подацима. Упоређивањем модела из приступа са циљним моделом можемо да видимо колико добро се прилагодио на циљни скуп коришћењем супарничког прилагођавања.

Трећи модел представља модел обучен супарничким учењем над оба домена, тако да је изворни скуп података коришћен за класификатор циљне променљиве док су оба скупа коришћена за класификатор домена. Ово је модел који поредимо са осталим моделима и чије перформансе желимо да евалуирамо. У идеалном случају перформансе овог модела теже ка перформансама трећег циљног модела, али је очекивана ситуација да се понаша лошије од њега али зато боље од изворног модела

У наставку ће бити описана два експеримента. Први који се бави прилагођавањем модела променама у домену са скупа података *MNIST* на скуп *MNIST-M* и други који се бави променама домена са скупа *SVHN* на скуп података *MNIST*.

MNIST → MNIST-M

Први експеримент као изворни скуп података користи *MNIST* а као циљни скуп података *MNIST-M*. Иако су ова два домена слична, довољно се разликују да модел обучен само на изворном домену даје поприлично лоше резултате на циљном моделу. У табели 4.1 може се видети да наш изворни модел

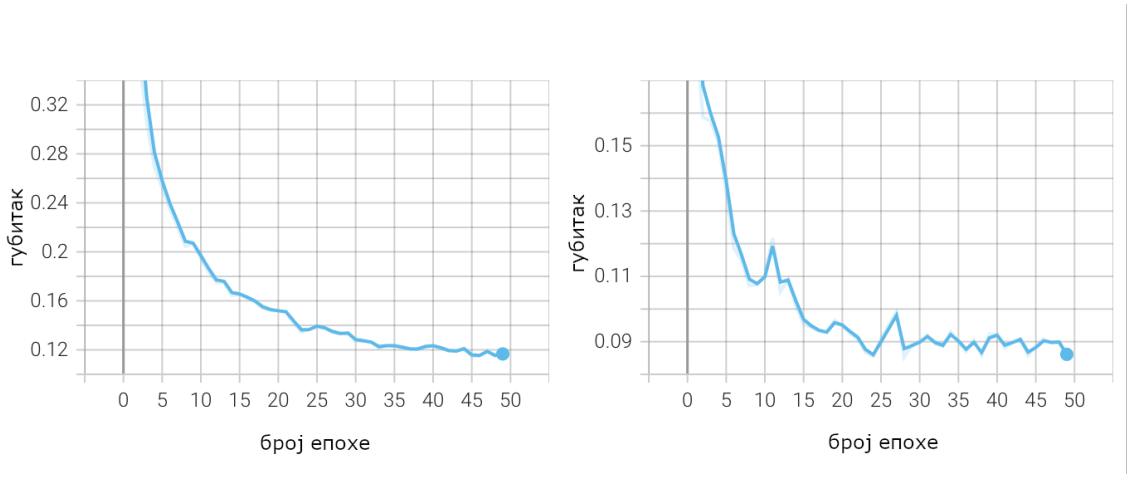
MNIST → *MNIST-M*	Постигнута тачност	Тачност из рада [5]
Изворни модел	.4512	.5749
Предложени приступ	**.8184**	.8149
Циљни модел	.9720	.9891

Табела 4.1: Поређење постигнутих резултата приступа са оригиналним радом над изворним скупом података *MNIST* и циљним скупом података *MNIST-M*.

постиже само 45.12% тачности а исти модел из изворног рада са којим се пореди је постигао знатно већу тачност – 57.49%. Разлог за ову разлику је највероватније коришћење аугментација у оригиналном раду које нигде нису наведене па нису биле коришћене током његовог репродуковања.

Циљни модели су доста слични по перформансама између оригиналног рада и резултата добијених у овом раду, али су перформансе у оригиналном раду ипак боље за 1.7%. У оригиналном раду постигнута је 98.91% тачност, док је у овом раду добијена 97.20% тачност.

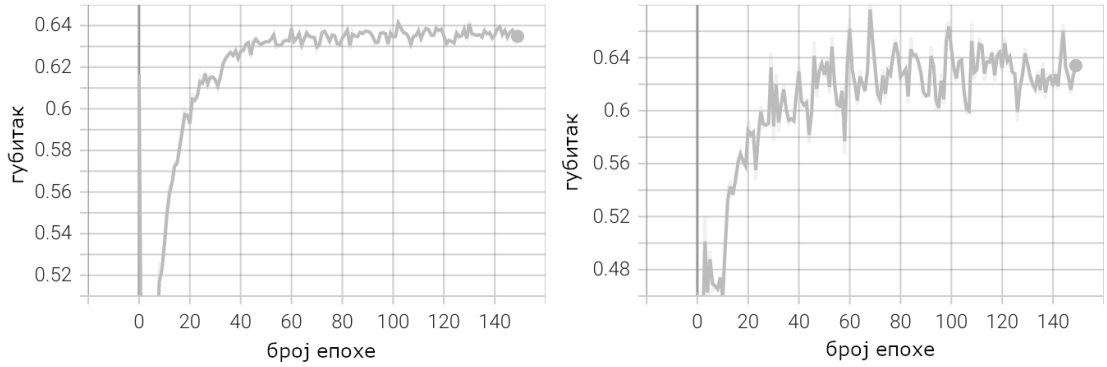
Модел из предложеног приступа у овом раду се понаша знатно боље од оба изворна модела са 81.84% тачности. Ово је упоредиво и са тачности коју је постигао предложени приступ у оригиналном раду – 81.49%. На основу овог скока можемо закључити да описани приступ има смисла и да за довољно сличне домене у пракси заиста може да допринесе скоку у перформансама које наговештава теорија.



Слика 4.5: График функције губитка циљног модела над тренинг и валидационим скуповима података током обучавања

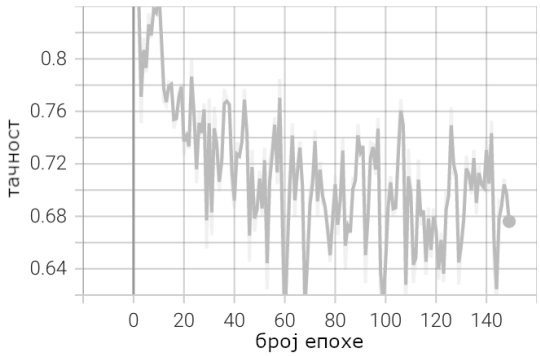
Као што се може видети са графика 4.5, функција губитка циљног модела се добро понаша и опада континуално током обучавања. Такође можемо

приметити да не долази до преприлагођавања будући да је вредност функције губитка на валидационом скупу података мања од губитка на скупу за тренинг.



Слика 4.6: Фунција губитка класификатора домена за модел из приступа над скуповима за тренинг и валидацију током обучавања

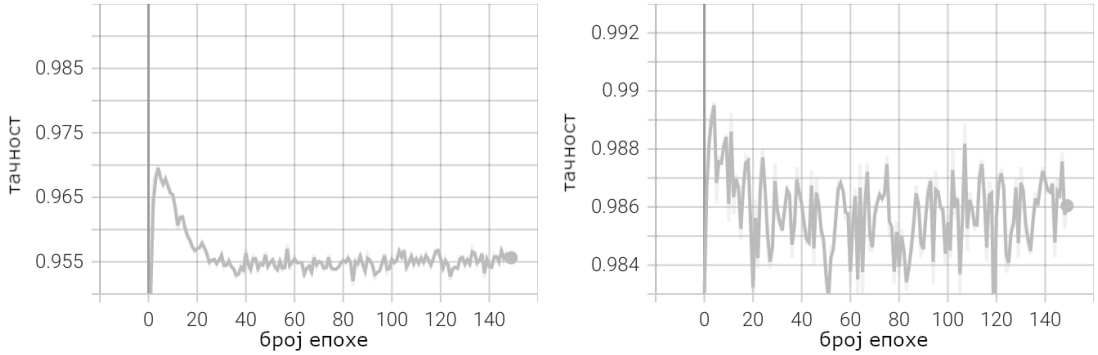
Током прилагођавања модела домену праћена је и функција губитка над класификатором домена. На графику 4.6 види се да губитак класификатора домена расте током обучавања на скуповима за тренинг и валидацију. Разлог за ово понашање које наиглед може изгледати погрешно је што је над овим класфикатором примењен слој обртања градијента. Узимајући то у обзир, раст губитка је очекивано и пожељно понашање.



Слика 4.7: Тачност класификатора домена за модел из изабраног приступа над валидационим скупом током обучавања

Тачност класификатора домена опада што је такође добар знак. У идеал-

ном случају он би за бинарни класификатор са балансираним класама тежио ка 50% а на слици 4.7 се видети да она заиста опада током обучавња. Такође се може приметити да је тачност поприлично нестабилна и брзо се мења из епохе у епоху. Иако ово није пожељно понашање, оно се често дешава приликом супарничког учења.



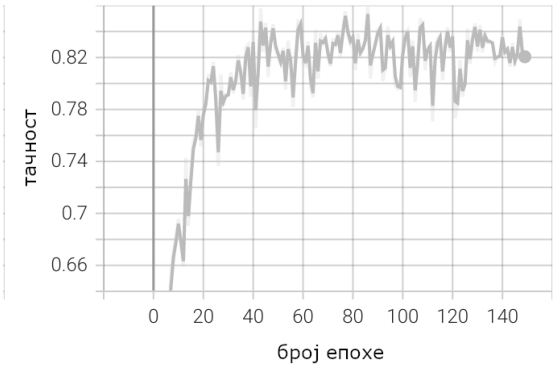
Слика 4.8: Тачност класификатора циљне променљиве за модел из приступа над валидационим скупом изворних података током обучавања

Интересантно је приметити и пад у перформансама модела над изворним доменом током прилагођавања над циљним доменом који се може видети на графику 4.8. Иако се тачност није знатно смањила и даље показује да се боља генерализација над циљним доменом плаћа благим губитком у перформансама над изворним доменом. Ово није пожељно понашање и неопходно је додатно истраживање да би се пронашао начин како спречити овај пад у перформансама, упркос томе што није велики.

Коначно, на графику 4.9 се може видети побољшање перформанси на циљном домену након прилагођавања модела. Овде се може приметити блага нестабилност која је и очекивана будући да се ова вредност није директно оптимизовала већ индиректно кроз класификатор домена. Упркос благе нестабилности се јасно види велико повећање тачности која је за изворни модел знатно мања.

SVHN → MNIST

Други експеримент тежи је од првог због већих разлика у доменима. Као изворни домен одабран је скуп података *SVHN* док је за циљни домен ода-



Слика 4.9: Тачност класификатора циљне променљиве за модел из приступа над валидационим скупом циљних података током обучавања

бран скуп података *MNIST*. Изворни домен је сада доста комплекснији од циљног. Слике изворног домена су из реалног света са много више дистракција и позадинског шума док су слике циљног домена црно-беле слике руком писаних цифара. У табели 4.2 приказане су тачности модела постигнуте у раду упоређене са оригиналним радом. Може се видети да изворни модел у раду постиже тачност од 59.19%, док модел је наш модел постигао бољу тачност од 66.45%.

Циљни модели у овом раду и у оргиналном раду су као и у претходном експерименту веома слични. У оригиналном раду добијена је тачност од 99.51% док је у овом раду тачност незнатно мања – 99.38%.

Најбитније су перформансе модела добијеног прилагођавањем над циљним доменом, који у овом раду са тачности од 74.13% постиже боље резултате и од изворних модела и од истог модела у оригиналном раду чија тачност износи 71.01%. Повећање тачности у односу на оригинални рад је највероватније узроковано појачаном регуларизацијом и увођењем слојева унутрашње стандардизације.

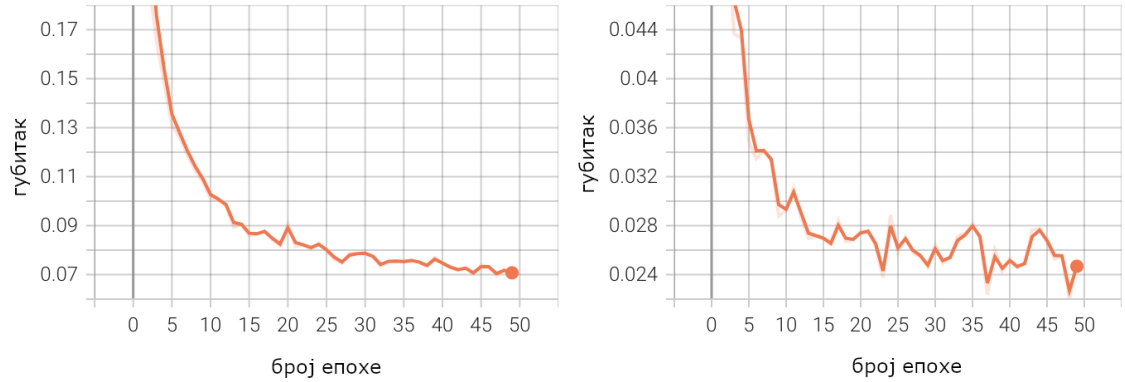
Скок у перформансама није велики у односу на изворни модел због велике разлике у доменима, али ипак показује да се овим приступом и у таквим случајевима могу побољшати перформансе.

На основу графика 4.10 се може закључити да је циљни модел адекватно обучен и то без преприлагођавања. Ово је неопходан услов за даљу евалуацију модела јер указује да су резултати постигнути овим моделом добри.

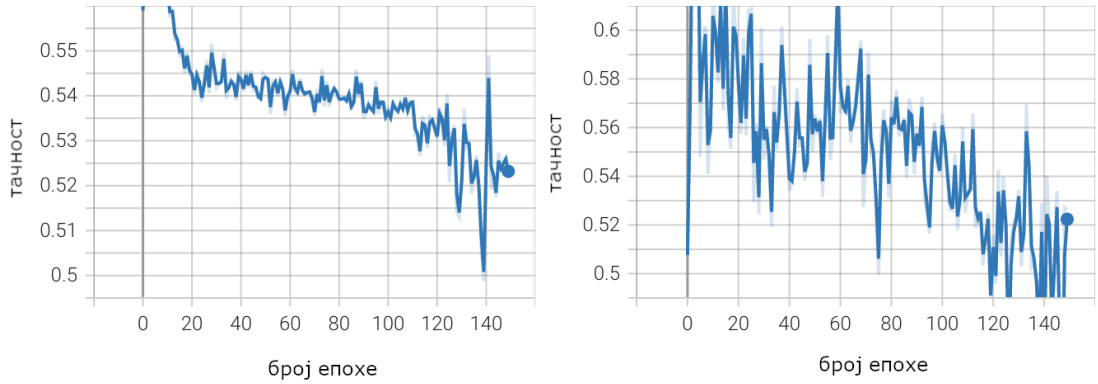
Тачност класификатора домена се овде понаша скоро идеално. Као што

SVHN → *MNIST*	Постигнута тачност	Тачност из рада [5]
Изворни модел	.6645	.5919
Предложени приступ	**.7413**	.7107
Цильни модел	.9938	.9951

Табела 4.2: Поређење постигнутих резултата приступа са оригиналним радом над изворним скупом података *SVHN* и цильним скупом података *MNIST*.



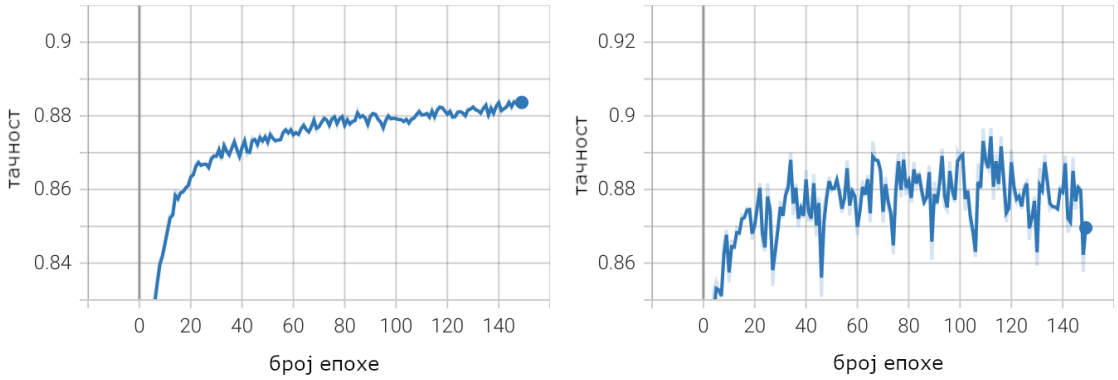
Слика 4.10: График функције губитка цильног модела над тренинг и валидационим скуповима података током обучавања



Слика 4.11: Тачност класификатора домена за модел из приступа над валидационим скупом током обучавања

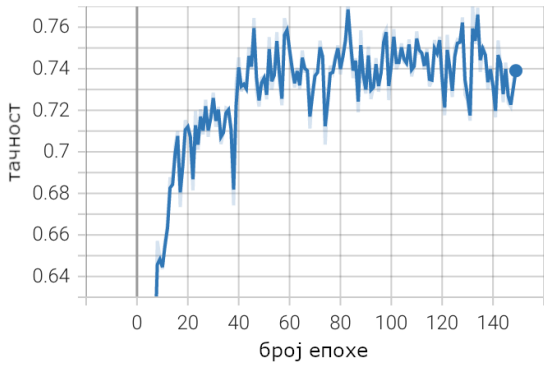
се може видети на графику 4.11 тачност тежи ка 50% што и јесте пожељно понашање за случај када имамо само две балансиране класе.

Као и у претходном експерименту приметан је пад у перформансама модела над изворним доменом током прилагођавања. Пад у тачности који се може видети на графику 4.12 је значајан будући да изворни модел постиже



Слика 4.12: Тачност класификатора циљне променљиве за модел из изабраног приступа над валидационим скупом изворних података током обучавања

тачност од 92% пре прилагођавања док након прилагођавања на изворном скупу постиже само 88% тачности.



Слика 4.13: Тачност класификатора циљне променљиве за модел из приступа над валидационим скупом циљних података током обучавања

На графику 4.13 се може видети побољшање над циљним доменом након прилагођавања модела. Ту се може приметити блага нестабилност која и није необична будући да се на њу утиче индиректно класификатором домена. Упркос томе види се јасан скок у перформансама који је био циљ овог рада.

Глава 5

Закључак

Област прилагођавања модела машинског учења променама у домену је изузетно важна за даљи развој аутоматизације система који користе технике машинског учења и самим тим за гарантовање бољих решења кроз дужи временски период.

Главни допринос овог рада је био показати да се постојеће технике могу успешно применити на проблем препознавања цифара и то са довољно добрим резултатима који указују да се приступ може применити и за комплексније проблеме.

Алгоритам прилагођавања модела супарничким учењем са слојем обртања градијента постигао је задовољиве резултате приликом промене домена на проблему препознавања цифара. Показано је да прилагођавање модела са комплекснијег домена на једноставнији домен даје знатно боље резултате него обучавање модела само над комплекснијим доменом. Показано је да је проблем преласка са једноставнијег домена на много комплекснији домен тежак проблем који захтева додатно истраживање.

У оквиру рада су приказани и основни концепти машинског учења као и опис области прилагођавања модела промени у домену. Са овим знањем заинтересовани истраживачи се могу упустити у детаљнији преглед области и померање њених граница.

Систем направљен за потребе рада поред решавања проблема препознавања цифара може се користити за произвољне проблеме класификације и разне домене. Најважнији услов за успешно коришћење система је да домени морају бити довољно слични.

Иако су резултати приказани у раду задовољавајући може се приметити

да и даље постоји доста простора за напредак пошто резултати које решење даје и даље нису близу идеалном решењу. Једна од интересантних идеја за напредак је коришћење одвојених слојева за унутрашњу стандардизацију за различите домене. Систем који би се добио коришћењем ове идеје не захтева ознаке над циљним доменом тако да се у потпуности може аутоматизовати као и приказани приступ.

Литература

[1] Martín Abadi и др. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems*. 2016. DOI: 10.48550/ARXIV.1603.04467. URL: https://arxiv.org/abs/1603.04467.

[2] Steffen Bickel, Michael Brückner и Tobias Scheffer. „Discriminative Learning Under Covariate Shift". У: *Journal of Machine Learning Research* 10.75 (2009), стр. 2137–2155. URL: http://jmlr.org/papers/v10/bickel09a.html.

[3] Li Deng. „The mnist database of handwritten digit images for machine learning research". У: *IEEE Signal Processing Magazine* 29.6 (2012), стр. 141–142.

[4] Basura Fernando и др. „Unsupervised Visual Domain Adaptation Using Subspace Alignment". У: *2013 IEEE International Conference on Computer Vision* (2013), стр. 2960–2967.

[5] Yaroslav Ganin и Victor Lempitsky. *Unsupervised Domain Adaptation by Backpropagation*. 2014. DOI: 10.48550/ARXIV.1409.7495. URL: https://arxiv.org/abs/1409.7495.

[6] Ian Goodfellow, Yoshua Bengio и Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[7] Ian J. Goodfellow и др. *Generative Adversarial Networks*. 2014. DOI: 10.48550/ARXIV.1406.2661. URL: https://arxiv.org/abs/1406.2661.

[8] Sergey Ioffe и Christian Szegedy. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. 2015. DOI: 10.48550/ARXIV.1502.03167. URL: https://arxiv.org/abs/1502.03167.

[9] Wouter M. Kouw и Marco Loog. *An introduction to domain adaptation and transfer learning*. 2018. DOI: 10.48550/ARXIV.1812.11806. URL: https://arxiv.org/abs/1812.11806.

[10] Yuval Netzer и др. „Reading Digits in Natural Images with Unsupervised Feature Learning". У: *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*. 2011. URL: http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf.

[11] Adam Paszke и др. „PyTorch: An Imperative Style, High-Performance Deep Learning Library". У: *Advances in Neural Information Processing Systems 32*. Ур. H. Wallach и др. Curran Associates, Inc., 2019, стр. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[12] Hidetoshi Shimodaira. „Improving predictive inference under covariate shift by weighting the log-likelihood function". У: *Journal of Statistical Planning and Inference* 90.2 (2000), стр. 227–244. ISSN: 0378-3758. DOI: https://doi.org/10.1016/S0378-3758(00)00115-4. URL: https://www.sciencedirect.com/science/article/pii/S0378375800001154.

[13] Ashish Shrivastava и др. *Learning from Simulated and Unsupervised Images through Adversarial Training*. 2016. DOI: 10.48550/ARXIV.1612.07828. URL: https://arxiv.org/abs/1612.07828.

[14] Nitish Srivastava и др. „Dropout: a simple way to prevent neural networks from overfitting". У: *J. Mach. Learn. Res.* 15 (2014), стр. 1929–1958.

[15] Richard S. Sutton и Andrew G. Barto. *Reinforcement Learning: An Introduction*. Second. The MIT Press, 2018. URL: http://incompleteideas.net/book/the-book-2nd.html.

[16] Eric Tzeng и др. *Adversarial Discriminative Domain Adaptation*. 2017. DOI: 10.48550/ARXIV.1702.05464. URL: https://arxiv.org/abs/1702.05464.

[17] Растко Ђорђевић. *Систем развијен током рада Прилагођавање модела машинског учења промени домена супарничким тренингом*. on-line at: https://github.com/leonardovlibido/unsupervised-adversarial-domain-adaptation. 2022.

[18] Анђелка Зечевић Младен Николић. *Машинско Учење*. http://ml.matf.bg.ac.rs/readings/ml.pdf. 2019.

Биографија аутора

Растко Ђорђевић (*Београд, 22. јануар 1996*) завршио је основну и средњу школу у Београду. Након средње школе 2014. године уписује информатички смер на Математичком факултету у Београду који завршава 2017. године. Исте године уписује мастер студије смера информатика на истом факултету. У овом периоду креће да се интересује за машинско учење похађањем разних курсева и летње школе *ПСИМЛ* чији је главни организатор Мајкрософт. Крајем 2018. године се запошљава као сарадник у настави на Математичком факултету на ком ради 2 године до краја 2020. године. Почетком 2020. године се запошљава као истраживач у области вештачке интелигенције у фирми *Everseen* где се бави разним проблемима рачунарског вида. Наредне године постаје вођа тима где добија прилику да води 2 производа, од којих један покреће од нуле и доводи до производа спремног за продукцију. У јулу 2022. године прелази на позицију менаџера производа у истој фирми где и даље ради. Као менаџер производа добија прилику да преузме одговорност над комплетним производима и води их до успешне реализације.